UPC

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Departament de Llenguatges i Sistemes Informàtics

# Inductive Reasoning and Reconstruction Analys
# Two Complementary Tools for Qualitative Faul
# Monitoring of Large–Scale Systems

DOCTORAL THESIS

presented by:

Alvaro de Albornoz Bueno

June, 1996.

# Inductive Reasoning and Reconstruction Analysis: Two Complementary Tools for Qualitative Fault Monitoring of Large–Scale Systems

**Alvaro de Albornoz Bueno**
Departament d'Enginyeria de Sistemes i Automàtica
Universitat Politècnica de Catalunya

**TESI DE DOCTORAT**

Presentada al Departament de Llenguatges i Sistemes Informàtics, dins del Programa de Doctorat en Intel.ligència Artificial, per a obtenir el Títol de Doctor en Ciències amb Especialitat en Informàtica.

Juny, 1996.

**Directors:**

François E. Cellier
Electric & Comp. Eng. Dept.
University of Arizona

Rafael M. Huber
Institut de Robòtica i Inf.
Univ. Politécnica de Cataluña

# Inductive Reasoning and Reconstruction Analysis: Two Complementary Tools for Qualitative Fault Monitoring of Large–Scale Systems

**Alvaro de Albornoz Bueno**
Departament d'Enginyeria de Sistemes i Automàtica
Universitat Politècnica de Catalunya

**TESI DE DOCTORAT**

Juny, 1996.

Aprovació del Director Extern:

François E. Cellier, Ph.D.
Electric and Computer Engineering Department
University of Arizona
Tucson, AZ, 85721 USA.
E_mail: cellier@ece.arizona.edu

# Inductive Reasoning and Reconstruction Analysis: Two Complementary Tools for Qualitative Fault Monitoring of Large–Scale Systems

**Alvaro de Albornoz Bueno**
Departament d'Enginyeria de Sistemes i Automàtica
Universitat Politècnica de Catalunya

**TESI DE DOCTORAT**

Juny, 1996.

Aprovació del Director Intern:

Dr. Rafael M. Huber
Institut de Robòtica i Informàtica (IRI)
Universitat Politècnica de Catalunya
Edifici Nexus, Gran Capità 2–4
08034 Barcelona, España
E_mail: huber@ic.upc.es

# Aknowledgments

Finalmente este trabajo está llegando a su culminación. Como en toda tesis, ha habido períodos de arduo trabajo combinados con otros de franco desánimo, por decir lo menos. De cualquier forma, ha sido una labor y un trabajo excitantes.

Varias personas me han acompañado a lo largo de todos estos años de retorno a la escuela. Ellas son:

François E. Cellier, codirector de la tesis, que devino en consejero y amigo de tiempo completo y que se ha convertido en mi "Doktorvater", que en su lengua alemana significa mi padre científico. Efectivamente, esa ha sido la función que ha llevado a cabo durante todos estos años y que seguirá llevando, dado que la condición de hijo del Doctor Padre no se pierde nunca. Gracias por todo François, pero sobretodo, por estar tan cerca siempre.

Rafael M. Huber, codirector de la tesis, padre también del invento ya que en su día me presentó con algo llamado "Simulación Cualitativa". Rafael resolvió todo aquello que hace que la mayoría de los doctorados no se terminen nunca, arregló los líos que armé entre la burocracia universitaria, que no fueron pocos, y consiguió los medios económicos y materiales para llevar a cabo esta investigación.

Los otros cuatro integrantes de lo que llamamos "FAPAS", un grupo de investigacíon, camaradería y amistad compuesto por (en estricto orden alfabético): Angela Nebot, Fina López, Paco Mugica, y Sebastián Medina. Que se han convertido realmente en mi familia de Barcelona, condición esta del parentesco adquirido que también se mantiene durante toda la vida. Gracias a los cuatro. Ya desde este momento los extraño como enano.

Patricia Caratozzolo, que se preocupó por esta tesis más que por la suya propia y que aguantó estoicamente las prisas del final. Gracias Patricia por absolutamente todo.

De entre todas las demás personas que de una u otra forma han tenido algo que ver con el desarrollo de esta tesis, quiero hacer patente mi agradecimiento a:

Javier Martín y Manel Puigbó, encargado y maestro, respectivamente, del

Laboratorio de Control Automático, donde he desarrollado esta investigación. Ambos, siempre dispuestos a ayudar, se han vuelto estupendos amigos míos.

Juán Carlos Ramos Pablos, cuyo modelo del reactor nuclear se utiliza en esta tesis.

El trabajo de investigación de esta tesis doctoral ha sido financiado por la *Comisión Interministerial de Ciencia y Tecnología*, de España, bajo los proyectos CICYT TIC 90–0711 and CICYT TIC 93–0447.

Mi estancia doctoral en la Universidad Politécnica de Cataluña, en Barcelona, ha sido posible gracias a dos becas. La primera, concedida por el *Instituto de Cooperación Iberoamericana*, de Espanã, y la segunda, por el *Consejo Nacional de Ciencia y Tecnología*, de México. Ayuda económica que agradezco infinitamente.

# Summary of the Doctoral Thesis

Research in the field of automated plant supervision, fault detection, and fault diagnosis of industrial processes has experienced a spectacular growth during the past 25 years. In the beginning, only analytical techniques were used for such purposes, but in the sequel, qualitative approaches have begun to play an ever more important role in all of these activities, partly employing techniques of qualitative modeling and simulation, spin–offs of astounding advances made in the fields of Artificial Intelligence, Qualitative Reasoning, Fuzzy Logic, and the like.

One serious problem that has haunted control engineers and artificial intelligence researchers alike from the early days on is the problem of *information overload.* Human plant controllers are easily overwhelmed by the sheer mass of information available to them for decision making in any real–time situation, but the same is true for automated agents operating under real–time constraints.

Large–scale systems present particular difficulties both with respect to simulation and control, and special difficulties arise when the plant to be controlled undergoes structural changes.

This thesis deals with various aspects of mixed quantitative and qualitative information processing, and tackles in particular problems related to fault monitoring, detection, characterization, isolation, and identification in large–scale systems. It addresses a number of the aforementioned difficulties, and contributes to advance the state–of–the–art in their treatment.

Problems of information overload on human operators of complex large–scale systems have been studied during the past few years mainly due to major accidents of aircrafts and in chemical, electrical, and industrial plants. Quite often, pilots/operators of such plants have had to work under immense psychological stress in situations where the onslaught of masses of unstructured information mixed with a multitude of alarms that all seemed to go off simultaneously, made an assessment of the situation under real–time constraints impossible, thereby preventing them from taking appropriate action, and yet, the consequences of a misjudgment were, at least potentially, catastrophic.

Does automatic feedback control provide us with a panacea for solving the "human overload" problem? Indeed, automatic controllers are not known to

be subject to psychological stress phenomena. Yet, the same generic difficulty presents itself in automatic feedback control in a different form. Any decision–making (or control, which is the same thing) activity invariably involves solving an optimization problem. The more input variables this optimization problem contains, the higher is the dimensionality of the search space, in which the optimal solution must be found. Human plant operators are kept aware of real–time constraints, reaching a decision always in time, yet possibly making a decision that may turn out to be inappropriate since the operator didn't have enough time to explore the entire search space, thereby overlooking both better solutions and undesired side effects of the one finally selected. Automatic controllers are usually unaware of the passing of time. Hence they can be made to always reach the optimal solution, however, the process may take too long for this solution to be of any practical use. On the other hand, if a controller is being equipped with a sense of time passing, it also must limit its search, and will then begin to make the same mistakes that human operators are bound to make. Such controllers will suffer from "stress syndromes," just like any human operator would. After all, "awareness of time running out" is one of the important human "stressors."

Thus, human operators and automatic controllers need to be advised by "intelligent supervision, and/or control, and/or decision–support systems." These intelligent systems are the topic of this dissertation. They are proposed to serve as tools that may help improve the decision–making process of human operators and/or automatic controllers alike. Their basic task must be to prevent the (human or automated) decision makers from committing errors and/or from misjudging the current situation, by providing them with additional quantitative and qualitative information that can be used in the decision–making process, for detecting and discriminating faults at an as early time as possible, and for dealing with developing emergencies in an informed fashion. In particular, qualitative information has proven to be very useful in large–scale systems for discerning what is really going on, for deciding the state in which the system is at any point in time, and for assessing what would be the consequences of taking or not taking a proposed emergency action.

Following the natural order of the research, this thesis is divided in three main sections: a) Introduction, b) Methodology, and c) Applications. Each section is composed of several chapters.

The **introductory section** is composed of Chapters 1 and 2. This is where the problem under study and its main characteristics are defined, the use of qualitative methodologies for tackling it is justified, the application domains are delineated, and a comparison of the different qualitative modeling and simulation techniques is performed including those used in the dissertation.

In the first chapter, some important concepts were introduced, such as *Large–Scale System*, *Variable Structure System*, *Intelligent Supervision and Control System*, and the paradox of human *vs.* automatic control, with the intention of establishing the problem of *information overload* in large–scale systems, which has been at the origin of this research effort. The true causes of this problem and its possible solutions were stated from both the human and automatic control perspectives, with emphasis on the combination of quantitative and qualitative methodologies to solve it. The contributions of this thesis to tackling this important problem were also stated in this chapter.

Chapter 2 presents a state–of–the–art survey of fault detection and troubleshooting of dynamic systems mentioning both the quantitative and qualitative approaches. This survey provides an insight into the different methodologies (from pure diagnosis to residual generation methods in the quantitative case, and from expert systems to model–based deep reasoners in the qualitative case) used for designing *Fault Monitoring Systems*, and a comparison of their advantages and disadvantages when applied to non–trivial dynamic systems. In this way, the adequacy of employing qualitative methodologies to fault detection and troubleshooting is substantiated. This concludes the state–of–the–art survey of the *status quo ante*. All subsequent sections describe new results obtained in this research effort.

The **methodological section** is comprised of Chapters 3 and 5, describing *Fuzzy Inductive Reasoning* and *Reconstruction Analysis* respectively, the two methodological tools used in this thesis. Fuzzy Inductive Reasoning alone suffices to deal with problems of fault detection and troubleshooting in small– and medium–sized systems, but is incapable of tackling the information overload problem indigenous to large–scale systems. Reconstruction Analysis will be needed as an additional tool when dealing with fault detection and troubleshooting in large–scale systems, precisely for addressing the information overload problem. It seemed more appealing to keep in this thesis the same order as it was followed chronologically during the research, that is, to demonstrate the utilization of Fuzzy Inductive Reasoning in qualitative modeling and simulation as well as fault detection and troubleshooting in small– and medium–sized systems prior to introducing the additional methodological tool of the Reconstruction Analysis. This is why the two methodological chapters are not consecutive.

Chapter 3 focuses on *Fuzzy Inductive Reasoning (FIR)*, providing full details of the development and implementation of this methodological tool. To this end, the chapter starts out with the foundations of the methodology that are rooted in *General Systems Theory.* It is shown how FIR can be used for qualitatively modeling and simulating continuous–time systems by means of an example (a third order linear system) that will be carried through all the phases of FIR modeling. In this chapter, the basis for a combined quantitative and qualitative modeling and simulation methodology using FIR is also provided. The chapter ends with an example of an application to mixed quantitative and qualitative modeling and simulation of a dynamic system (a hydraulic motor with a four–way servo valve). The applicability of the proposed approach to mixed quantitative and qualitative modeling and simulation is demonstrated by comparing the results of the mixed simulation with those of a purely quantitative simulation of the same system.

Chapter 5 focuses on *Reconstruction Analysis (RA)*. As in Chapter 3, the technique is explained in full, starting out from its methodological roots in *General Systems Theory*. An example is carried on through the chapter in order to show the capabilities of RA to perform a causal and temporal analysis on a set of behavior variables. In the second part of this chapter, the three refinement algorithms of the Optimal Structure Analysis are applied to the same example (in the crisp and fuzzy cases), and their results are compared.

The **applications section** is focused on the development of a qualitative Fault Monitoring System and its application to real–world continuous–time large–scale engineering processes. This section is composed of Chapters 4, 6, and 7.

In Chapter 4, entitled "Qualitative Fault Monitoring," the necessary mechanisms used for designing a Fault Monitoring Systems are presented. The processes of causal grouping of variables, the generation of hierarchies of inductive subsystems, as well as the concepts of fault detection, identification, characterization, isolation, and diagnosis, are explained in full. Three different operating modes for the Fault Monitoring System are proposed, including one that can also be used for Variable Structure Systems. The operating modes are: *Back to Training Mode*, *Qualitative Models Library*, and *Forecasting All Possible Structures*. In the second part of this chapter, the Qualitative Models Library operating mode is applied to a Boeing 747 aircraft model to demonstrate its capabilities. This example also demonstrates the enhanced discriminatory power and improved forecasting capability of a fuzzy inductive reasoner over a crisp inductive reasoner. In the third part of this chapter, the Forecasting All Possible Structures operating mode is applied to the problem of structure identification in Variable Structure Systems. Two examples, a fairly simple two–water–tank system and a rather involved electric circuit example are included to demonstrate the detection, discrimination, and identification of structural changes.

Chapter 6 is focused on the selection and causal grouping of variables problem. To this end, heuristic recipes used to deal with the large number of subsystems that result from the application of the *Optimal Structure*

*Analysis* technique to a large–scale system are presented. The second part of this chapter presents a comparison, from the point of view of their forecasting capabilities, between the *Optimal Mask Analysis* used by the Fuzzy Inductive Reasoning methodology to obtain qualitative models, and the *Optimal Structure Analysis* used by the Reconstruction Analysis methodology to obtain subsets of related variables that can be treated as subsystems. This comparison is made by applying both techniques to the generic third–order linear system model shown in Chapter 3 and to the Boeing 747 aircraft model that had been introduced earlier in Chapter 4.

Chapter 7 builds upon what has been developed in previous chapters, i.e., the Fuzzy Inductive Reasoning methodology (Chapter 3), the development of a Fault Monitoring System (Chapter 4), the Reconstruction Analysis methodology (Chapter 5), and the heuristic recipes for the selection and causal grouping of variables (Chapter 6). In the first part of this chapter, a combined Fuzzy Inductive Reasoning/Reconstruction Analysis (FIR/RA) methodology is proposed, and its advantages for fault detection and troubleshooting in large–scale systems are explained. In the second part of the chapter, this combined FIR/RA Fault Monitoring System is applied to a sophisticated large–scale system model, namely, a boiling water nuclear reactor model.

Finally, Chapter 8 provides a summary of the obtained results, and presents a list of open problems and possible future research efforts extending the work presented in this thesis.

# Index

# List of Figures

# Chapter 1

# Introduction

Over the past few decades, our high–technology society has developed technological processes of ever increasing complexity. The complexity of a process can be measured either in terms of the number of its components, or in terms of the number of interactions between these components. Using either of these two measures, the average complexity of equipment in use has grown consistently over time with a faster than proportional rate. Safety requirements as well as the high value of many of the processes in use have called for increasingly elaborate tools for automated plant monitoring, maintenance, and testing for improving reliability and robustness, and for reducing the periods during which the equipment is unavailable due to failure. Such tools have become even more important as the risk of failure grows at least quadratically with the complexity of a process [Motaabbed, 1992].

For the above reasons, research in the field of automated plant supervision, fault detection, and fault diagnosis of industrial processes has experienced a spectacular growth during the past 25 years. In the beginning, only analytical techniques were used for such purposes, but in the sequel, qualitative approaches have begun to play an ever more important role in all of these activities, partly employing techniques of qualitative modeling and simulation, spin–offs of astounding advances made in the fields of Artificial Intelligence, Qualitative Reasoning, Fuzzy Logic, and the like.

The use of techniques derived from Artificial Intelligence and other fields of symbolic knowledge processing in control, supervision, and diagnosis of complex large–scale systems has influenced in major ways the development of areas such as Intelligent Control, Intelligent Supervision, Intelligent Troubleshooting, High–Autonomy Systems, Fault–Tolerant Control, Self–Aware Control, Cognizant Control, Qualitative Modeling and Simulation, Computer–Assisted Decision Making, etc. [Cellier *et al.*, 1992a]. A fundamental concept uniting all these research fields is their use of a combination of quantitative and qualitative reasoning mechanisms for capturing and mimicking human assessment processes when dealing with tasks of control, supervision, and diagnosis of complex dynamic processes.

One serious problem that has haunted control engineers and artificial intelligence researchers alike from the early days on is the problem of *information overload*. Human plant controllers are easily overwhelmed by the sheer mass of information available to them for decision making in any real–time situation, but the same is true for automated agents mimicking the human decision making processes. Is there a preference of one over the other? There seems to exist a paradox relating to the assessment of the relative merits of human *vs.* automatic controllers. Large–scale systems present particular difficulties both with respect to simulation and control, and special difficulties arise when the plant to be controlled undergoes structural changes.

This thesis deals with various aspects of mixed quantitative and qualitative information processing, and tackles in particular problems related to fault monitoring, detection, characterization, isolation, and identification in large–scale systems. It addresses a number of the aforementioned difficulties, and contributes to advance the state–of–the–art in their treatment. This introductory chapter properly defines the aforementioned problems, and intends to provide the reader with a clear idea of the motivations, scope, and objectives of this dissertation.

## 1.1 Information Overload in Large–Scale Systems

Large–Scale Systems can be defined as systems with a high degree of complexity in the spatial and temporal relations between the subsystems that they are composed of, which, in turn, can be treated as independent systems [Kuipers, 1989b]. A consequence of this definition is that a large–scale system needs to be divided into its subsystems in order to be represented, modeled, simulated, or controlled [Puccia and Levins, 1985]. Frequently, several of these subsystems are characterized by highly non–linear behavior. Typical examples of large–scale systems can be found in both the "hard" and "soft" sciences. An electric power plant, a chemical plant, a modern aircraft, a spacecraft, and any complex industrial, defense, or transportation system are examples of large–scale systems in "hard" sciences. Biomedical systems, the evolution of economics, and social system components, such as capital investment, national debt, the development of the gross national product, the welfare system, the educational system, immigration patterns, etc. are all examples of large–scale systems in the "soft" sciences. The latter are beyond the scope of this dissertation.

Operators of complex large–scale systems are confronted with an overwhelming amount of information during normal and abnormal situations. Batteries of operator consoles, flashing lights, buzzers, page–long lists of log values ejected by line printers, and a variety of other signals have to be simultaneously assimilated and reacted to. It is impossible for a human operator to maintain a complete picture of all events and actions that may influence the situation at hand. The human task to be performed is extremely difficult and stressful. The higher the complexity of the plant, the larger will be the volume of incoming sensory data, and consequently, the more will the operator be stressed. Even people who are well trained to perform this line of work are destined to make mistakes under the described conditions. The more crucial it is that their decisions are indeed correct, i.e., if human lives are at stake, such as in the case of an air traffic controller or in the case of an operator of a nuclear power plant, the more will the operator be under stress, and the more likely will it be that he or she eventually makes a fatal mistake.

Problems of information overload on human operators of complex large–scale systems, generally known as *"human overload,"* have been studied during the past few years mainly due to major accidents of aircrafts and in chemical, electrical, and industrial plants [Puccia and Levins, 1985]. Quite often, pilots/operators of such plants have had to work under immense psychological stress in situations where the available information was not enough to provide even minimal control, and where the consequences of a mistake were catastrophic. For example, the cockpit of a commercial aircraft contains more than 1000 different sensors and controls. Similarly, the operating room of a nuclear power plant is equipped with thousands of plant status indicators (sensors), and the operators can influence the behavior of the plant by means of hundreds of different plant set point selectors (actuators). It is not reasonable to assume that, in an emergency situation, a small number of human pilots/operators would be capable of reliably monitoring all of these sensors and manually operating all of these controls properly and adequately without any additional help [Dvorak and Kuipers, 1989].

Psychological tests have revealed that the average human, after being presented with a number of facts such as given in a news broadcast, can recall approximately 10 of these facts from short–term memory when asked to remember what had been said. While there exists a noticeable variation in individual human capabilities, psychologists tend to agree that most humans cannot reliably and safely tend to more than 10 different items at a time before they begin to misjudge some of the circumstances and make serious mistakes. Consequently, it makes little sense to provide the operator of a complex plant with hundreds of status indicators simultaneously, and expect him or her to monitor them reliably and react to them adequately. In the case of a major emergency, it is very likely that many subsystems will signal problems almost instantaneously, and it will be very difficult for the human operators to discern the true causes from their consequences, i.e., to know which subsystem experienced problems first [Pan, 1986].

Of course, complex large–scale systems are not controlled only by human operators. They usually also include automatic controllers to take care of activities that human operators are not capable of handling, or that must be automated to relieve them of routine tasks, letting them concentrate on the more important problems such as the analysis of abnormal plant behavior. It is a well–known fact that human operators are much more capable than any of today's automatic controllers to deal reliably and efficiently with unanticipated problems and unforeseen emergencies in complex large–scale systems, but at the same time, human operators constitute the least predictable and therefore the least reliable element of the overall control system [Cacciabue *et al.*, 1990]. Since human operators are normally responsible for safety and emergency procedures, any help that can be provided by any means that will improve the decision–making processes of the human operators and/or automatic controllers must be welcomed, particularly in the context of fault detection and troubleshooting.

### 1.1.1 Human Control *vs.* Automatic Control Paradox

What role have symbolic and/or qualitative reasoning tools to play in control applications that call for the maximum precision that the controller can possible achieve? Before answering this question, it is useful to state and discuss the human *vs.* automatic control paradox. Automatic controllers are well suited for carrying out complex routine operations, but have difficulties when confronted with abnormal behavior and/or unforeseen situations. In contrast, human controllers have a high probability of eventually committing a mistake when asked to perform repeated routine tasks, but they are excellently equipped for dealing with emergencies and/or unforeseen events. In some sense, it can be said that human controllers are needed to prevent possible failures of automatic controllers; yet that automatic controllers are needed to prevent human controllers from making mistakes. Most human operators of large–scale systems believe that their most important task is to fend against controller error, yet at the same time, one of the most important considerations to be taken into account by designers of automatic controllers is the prevention of human mistakes. The goal is then to provide the human controller with tools that will improve his or her decision making processes, while providing the automatic controllers with some of the human reasoning capabilities, in order to enable them to recognize human mistakes as they are made, and in order to enable them to communicate with the human plant operators in terms that are understandable to them. Hence it is an extremely useful and important facet of modern large–scale system control software to be able to pair quantitative with qualitative modes of reasoning, to combine precision with vagueness, to react to available quantitative information with swiftness and certainty yet being able to also reach decisions under uncertainty.

Since human controllers are incapable of solving sets of differential equations in their heads, i.e., cannot perform quantitative reasoning to a meaningful extent, why then should automatic controllers be able to perform such tasks? The objective of attempting to formulate decision tools with the type of precision usually associated with classical control techniques could be neither attainable nor necessary. By permitting a certain amount of imprecision in the reasoning processes, robustness is provided that allows to control and supervise complex situations that might not otherwise be controllable, and also a means is offered that allows to account for the vagueness inherent in human control input reflecting the way in which humans conceptualize the world.

The role of human operators inside the overall control strategy depends on the safety decisions taken. The plant management must select between two basic approaches to large–scale system control:

a) Human operators are in charge of almost every action. Automatic controllers are used to take care of (i) routine tasks that are not safety critical, (ii) emergency procedures that call for an extremely fast initial reaction, (iii) standard supervision and monitoring functions, and –most importantly of all– (iv) providing the human operators with quantitative and qualitative information to support their decision–making.

b) Automatic controllers perform the real control of the system while providing human operators with high–level qualitative and quantitative information. Human operators are needed (i) for global plant supervision, (ii) to select the set points of the automated controllers, (iii) to take care of unforeseen situations, and (iv) to decide on ultimate safety and emergency procedures.

Neither is approach (a) better than approach (b), nor is (b) better than (a). Approach (a) is widely used in the United States and its influence area, whereas approach (b) is mostly used in Europe [Cacciabue *et al.*, 1990]. The decision, which of these two approaches to adopt in any given situation, is usually based more on political and/or psychological considerations than substantiated by technical insight.

For example, the U.S. space program is clearly following approach (a) to an extreme extent. Consequently, the Space Shuttle is equipped with more than 5000 sensory readout functions, and every single actuator is equipped with a manual override capability, which, on technical grounds, makes fairly little sense since, in an emergency situation, the crew of five or seven astronauts is ill equipped to operate the spacecraft safely and reliably. On the other hand, the Russian space program is an extreme example of approach (b). Russian manned space missions were always carried out in spacecrafts that were fully automated with hardly any manual override capabilities. The cosmonauts are merely passengers of their craft until they reach orbit. Again, this makes little sense on technical grounds since, indeed, cosmonauts become helpless victims if anything should ever go wrong. The example shows that the two approaches (a) and (b) outlined above are not really two alternatives, but only the two extremes in a continuous spectrum of intermediate options. It is hoped that, in the future, technical insight may play a more prominent role in large–scale system design, and that more organically interwoven overall system architectures can be found that support a well–integrated cooperation between human plant operators and automated plant controllers.

Does automatic feedback control provide us with a panacea for solving the "human overload" problem? Indeed, automatic controllers are not known to be subject to psychological stress phenomena. Yet, the same generic difficulty presents itself in automatic feedback control in a different form. Any decision–making (or control, which is the same thing) activity invariably involves solving an optimization problem. The more input variables this optimization problem contains, the higher is the dimensionality of the search space, in which the optimal solution must be found. Human plant operators are kept aware of real–time constraints, reaching a decision always in time, yet possibly making a decision that may turn out to be inappropriate since the operator didn't have enough time to explore the entire search space, thereby overlooking both better solutions and undesired side effects of the one finally selected. Automatic controllers are usually unaware of the passing of time. Hence they can be made to always reach the optimal solution, however, the process may take too long for this solution to be of any practical use. On the other hand, if a controller is being equipped with a sense of time passing, it also must limit its search, and will then begin to make the same mistakes that human operators are bound to make. Such controllers will suffer from "stress syndromes," just like any human operator would. After all, "awareness of time running out" is one of the important human "stressors."

Thus, either of the two options presented above requires human operators and/or automatic controllers to be advised by "intelligent supervision, and/or control, and/or decision–support systems." These intelligent systems are the topic of this dissertation. They are proposed to serve as tools that may help improve the decision–making process of human operators and/or automatic controllers alike. Their basic task must be to prevent the (human or automated) decision makers from committing errors and/or from misjudging the current situation, by providing them with additional quantitative and qualitative information that can be used in the decision–making process, for detecting and discriminating faults at an as early time as possible, and for dealing with developing emergencies in an informed fashion. In particular, qualitative information has proven to be very useful in large–scale systems for discerning what is really going on, for deciding the state in which the system is at any point in time, and for assessing what would be the consequences of taking or not taking a proposed emergency action.

### 1.1.2 Difficulties with Large–Scale Systems

The main problems encountered when dealing with large–scale systems from a simulation and supervision perspective are [Berkan *et al.*, 1991]:

- The great number of variables and subsystems. This fact makes the identification of a minimum set of meaningful variables to work with a difficult task and consequently, impedes the modeling, simulation, and control processes.
- The number of causal and temporal relations among the variables and subsystems. Causal and temporal analyses must be performed in order to obtain behavior patterns and structure of the analyzed system.

- The overwhelming amount of information received by the human operators and/or automatic controllers.

- The highly non–linear characteristics of most of these systems. This non–linearity becomes an insuperable obstacle for the quantitative modeling and simulation of these systems, making the application of traditional control and troubleshooting techniques impossible.

- The physical limitations in exciting the system under study (and often even a quantitative model thereof) for proper characterization of all of its possible modes of behavior.

- The modeling and simulation of these systems under a given real–time constraint.

- The variable structure of some subsystems due to faults or transients taking place. Most numerical models are not capable at all of dealing with systems in which a fault produces a change in the system's structure.

Many of these difficulties will be revisited along this thesis, but particularly the variable structure problem will easily turn out to be one of the most important and difficult problems to be solved on the way of designing a qualitative supervision and decision–support system, since any fault can be viewed as a structural change.

### 1.1.3 Variable Structure Systems

Variable structure systems and controllers associated with them have originally been mostly studied by Soviet researchers [Emelyanov, 1970; Utkin, 1977; Utkin, 1984], but lately, an interest in these systems has also been expressed by researchers in the West [Slotine, 1984; Sira, 1990; Hung *et al.*, 1993)]. In this dissertation, a somewhat specific definition of Variable Structure Systems (VSSs) will be used: A VSS is a system in which the computational causality of one or several laws governing the behavior of that system changes as a consequence of a change in the value of a boolean variable in the model [Elmqvist *et al.*, 1993]. Such systems present serious difficulties to both simulation and control.

From the point of view of simulation, difficulties are caused precisely by the changing computational causalities. In the past, different programs were usually written, one for each of the structures of the VSS, and a mechanism was encoded to switch at run–time from one model to another. An alternative solution was recently proposed by [Elmqvist *et al.*, 1993] that allows to encode a single simulation program that encompasses and is valid for all different structures within the VSS. Another possibility, that will be discussed later in this dissertation, is to write different programs, one for each of the structures of the VSS, and simulate with all of them in parallel [de Albornoz *et al.*, 1994].

From the point of view of control, difficulties are caused by the abrupt change in the system structure. Even if the controller is switched at the same time as the plant, the control system nevertheless experiences a shock that may lead to undesirable transient behavior. Such shocks need to be dampened out either by means of quite complex and costly non–linear compensation algorithms, or by means of a geometrical approach that considers each transition from one structure to another as a finite surface along which the system is to slide. This approach is known as Sliding Motion Control [Sira, 1990].

In a VSS, the most important events and consequently the most important things to know are, on the one hand, when a transition from one structural mode to another takes place and, on the other hand, to which other mode (or structure) the system changes at this point in time. These questions are not always easily decidable, since the switching condition may itself be internal to the system or to its model, or can be produced by a malfunction or a transient. In this dissertation, two kinds of VSSs will be treated; the ones in which structural changes occur normally and regularly, and the ones in which a structural change can be considered a fault or abnormal event.

## 1.2 Motivation, Aims, and Scope

Artificial Intelligence has the purpose of reproducing facets of human mental reasoning processes to build "intelligent" and autonomous systems [Charniak and McDermott, 1987]. The research in this area has proceeded along multiple lines. Some of these efforts are considered "pure," such as the design of knowledge representation schemes, most of the work in machine learning, the development of new search strategies, automated reasoning techniques, problem–solving tools, etc. [Newell and Simon, 1972]; some others are considered "applied," such as the design of knowledge–based (expert) systems and connectionist approaches to inductive modeling [Firebaugh, 1988]. Whereas pure Artificial Intelligence research ponders in a general and necessarily somewhat vague fashion about what "intelligence" really entails, applied Artificial Intelligence research must face real–world problems, and therefore takes a much more modest approach when looking at the problem of intelligence, restricting its view to important facets of the specific application at hand. What counts is the ability to build an automated reasoning system that is capable of reproducing convincingly the fashion in which a human would approach and reason about the facts he or she is being presented with, and not a deeper understanding of the mechanisms of intelligence *per se*.

This dissertation clearly belongs to the area of applied Artificial Intelligence. It tackles real–world engineering problems and provides some answers as to how they can successfully be dealt with. **The scope of the dissertation is limited to the intersection between applied Artificial Intelligence and the problem of troubleshooting complex engineering systems.**

It was mentioned that a human operator is capable of controlling a complex system in a much better way than any of today's automatic controllers. In the decision–making process, human operators use a combination of quantitative and qualitative information to perform a basically qualitative and suboptimal reasoning process about the behavior of the system without solving any kind of algebraic or differential equations, and, in some cases, even without knowing the system's internal structure. Their whole reasoning process is based predominantly on input/output behavioral patterns, prior experience, and common sense [Carbonell and Minton, 1985]. The primary purpose of combining Artificial Intelligence with engineering approaches to troubleshooting is to provide automated supervision and control systems with the necessary capabilities to replicate the aforementioned processes of qualitative reasoning based precisely on input/output behavioral patterns, prior experience, and common sense.

"Intelligent systems," such as the aforementioned ones, necessarily operate in a discrete and suboptimal way using a symbolic knowledge representation, that is, as qualitative systems. Normally, intelligent systems must interact with the real physical world that is based on continuous time and space in the same way that human operators do, or cooperate with lower–level controllers that are realized by means of more classical quantitative signal and system representations. The interaction between these two types of systems calls for a mixed quantitative and qualitative modeling and simulation paradigm that is able to compensate for the shortcomings inherent in each of the two types of knowledge representation schemes, and may help to solve problems that are beyond the capabilities of either of these two methodologies alone. **The first objective of this research effort is the development of a combined quantitative/qualitative modeling and simulation methodology to be applied to continuous–time dynamic processes.**

A combined Artificial Intelligence and numerical technique may be able to tackle problems integrating discrete and qualitative symbolic problem–solving with continuous and quantitative perception and actuation. Furthermore, the application of Artificial Intelligence tools such as knowledge–based systems and artificial neural networks, implementing Artificial Intelligence techniques such as genetic algorithms, inductive learning, and qualitative simulation, to real–world engineering problems, such as fault detection and troubleshooting in large–scale systems, helps to exhibit their grade of maturity and assess their real capabilities.

**The motivation of this thesis is to help bridge the gap between the two worlds of quantitative computation and qualitative reasoning. The second (and most important) objective is then the development and application of qualitative methodologies to solve information overload problems during normal and abnormal operation of quantitative complex large–scale systems.** Components of these methodologies stem from the worlds of *Artificial Intelligence* (Qualitative Simulation and Model–Based Reasoning), of *General System Theory* (Reconstruction Analysis and Fuzzy Inductive Reasoning), and of *Fault Diagnosis Theory* (Fault Detection and Troubleshooting). The resultant methodology can best be understood as a tool for qualitative modeling and simulation of continuous–time processes capable to help in the detection, isolation, characterization, identification, analysis, and possibly even prevention of faults.

To this end, the qualitative model will be trained to determine when a malfunction occurs in the quantitative model, it will hypothesize about the nature of this malfunction, and finally, it will suggest a global control strategy that would allow to safely operate the quantitative model under the modified structural conditions. **The third objective is thus the development of a Fault Monitoring System based on the aforementioned combined quantitative and qualitative methodology.**

## 1.3 Structure of this Thesis

Following the natural order of the research, this thesis is divided in three main sections: a) Introduction, b) Methodology, and c) Applications. Each section is composed of several chapters.

The **introductory section** is composed of Chapters 1 and 2. This is where the problem under study and its main characteristics are defined, the use of qualitative methodologies for tackling it is justified, the application domains are delineated, and a comparison of the different qualitative modeling and simulation techniques is performed including those used in the dissertation.

In the first chapter, some important concepts were introduced, such as *Large–Scale System*, *Variable Structure System*, *Intelligent Supervision and Control System*, and the paradox of human *vs.* automatic control, with the intention of establishing the problem of *information overload* in large–scale systems, which has been at the origin of this research effort. The true causes of this problem and its possible solutions were stated from both the human and automatic control perspectives, with emphasis on the combination of quantitative and qualitative methodologies to solve it. The contributions of this thesis to tackling this important problem were also stated in this chapter.

Chapter 2 presents a state–of–the–art survey of fault detection and troubleshooting of dynamic systems mentioning both the quantitative and qualitative approaches. This survey provides an insight into the different methodologies (from pure diagnosis to residual generation methods in the quantitative case, and from expert systems to model–based deep reasoners in the qualitative case) used for designing *Fault Monitoring Systems*, and a comparison of their advantages and disadvantages when applied to non–trivial dynamic systems. In this way, the adequacy of employing qualitative methodologies to fault detection and troubleshooting is substantiated. This concludes the state–of–the–art survey of the *status quo ante*. All subsequent sections describe new results obtained in this research effort.

The **methodological section** is comprised of Chapters 3 and 5, describing *Fuzzy Inductive Reasoning* and *Reconstruction Analysis* respectively, the two methodological tools used in this thesis. Fuzzy Inductive Reasoning alone suffices to deal with problems of fault detection and troubleshooting in small– and medium–sized systems, but is incapable of tackling the information overload problem indigenous to large–scale systems. Reconstruction Analysis will be needed as an additional tool when dealing with fault detection and troubleshooting in large–scale systems, precisely for addressing the information overload problem. It seemed more appealing to keep in this thesis the same order as it was followed chronologically during the research, that is, to demonstrate the utilization of Fuzzy Inductive Reasoning in qualitative modeling and simulation as well as fault detection and troubleshooting in small– and medium–sized systems prior to introducing the additional methodological tool of the Reconstruction Analysis. This is why the two methodological chapters are not consecutive.

Chapter 3 focuses on *Fuzzy Inductive Reasoning (FIR)*, providing full details of the development and implementation of this methodological tool.[1] To this end, the chapter starts out with the foundations of the methodology that are rooted in *General Systems Theory*. It is shown how FIR can be used for qualitatively modeling and simulating continuous–time systems by means of an example (a third order linear system) that will be carried through all the phases of FIR modeling. In this chapter, the basis for a combined quantitative and qualitative modeling and simulation methodology using FIR is also provided. The chapter ends with an example of an application to mixed quantitative and qualitative modeling and simulation of a dynamic system (a hydraulic motor with a four–way servo valve). The applicability of the proposed approach to mixed quantitative and qualitative modeling and simulation is demonstrated by comparing the results of the mixed simulation with those of a purely quantitative simulation of the same system [Cellier *et al.*, 1992, 1996a].

Chapter 5 focuses on *Reconstruction Analysis (RA)*. As in Chapter 3, the technique is explained in full, starting out from its methodological roots in *General Systems Theory* [Cellier *et al.*, 1996b]. An example is carried on through the chapter in order to show the capabilities of RA to perform a causal and temporal analysis on a set of behavior variables. In the second part of this chapter, the three refinement algorithms of the Optimal Structure Analysis are applied to the same example (in the crisp and fuzzy cases), and their results are compared.

The **applications section** is focused on the development of a qualitative Fault Monitoring System and its application to real–world continuous–time large–scale engineering processes. This section is composed of Chapters 4, 6, and 7.

In Chapter 4, entitled "Qualitative Fault Monitoring," the necessary mechanisms used for designing a Fault Monitoring Systems are presented. The processes of causal grouping of variables, the generation of hierarchies of inductive subsystems, as well as the concepts of fault detection, identification, characterization, isolation, and diagnosis, are explained in full. Three different operating modes for the Fault Monitoring System are proposed, including one that can also be used for Variable Structure Systems. The operating modes are: *Back to Training Mode*, *Qualitative Models Library*, and *Forecasting All Possible Structures*. In the second part of this chapter, the Qualitative Models Library operating mode is applied to a Boeing 747 aircraft model to demonstrate its capabilities. This example also demonstrates the enhanced

[1]The research on Fuzzy Inductive Reasoning and the development of a combined quantitative/qualitative modeling and simulation tool were carried out by the author of this dissertation and two more Ph.D. students whose names, and complete references, are given in the Introduction section of Chapter 3.

discriminatory power and improved forecasting capability of a fuzzy inductive reasoner [de Albornoz and Cellier, 1993a, 1994], over a crisp inductive reasoner [Vesanterä, 1988; Vesanterä and Cellier, 1989]. In the third part of this chapter, the Forecasting All Possible Structures operating mode is applied to the problem of structure identification in Variable Structure Systems. Two examples, a fairly simple two–water–tank system and a rather involved electric circuit example are included to demonstrate the detection, discrimination, and identification of structural changes [de Albornoz *et al.*, 1994, 1996].

Chapter 6 is focused on the selection and causal grouping of variables problem. To this end, heuristic recipes used to deal with the large number of subsystems that result from the application of the *Optimal Structure Analysis* technique to a large–scale system are presented. The second part of this chapter presents a comparison, from the point of view of their forecasting capabilities, between the *Optimal Mask Analysis* used by the Fuzzy Inductive Reasoning methodology to obtain qualitative models, and the *Optimal Structure Analysis* used by the Reconstruction Analysis methodology to obtain subsets of related variables that can be treated as subsystems. This comparison is made by applying both techniques to the generic third–order linear system model shown in Chapter 3 and to the Boeing 747 aircraft model that had been introduced earlier in Chapter 4 [de Albornoz and Cellier, 1996].

Chapter 7 builds upon what has been developed in previous chapters, i.e., the Fuzzy Inductive Reasoning methodology (Chapter 3), the development of a Fault Monitoring System (Chapter 4), the Reconstruction Analysis methodology (Chapter 5), and the heuristic recipes for the selection and causal grouping of variables (Chapter 6). In the first part of this chapter, a combined Fuzzy Inductive Reasoning/Reconstruction Analysis (FIR/RA) methodology is proposed, and its advantages for fault detection and troubleshooting in large–scale systems are explained. In the second part of the chapter, this combined FIR/RA Fault Monitoring System is applied to a sophisticated large–scale system model, namely, a boiling water nuclear reactor model [de Albornoz and Cellier, 1993b; de Albornoz *et al.*, 1996b].

Finally, Chapter 8 provides a summary of the obtained results, and presents a list of open problems and possible future research efforts extending the work presented in this thesis.

This thesis ends with an extensive bibliography of references used in this research effort. The publications that were derived from this thesis are concentrated in a separate section placed at the beginning of the bibliography.

# Chapter 2

# Fault Detection and Troubleshooting: A Comparative Study

## 2.1 Introduction

Automatic and human controlled systems are becoming more and more complex and their control algorithms more and more sophisticated. Since the probability of faults in a system grows at least quadratically with its complexity, the average time between faults gets shorter and shorter. Consequently, there is a growing demand for fault tolerance, which can be achieved by means of the use of *Fault Monitoring Systems* (FMS). The basic functions of a FMS have always been to register an alarm when an abnormal condition develops in the monitored system, as well as to identify the faulty component. However, the demands put on FMS schemes have also grown over time. Modern FMSs are supposed to be capable of detecting, isolating, identifying, diagnosing, and analyzing anomalous behavior.

Over the past two decades, research on fault detection and troubleshooting of dynamic systems has gained increasing consideration. This development has been stimulated by three major factors:

1. The trend of automation towards increased complexity, together with the demand that these more complex systems be equally *reliable*, i.e., equally safe, secure, and available, where:

   - *safety* relates to system behavior under internal faults, i.e., if and when an irrecoverable fault occurs inside the system, the FMS should perform a graceful shutdown of the system, preventing the system from suffering any permanent damage, and ensuring that the system does not cause any hazard to the operating crew or other humans in its environment;
   - *security* relates to system behavior under external faults, i.e., if the system operator misjudges the situation and issues a potentially harmful command, the FMS should prevent the system from reacting to that command in hazardous ways; and
   - *availability* relates to the percentage of time that the system is down for maintenance or repair. This factor is evidently closely related to the average time between failures.

2. The powerful control techniques introduced by Modern Control Theory make the task of fault monitoring increasingly difficult. The problem is the following. Beside from the controller's main duties of ensuring the plant's *stability* and *accuracy*, it is an important task of any controller to desensitize the system's behavior to uncontrollable external factors (disturbance suppression) as well as uncontrollable internal factors (parameter variability and aging). Evidently, the controller has no way of distinguishing between these factors and minor anomalies that the FMS should know about. Thus, the control system has a tendency to reduce the measurable effects of minor faults, thereby counteracting the efforts of the FMS for fault detection. The better the controller works, the more difficult will be the early detection of faults, unless they are of the catastrophic kind.

3. Techniques derived from Artificial Intelligence such as knowledge–based systems, connectionist systems, and qualitative model–based deep reasoners have provided system engineers with an entire palette of new and powerful tools that may sharpen the FMS's sensitivity and resolution power, hence offering means to provide better and more reliable fault monitoring of ever more complex plants, compensating for the above–mentioned difficulties that were caused by improved control technology.

Fault detection in dynamic systems has traditionally been performed using *limit–value detectors* and *hardware redundancy* [Pau, 1981]. In the first case, measurable variables are directly checked for upper or lower transgression of fixed limit or setpoint values. As long as all of the monitored measuring variables are within their assigned ranges, the system is assumed sane. As soon as one of the variables leaves its range, an alarm is set off indicating the occurrence of an anomaly in the system. In the second case, the most *central* components of the system, i.e., those parts the failure of which would be most catastrophic, such as the central controlling computer, as well as the most *vulnerable* components, i.e., those parts that are most likely to fail, are duplicated or even triplicated. Duplication allows to set off an alarm whenever the two devices placed in parallel disagree on their outputs; triplication allows to continue in a safe way after breakdown of a component by employing a voting scheme. If two of the three devices agree on their outputs, it is assumed that they are right, whereas the third device is assumed faulty and will henceforth be ignored. The human operators are then alerted to the fact and are asked to replace the faulty component, after which time the system will resume its former voting scheme. These approaches to fault monitoring are quite simple and in many situations reasonably straightforward to apply, and are thus widely used. They are essential in the control of aircrafts, spacecrafts, electric power plants, nuclear reactors, and generally, in all safety–critical processes with a sufficiently high degree of complexity.

Recently, new quantitative and qualitative approaches to fault monitoring have emerged based on the idea that several dissimilar sensors measuring different variables and therefore producing entirely different signals, can be used in a comparison scheme more sophisticated than simple limit–value or majority–vote logic to detect and isolate a fault [Frank, 1990]. The basis for this idea is that even though the sensors are dissimilar, they are all driven by the same dynamic state of the system and are thus functionally related. This implies that the inherent redundancy contained in the static and dynamic relationships among the system inputs and outputs can be used for fault detection purposes. These new schemes for fault detection and troubleshooting have been termed *Analytical Redundancy*, if only quantitative relationships are considered [Patton *et al.*, 1989], and *Functional Redundancy*[1] if qualitative relationships are also considered [Tzafestas, 1989].

Functionally–redundant FMS techniques are basically signal processing methodologies employing mathematical modeling and simulation, state estimation, parameter identification, adaptive filtering, variable threshold logic, statistical decision theory, pattern recognition, fuzzy logic, knowledge–

[1]The term Functional Redundancy is used as synonymous to Analytical Redundancy by researchers in the field of quantitative fault detection, whereas it is used as a term that encompasses Analytical Redundancy by researchers in the field of qualitative fault detection.

based reasoning, qualitative simulation, and various combinatorial and logical operations. Thus functional redundancy is a methodological approach that provides a set of tools for quantitative and qualitative fault detection and troubleshooting of dynamic engineering processes [Clark, 1989]. Functional redundancy encompasses some of the earlier used analytical methods for fault detection such as model–based methods, estimation methods, filtering methods, generation–of–residuals methods, and observers methods.

The combination of quantitative and qualitative fault monitoring approaches opens up a new dimension in fault diagnosis of complex large scale systems with incomplete or difficult to discriminate knowledge, by allowing the evaluation of all available information and knowledge (quantitative and qualitative) of the system for fault monitoring purposes.

## 2.2 Fault Monitoring of Dynamic Systems

To determine why something has ceased to work or why it is no longer working as it used to, it is useful to know how it was supposed to work in the first place. The fundamental issue of diagnosis and troubleshooting can best be understood as the interaction of observation and prediction [Davis and Hamscher, 1988]. Observation indicates what the system is actually doing, whereas prediction indicates what it is supposed to do. A continued comparison between observation and prediction guarantees that any difference or discrepancy will be detected at the earliest possible moment.

Detection is the first step of automated fault monitoring. Fault Monitoring Systems use a combination of knowledge–based and pattern–based approaches to achieve their goal. Automated fault monitoring can be decomposed into six stages [Basseville and Nikiforov, 1993; de Kleer and Williams, 1989]:

a) **Fault detection**: During this stage, the fault monitoring system detects that the plant behavior is abnormal.

b) **Fault isolation**: In this phase, the fault is localized to a particular physical subsystem and/or a functional aspect of operation.

c) **Fault characterization**: At this point, the abnormality is classified in order to simplify the hypothesis formulation process of the diagnosis.

d) **Fault diagnosis**: During this stage, the fault monitoring system traces observed symptoms back to hypothesized failures that might have caused them. This stage is composed of:

- Hypothesis Generation.
- Hypothesis Testing.
- Hypothesis Discrimination.

e) **Fault analysis**: At this point, the system reasons about possible remedies for the previously diagnosed fault.

f) **Fault reporting**: During this stage, the fault monitoring system reports its findings back to the control system or to the human operators, or both.

Stage (a) is naturally pattern–based. It can consist of simple threshold detectors, or time–window detectors [Wang and Cellier, 1991], or more involved demonized routines called "watchdog monitors" [Cellier *et al.*, 1992b].

Stages (b), (c), and (d) can be purely pattern–based, e.g. using analytical or statistical techniques, or purely knowledge–based, e.g. using a rule–based (expert) diagnostic engine, or a mixture of both, e.g. using a model–based deep reasoner [Davis and Hamscher, 1988].

Stage (e) is in all likelihood predominantly knowledge–based. An automated knowledge acquisition system can be used to generate a data base that relates symptoms and failures back to previously successful repair activities [Wang, 1991; Motaabbed, 1992]. Finally, stage (f) is usually straightforward. More refined systems may carry a model of the person they are reporting to, adapting themselves to the perceived know–how of the human operator with respect to the amount of detail presented to him or her in fault reporting [Cacciabue *et al.*, 1990; Wang, 1991].

### 2.2.1 Performance of a Fault Monitoring System

To test the performance of a FMS, a fault can be induced, and the reaction of the detection system can be observed. If the imposed fault induces no response from the detection system, this constitutes a missed detection. On the other hand, if the FMS reacts when no fault is induced, this constitutes a false alarm. Not all faults occur suddenly or persist after they once occurred. Slowly developing or small faults, often known as *incipient faults*, such as bias or drift in an instrument, or intermittent faults, such as a bad contact in an electronic circuit, must be assessed in a different way. Some of the foremost criteria for assessing the performance of a FMS are [Clark, 1989]:

i) Missed fault detections.

ii) Promptness of detection.

iii) Sensitivity to incipient faults.

iv) False alarm rate.

v) Incorrect fault identification.

vi) Robustness.

**Missed fault detection.** A fault has occurred, but the FMS does not detect anything. A missed detection may be acceptable for inconsequential faults, as for example, a small bias on the signal from a relatively unimportant sensor, or unacceptable if the fault has a serious impact on the operation of the monitored system. The sensitivity of the FMS should be adjusted to detect as many small faults as early as possibly while avoiding a high false alarm rate.

**Promptness of detection.** Assuming that a fault is detected successfully, the issue of promptness may be of vital importance. In some aerospace and electric power applications,[2] a fault that persists for a few seconds without detection, identification, and correction, can destroy the faulty component, if not the whole system itself. In other applications it may be more desirable to have reliable detection of minor faults at the sacrifice of speed in detection time or promptness.

**Sensitivity to incipient faults.** In some systems, it is desirable to detect small or slowly developing faults. This is important if the FMS is intended for enhancing maintenance operations in plants by early detection of faulty equipment, in which case promptness of detection may be of secondary importance to sensitivity. In other systems, sensitivity and promptness may both be required. This leads to more complex detection schemes, possibly requiring both hardware and analytical redundancy, for example by using fault–tolerant computer techniques [Gomm *et al.*, 1992].

**False alarm rate.** False alarms are generally indicative of poor performance in a FMS. Even a small false alarm rate during normal operation of the monitored system can be awkward and possibly dangerous because it quickly leads to unacceptably large down times of the system, and, possibly even more

[2] In the introductory section of Chapter 4, the example of an electrical power grid with several interconnected power plants, with one of them suffering a short circuit, is used to demonstrate how fast the FMS should react to avoid an entire region black–out, i.e., it demonstrates the concept of promptness of detection.

critical, to a lack of confidence in the detection system. Chances are that, when a true emergency occurs, the human operators will ignore the alarm because they had been alerted in vain all too often before. A plant operator will simply switch off the monitoring system or at least its audible alarm component if the device goes off and blurts into his or her ears once every 30 seconds. A detection system that has an acceptable false alarm rate during normal operation might still register false alarms while the monitored plant undergoes a real emergency. This might be acceptable in some applications, in which it is preferable to confirm the fault before reacting to it, and unacceptable in other applications, in which even small faults may be so cumbersome that it is preferable to react to false alarms than to suffer deteriorated performance from an undetected, though small, fault. The compromises to be made in the design of a FMS related to false alarm rate, sensitivity to incipient faults, and promptness of detection, are difficult to describe in general or algorithmize, because they require detailed knowledge of the working environment, and an explicit understanding of the vital performance criteria of the monitored system.

There are several ways to reduce the false alarm rate by design. For example, in [Clark, 1989] an approach is described to have adaptive fault detection thresholds in a fault monitoring scheme. The adaptation takes into account large or rapid changes in the input signals (either control signals or measured disturbance signals) of the monitored system, together with the associated transient behavior. By adapting the thresholds, the false alarm rate, at least as far as it is due to deterministic signals, can be minimized. Another approach is described in [Walker, 1989]. It consists on the use of Markov models for threshold determination, which can be a valuable technique when probabilistic information from sensors and system processes can be gathered together.

**Incorrect fault identification.** In the case of an incorrect identification, the detection system correctly registers that a fault has occurred, but incorrectly identifies the subsystem that has failed. If a reconfiguration system exists, it will then proceed to compensate for the wrong fault, an action that could produce consequences as serious as those of a missed detection.

In assessing the performance of a given FMS scheme, or in comparing two different schemes, it is necessary to understand the working environment of the monitored system and to have measures of the aforementioned five criteria. It is further necessary to establish a *fault base* of the monitored system, i.e., a comprehensive set of faults of which it is believed that they may occur in the system, and which are to be detected by the FMS scheme when they occur. This is a difficult task that is closely related to the *robustness* issue to be discussed in the sequel.

### 2.2.2 Robustness

The robustness of a FMS is the degree to which its performance is unaffected by unforeseen conditions in the monitored system. Evidently, since it is the very purpose of the FMS to report errors, robustness is essential to its mission. Robustness can be discussed with respect to the following characteristics of the monitored system as [Basseville and Nikiforov, 1993; Clark, 1989]:

i) Parameter uncertainties.

ii) Unmodeled non–linearities or uncertain dynamics.

iii) Disturbances and noise.

iv) Fault types.

**Parameter uncertainties.** A major problem in the field of robustness in redundant FMS schemes is caused by uncertainties in the physical variables of the monitored system. In these schemes, variables that are not directly measured are reconstructed by means of state estimation techniques from the signals that are measured. These state estimators are essentially mathematical models of the monitored system. For that reason, they critically depend on a correct knowledge of the values of the many physical characteristics and parameters of the plant. If these are all known with precision, the state estimates will be accurate, and the FMS scheme will be remarkably sensitive to incipient faults and immune to false alarms. However, in most processes, even if they are structurally modeled correctly, some physical parameters are only approximately known. Consequently, the state estimators must be designed using only nominal values for the uncertain parameters or using some accommodating mechanism to compensate for the uncertainty. The result is that the state estimates are always in error, the severity of which depends upon the maneuvers of the monitored plant in ways that are not easily determined. Inaccurate state estimation of non–measured signals often leads to false alarms, or, if the FMS is protected against false alarms, may lead to missed detections.

**Uncertain dynamics.** All dynamic plants are essentially non–linear. However, many plants can be linearized around an operating point or an operating trajectory. Many FMS schemes are based on such linearized models, which may be quite satisfactory as long as the plant doesn't deviate too far from its nominal operating point or trajectory. However outside this range, the non–linearities of the plant produce signals that are not modeled accurately by the FMS scheme, and this may then lead to either false alarms or missed fault detections.

**Disturbances and noise.** Dynamic plants are always subjected to inputs other than those intended by the system designer. These inputs, called disturbances, are usually random functions originating in the environment. Furthermore, the sensors usually have superimposed noise on their signals. This noise is also random, but it originates from a different source, and is normally uncorrelated with the disturbances. Most signal processing tools used by designers to account for random fluctuations of this sort, are based on the assumption that those fluctuations are stationary Gaussian processes having known characteristics. If the actual disturbances and noise are non–stationary, non–Gaussian, or even correlated in some way, then the FMS scheme will perform below its expected level.

**Fault types.** A given system can malfunction in many ways. For example, a sensor can break down or suffer a change of scale factor, a bias may not remain constant, a part may change its characteristics due to wear or friction, a moving part can get stuck in a fixed position, there can be excessive noise in the signals, etc. A majority of FMS schemes are designed to detect and discriminate between a set of prespecified fault types, and even this becomes cumbersome as the number of fault types is increased. Clearly, if a malfunction should occur that is not in the knowledge base of the FMS, then the FMS scheme may still detect that something strange has occurred, but it wouldn't be able to recognize the fault. Many FMS schemes will then go ahead and propose the closest match as the hypothesized fault, rather than declaring the fault as being of unknown origin. In order to recognize a fault as unknown, the FMS needs an assessment of the sanity of its own reasoning processes, i.e., an estimate of the error contained in its predictions. Many methodologies used in FMS schemes do not offer any self–assessment capabilities, which can be a severe shortcoming of these tools.

A FMS that is robust to at least some fault types must include both a hypothesis generation process and a hypothesis testing process. Hypothesis generation can be performed using either a quantitative model–based approach based on state or parameter estimation, or through the use of qualitative techniques such as knowledge–based, connectionist, or model–based systems.[3] The robustness of a FMS scheme is directly related to the quality of its hypothesis generation engine. Human operators, through their experience

[3]Model–based FMS schemes can use either quantitative or qualitative models.

and knowledge of a specific system or process, will generate hypotheses based on a mixture of quantitative and qualitative information. The ingenuity of a human operator cannot currently be matched by any automated FMS. In order to advance the state of the art of automated FMS schemes, it could thus be argued that the hypothesis generation engine of a powerful FMS scheme should be based on both quantitative and qualitative reasoning.

## 2.3 Quantitative Fault Detection

The field of fault detection and troubleshooting can be divided into two main research areas: quantitative and qualitative fault detection. In spite of the fact that both research directions started approximately at the same time, the research on quantitative approaches to fault detection has produced much more quickly generally applicable results than its qualitative counterpart, and consequently, these approaches are more widely used today. However, in the past decade, qualitative fault detection techniques have finally caught up with their quantitative cousins, and have become an attractive alternative to the formerly used quantitative methods due to their accuracy, robustness, simplicity, and low computational cost.

Quantitative fault detection is primarily based on statistical techniques, first order logic, control theory, mathematical modeling, and computer simulation. Among these techniques, the most widely used are those that make use of measurable variables, such as the *Limit and Trend Checking Techniques* and *Automated Statistical Diagnosis*, and those that also work with non–measurable state variables and process parameters, such as the *Control Derived Techniques*. Two very good general summaries of fault detection and troubleshooting of dynamic systems have been presented by [Himmelblau, 1986] and [Isermann, 1981] that include lots of examples. Good summaries of the statistical approaches to fault detection and troubleshooting can be found in the books by [Pau, 1981], [Basseville and Benveniste, 1985], and [Basseville and Nikiforov, 1993], and the paper by [Willsky, 1980]. Control theory derived approaches became fashionable in the late eighties. They form part of the *Analytical Redundancy* methods. Good summaries are provided in the papers by [Isermann, 1984], and [Frank, 1990], and the book by [Patton *et al.*, 1989]. There are a series of authors that combine statistical and control approaches to fault detection and troubleshooting. Summaries of this work can be found in [Basseville and Nikiforov, 1993; Chow and Willsky, 1984; Kumamaru *et al.*, 1984; Kumamaru *et al.*, 1989].

### 2.3.1 Limit and Trend Checking Techniques

In the early days, supervision of technical processes was restricted to directly checking measurable variables for upper or lower transgression of fixed or adjustable limit or setpoint values. This technique was automated by using simple limit–value monitors in an approach known as *Limit and Trend Checking*. This is the classical quantitative fault detection technique, which is widely used in combination with other quantitative and qualitative methods. It works by directly checking a measurable variable $y(t)$. A signal to an alarm or an actuator is released as soon as a maximum value $y_{max}$ is passed in the positive direction, or a minimum value $y_{min}$ is passed in the negative direction. The normal state is:

$$y_{min} < y(t) < y_{max} \tag{2.1}$$

The limits are usually set such that, on the one hand, a large enough distance to the occurrence of damage is retained, and on the other hand, unnecessary fault alarms are avoided, i.e., the false alarm rate is kept as low as possible. This is referred to as *absolute value check*.

The limit check can also be applied to the trend $\dot{y}(t)$ of the variable $y(t)$. If the limit values are set small enough, the fault alarm can take place earlier than in the former case since the trend permits a prediction of the variable progression. The normal state is:

$$\dot{y}_{min} < \dot{y}(t) < \dot{y}_{max} \tag{2.2}$$

A combination of absolute value and trend checking is shown in [Isermann, 1981]. If only limit checking is applied, the limits are usually set on the safe side to allow sufficient time for counteractions. However, this can lead to false alarms if the variable returns to the normal state without external action. This disadvantage can be avoided if a generalized threshold is established that measures the amount of time that the variable $y(t)$ remains outside the limits $y_{min}$ or $y_{max}$ within a given time window. An alarm is set off whenever the so filtered signal passes a given threshold value. Unfortunately, methods that avoid false alarms always increase the rate of missed faults.

Almost every FMS contains one or more limit value monitors. Consequently, there have been reported many application examples. Among them, some representative papers are: [Shapiro and Decarli, 1979] for fault monitoring of a Lockheed L–1011 Tristar aircraft, [Brown and Rao, 1985] for fault monitoring in a F–100 non–linear aircraft simulator, [Walker, 1989] for fault monitoring of the space shuttle inertial measurements units, and [Berkan and Upadhyaya, 1991] for fault monitoring of noise in control systems of nuclear plants.

### 2.3.2 Automated Statistical Diagnosis

*Automated Statistical Diagnosis* is derived directly from pattern recognition in medical diagnosis, and is defined as the recognition of possible failure causes from observed symptoms and previous operating history. Since sometimes medical diagnosis is based exclusively on the analysis of the answers given by patients to certain questionnaires without the clinical examination of a physician, it can be envisaged in a similar way for engineering applications that failure diagnosis can be based on statistical analysis of available information about the system operation and environment prior to the time of failure. This statistical analysis could be automated with or without dismantlement of the analyzed system component, i.e., with or without the intervention of the physician.

The simplest example of Automated Statistical Diagnosis is *classification* by association of observed symptoms with a suspected failure cause using a symptom–failure matrix; however, since failure causes may be internal or external, or may not have been seen before, direct classification proves normally inadequate. If the symptoms are represented by a measurement vector, this vector constitutes a pattern in the terminology of *pattern recognition*. In this case, an automated diagnosis is the automatic recognition of a symptom pattern from the set of all observable patterns divided into failure modes or cause classes.

Recognition always assumes previous learning. This learning is made by discovering similarities or patterns among the available data. Thus the basic idea of Automated Statistical Diagnosis is that of performing a high–order classification of patterns or sets of similarities to generate a symptom–failure table. The main steps of this technique are [Pau, 1981]:

i) **Recognition of similarities.** This is the learning step. Training data generally consist of large numbers of similar patterns. Two scenarios are possible: a) the training data have previously been assigned to classes, or b) they have not previously been classified. In the former case, learning can be limited to selecting representatives of each class; whereas in the latter case, clustering methods must be used to group or cluster similar patterns for constructing a set of natural classes, after which a type (b) problem becomes a type (a) problem. Clustering methods are based on similarity functions and require the determination of the maximum number of natural classes. Common techniques to carry out the learning phase are:

- Correlation analysis.
- Correspondence analysis.
- Principal component analysis.
- Discriminant analysis.
- Canonical analysis.

ii) **Classification of similarities into classes.** The classification is performed combining the estimation of the class conditional probabilities with a certain decision rule. The objective is the discrimination of similarities belonging to different classes while preserving the maximum of the discriminatory information. Commonly used methods are:

- Neighbors distance measures (nearest, farthest, and average neighbor metrics).
- Geometric distance measures (Minkowsky, Chebychev, and quadratic metrics).
- Separation measures (Shannon entropy, Bayesian distance, and Kullbach divergence metrics).
- Generalized likelihood ratio.
- Generalized Jensen differences.

iii) **Diagnosis.** The diagnosis is performed by means of comparing the previously classified patterns of the training data with those of the testing data, i.e., the currently observed patterns.

It is important to understand that the goal of Automated Statistical Diagnosis in medical or engineering applications is not the localization of the damaged elements, but rather, the determination of the most probable failure causes (or modes) given the observations. This technique is particularly well suited for complex failure situations involving wear and multiple failures, and also when direct measurements and logical tests prove inadequate, but it requires numerous preliminary tests to obtain training data about all possible operating conditions including failures.

Numerous applications of this approach are described in the literature. Among them, [Wilbers and Speyer, 1989] used a Bayesian hypothesis testing metric as the classification method for automated statistical diagnosis of aircraft sensors. [Weiss *et al.*, 1985] used a sequential probability ratio test for fault detection in a F–100 linear aircraft simulator. The same application is presented in [Emami–Naeini *et al.*, 1986] with a Bayesian decision metric for hypothesis testing. [Pau, 1981] showed a fault detection scheme based on the nearest neighbor classification rule for quality control in a discrete parts manufacturing line. [Basseville and Nikiforov, 1993] and [Kumamaru *et al.*, 1989] used the generalized likelihood ratio, the Kullback discrimination metric, and the Jensen difference measure as classification techniques for automated statistical diagnosis of black box systems.

### 2.3.3 Control–Derived Methodologies

The control–derived methodologies,[4] currently summarized in the term *Analytical Redundancy*, normally predict some output variables, or internal variables, or model parameters, or model structure to compare them against those obtained from the real monitored system. Thus, control–derived methodologies are based on mathematical or behavioral models of the monitored systems, i.e., they are all model–based. These techniques are particularly useful when process faults are indicated by internal, non–measurable process state variables or by internal, non–measurable physical process coefficients, or even when the internal structure of the system is unknown (black box systems). In the first case, attempts can be made to reconstruct or estimate these state variables from the measurable variables by using a known process model. In the second case, attempts can be made to determine the changes of the process coefficients via the changes in the process model parameters. In the third case, attempts can be made to identify the possible structure of the system by analyzing the input/output data. In all three cases, the difference that arises from a comparison between the real monitored system and the modeled system is known as the *residual*, and it serves as the fault indicator.

From a model–based perspective, the control–derived approaches to fault

[4] With the term Control–Derived Methodologies, we denote what other authors call Quantitative Model–Based Approaches

detection and troubleshooting that have been reported in the literature during the last years can be traced back to a few basic historical concepts. Among them, the most important ones from the point of view of the construction of a FMS are: the Detection Filter [Beard, 1971; Jones, 1973], the Residuals Generation by means of a single Kalman Filter [Mehra and Peshon, 1971] or banks of Kalman Filters or Luenberger Observers [Clark *et al.*, 1975; Montgomery and Caglayan, 1974], the Parity Space approach [Deckert *et al.*, 1977], the Parameter Estimation technique [Kitamura, 1980; Zhu and Backx, 1993], the structure identification algorithms [Isermann, 1989], and the methodological set of tools known as Analytical Redundancy [Isermann, 1984; Frank, 1990].

The procedure of evaluating the redundancy contained in the mathematical model of the system can be roughly divided into the following two steps:

i) Generation of the so–called residuals, i.e., functions that are accentuated by a fault vector $f$.

ii) Detection, isolation, and diagnosis of the faults (time, location, etc.).

In essence, there are two different ways of generating fault–accentuated signals (residuals) using Analytical Redundancy: a) by parity checks, observer schemes, and detection filters on the one side, all of which use state estimation techniques; and b) by parameter estimation on the other side. The resulting signals are employed to form decision functions as, for example, norms or likelihood functions.

A detailed structural diagram of the overall procedure is depicted in Figure 2.1.[5] Notice that three kinds of models are required for the residual generation, namely: one for the nominal system, one for the observed system, and one more for the faulty system. In order to achieve a high performance of fault detection with a low false alarm rate, the nominal model should be tracked and updated by the observation model.

The analytical redundancy approach requires that the residual generator performs some kind of validation of the nominal relationships of the system, using the actual input $u$, and the measured output $y$. The redundancy relations to be evaluated can simply be interpreted as input–output relations of the dynamics of the system. If a fault occurs, the redundancy relations are no longer satisfied, and a residual $r \neq 0$ results. The residual is then used to form

[5]This figure has been taken from [Frank, 1990], and originally came from [Isermann, 1984].

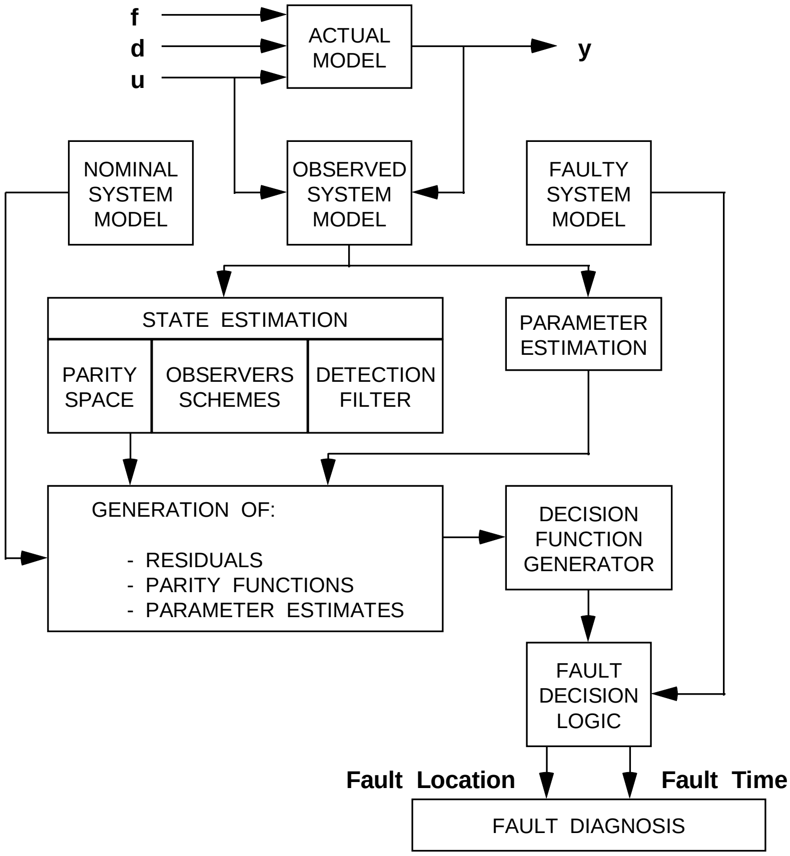


Figure 2.1: General scheme of a FMS using Analytical Redundancy.

appropriate decision functions that are evaluated in the fault decision logic in order to monitor both the time of occurrence and the location of the fault.

The basis for a decision on the occurrence of a fault is the *fault signature*, i.e., a signal that is obtained from some kind of faulty system model defining the effects associated with a fault [Isermann, 1984]. In most applications the process is completed when the fault location and fault time are identified. However, it may be desirable to get a deeper insight into the situation by knowing the fault type, size, and cause which can be acquired by subsequent fault diagnosis. To this end, deeper knowledge about the nature of the process, such as the degree of aging, the operational environment, the history of operation and maintenance, the fault statistics, etc. is required. This task is therefore commonly tackled with the aid of qualitative techniques such as expert systems, neural networks, model–based deep reasoners, and the like.

Since the most important approaches are based on the aforementioned techniques, they shall be explained here in some detail.

#### 2.3.3.1 Parity Space Approach

The central idea of the Parity Space approach, also called *Dedicated Dead–Beat Observer approach*, is to check the parity (consistency) of the mathematical equations of the system (relationships in analytical redundancy terms) by using the actual measurements. A fault is detected once preassigned error bounds are exceeded. There are basically two forms of analytical redundancy relations:

i) *Direct redundancy:* Relations among instantaneous redundant sensor outputs (algebraic relations).

ii) *Temporal redundancy:* Dynamic relationships between sensor outputs and actuator inputs (differential or difference equations).

To outline the basic idea of this approach, the simplified case of redundant measurements that can be obtained directly or from analytical sources will be considered. These redundant measurements can be modeled by the algebraic measurement equation:

$$y = Cx + \Delta y \tag{2.3}$$

where

$$\begin{array}{lcl} y & = & (q \times 1) \text{ measurement vector;} \\ C & = & (q \times n) \text{ measurement matrix of rank n;} \\ x & = & (n \times 1) \text{ true measurement value;} \\ \Delta y & = & (q \times 1) \text{ error vector.} \end{array}$$

$\Delta y > b_i$ defines a faulty operation indicated by the $i$th measured variable. For detection of $\Delta y$, the vector $y$ can be combined to a set of linearly independent parity equations given by

$$p = Vy \tag{2.4}$$

where $p$ is a $(q-n)$–dimensional parity vector, and $V$ is the $(q-n)\times q$ projection matrix. A parity equation is simply an input–output model for a part of the dynamics of the system. In other words, instead of checking the consistency of the overall mathematical model, the approach allows to check individual relations that are part of the model, which permits the selection of the most reliable relations and thus the creation of a robust fault monitoring strategy.

The parity equations contain only the errors due to the faults, independently of $x$ which is not directly measured. Moreover, in the parity space, the columns of $V$ define $q$ distinct fault directions associated with each measurement. This ensures that a fault in measurement $i$ implies a growth of $p$ in the $i$th direction. The $q$–dimensional residual vector is thus,

$$r \;=\; y - C\hat{x} \tag{2.5}$$

where $\hat{x}$ is the least squares estimate of $x$. In these terms, the fault detection and isolation problem can be formulated as follows: given $q$ redundant measurements $y_1, \ldots, y_q$ of a process variable, and symmetric error bounds $b_1, \ldots, b_q$ characterizing the faulty behavior:

1. Find an estimate $\hat{x}$ of the process variable from the most consistent subset of measurements.

2. Identify the faulty measurement by parity checks. To detect a single fault among $p$ components, at least $p-1$ parity relations are required.

The generalization of this approach for the case of using the temporal relations of the dynamic system can be found in [Basseville and Nikiforov, 1993; Chow and Willsky, 1984; Lou *et al.*, 1986]. Applications of this approach to SISO[6] engineering systems are presented in [Desay and Ray, 1981; Massoumnia, 1986], and to MIMO[7] engineering systems in [Frank and Wünnenberg, 1989].

#### 2.3.3.2 Dedicated Observer Approach

Some authors have approached the fault detection problem by directly starting with banks of Luenberger observers or Kalman filters. The basic idea of the

[6] Single input single output system

[7] Multiple input multiple output system

observer approach is to reconstruct the true output of the system from the available perturbed measurements, or subsets thereof, with the aid of observers or Kalman filters using the estimation error or innovation, respectively, as a residual for the detection and isolation of faults. It is known from observer theory that, for state or output estimation, linear or non–linear, full– or reduced–order state observers can be used in the deterministic case, or Kalman filters in the stochastic case when noise has to be considered.

The fundamental configuration of a linear full–order state estimator simply consists of a parallel model of the process with a feedback of the output estimation error $e = y - \hat{y}$. The feedback is important for several reasons:

- To compensate for differences in the initial conditions.
- To influence the stability properties of the observer or filter model, i.e., make the observer (or filter) sufficiently fast.
- To provide the filter design with additional free design parameters, e.g. for decoupling the effects of faults to be detected from those of other faults or disturbances.

For illustration purposes, the case of a linear process shall be considered, in which all the faults are contained in the fault vector $f$, whereas all the other effects that obscure the fault detection are contained in the disturbance vector $d$. The system can be described by the state–space model:

$$\dot{x} = Ax(t) + Bu(t) + Ed(t) + Kf(t) \tag{2.6}$$

$$y(t) = Cx(t) + Fd(t) + Gf(t) \tag{2.7}$$

where

| | | |
|---|---|---|
| $x$ | $=$ | state vector; |
| $u$ | $=$ | known input vector; |
| $y$ | $=$ | measured output vector; |
| $A, B, C, E, F, G, K$ | $=$ | known matrices of appropriate dimensions. |

The state $x$, and the output $\hat{y}$ of a full–order observer are governed by:

$$\dot{\hat{x}} = (A - HC)\hat{x} + Bu + Hy \tag{2.8}$$

$$\hat{y} = C\hat{x} \tag{2.9}$$

where $H$ denotes the feedback gain matrix that has to be chosen properly to achieve the desired observer performance. With equations 2.6, 2.7, 2.8, and 2.9, the relations for the state estimation error $\epsilon = x - \hat{x}$, and the output estimation error $e = y - \hat{y}$ become

$$\dot{\epsilon} = (A - HC)\epsilon + Ed + Kf - HFd - HGf \tag{2.10}$$

$$e = C\epsilon + Fd + Gf \tag{2.11}$$

As can be seen, the output estimator error $e$ is a function of $f$ and $d$, but not of $u$. Hence, $e$ can be used as the residual $r$, for the purpose of detection and isolation of the fault. When no fault occurs, i.e., $f = 0$, then $r$ will be only influenced by the unknown input $d$, whereas, if there is a fault, i.e., $f \neq 0$, $r$ will be increased. Thus a fault can be detected by checking the increment of $r$ caused by $f$. In the simplest case, this can be done by a threshold logic [Clark, 1978a,b]. In a similar way, residuals can be generated using reduced–order or non–linear estimators [Frank, 1987a,b].

The main task associated with the design of state estimators for fault detection and isolation is their optimization by a proper choice of the feedback gain matrix $H$. The most simple configuration is composed of a single estimator (observer or Kalman filter), where a single full– or reduced–order estimator is driven by only one (the most reliable) sensor output, and the full output is reconstructed [Clark, 1978a]. The comparison of the actual output $y$ with the estimated output $\hat{y}$ using a threshold logic allows a unique detection and isolation of a single fault.

In [Mehra and Peshon, 1971], the use of a single Kalman filter driven by a full output vector is demonstrated. The occurrence of a fault is monitored by *statistical innovation tests* of mean and covariance. The fault isolation is carried out on the basis of different fault hypotheses. The *Multiple Hypothesis Testing* can be carried out using Bayesian Decision Theory [Willsky, 1976].

More flexibility in the isolation of a fault can be achieved by using an estimator scheme, i.e., a bank of estimators driven by the actual output vector $y$. In this case, each of the estimators is designed for a different fault hypothesis. These hypotheses are then tested in terms of likelihood functions using Bayesian Decision Theory [Wilbers and Speyer, 1989]. Approaches to hypothesis testing can be found in [Montgomery and Williams, 1989; Gorton and Gray, 1989]. These authors show how Kalman filters based on multiple dynamic models can be used for hypothesis testing by processing the generated residuals. In the case of the former of these papers, this process is carried out by means of sequential probability ratio testing, whereas in the case of the latter publication, the process is based on multiple model hypothesis testing using a bank of Kalman filters.

A variation of the Observer approach can be obtained by assigning a dedicated estimator to each of the sensors. In this *Multiple Observer Scheme* [Clark, 1989], each estimator is driven by a different single sensor output, and the complete output vector $y$ is estimated. With this scheme, multiple simultaneous faults can be detected and isolated by checking properly structured sets of estimation errors, for example, with the aid of a threshold logic. If a certain sensor fails, the related output estimate reconstructed by the corresponding estimator will be erroneous, which can be identified by the logic.

An alternative version, the so–called *Generalized Observer Scheme* [Frank, 1987a,b] provides that an estimator dedicated to a certain sensor be driven by all outputs except that of the respective sensor. This scheme allows the detection and isolation of a single fault in any of the sensors, however, with increased robustness with respect to the unknown inputs.

For robust FMSs based on observers or state estimators, the computation of parameter and dynamics uncertainties can be combined. In [Frank and Wünnenberg, 1989], it is shown how the unmodeled or uncertain dynamics of a system can be contained in an uncertain disturbance distribution matrix, turning the robust fault detection problem into one of *disturbance decoupling* by design. If an observer or state estimator is used, modeling errors and dynamic uncertainty can be shown to act like a disturbance on a linear system. If the observer is a linear system also, then the approach is known as the *Unknown Input Observer Scheme*. The same approach can be found in [Patton and Kangethe, 1989]; however, they based the uncertain disturbance distribution matrix on *eigenstructure assignment*.

Another option closely related to those mentioned above is one that combines software and hardware redundancy. The most common practice of a *duplex sensor system* uses two identical sets of instruments, each set being supervised by a FMS scheme of one of the above mentioned types [Deckert *et al.*, 1977; Onken and Stuckenberg, 1979]. Once a fault in one of the sensors has occurred, it is detected with the aid of an observer scheme, and the system is then switched to the healthy sensor. The motivation for using two identical sensors is to detect the occurrence of a fault by hardware redundancy and only perform the isolation task by Analytical Redundancy. This saves computational burden, increases the reliability, and allows the Analytical Redundancy tests to be triggered. This concept has already gained some practical importance for FMSs in aircrafts [Labarrère, 1987]. Statistical and geometric representations of this approach are explained in [Basseville and Nikiforov, 1993].

The Observer or Kalman Filter approach is the most widely used technique in Analytical Redundancy. Examples of the application of this fault monitoring technique to real engineering problems are: inertial navigation systems using one Kalman filter for each instrument [Kerr, 1987], detection of instrument and component faults in a nuclear reactor [Niccoli *et al.*, 1985], and non–linear aircraft modeling using Eigenstructure assignment of observers [Patton and Kangethe, 1989].

#### 2.3.3.3 Fault Detection Filter

The fault detection filter or fault sensitive filter is a full–order state estimator with a special choice of $H$, so that the residual $r$ due to a particular fault $f_i$ is constrained to a single direction or plane in the residual space, independent of the mode (size or time history) of $f_i$. To describe the underlying idea, let us start with the system state equations in the form:

$$\dot{x} = Ax(t) + Bu(t) + K_i f_i(t) \tag{2.12}$$

$$y(t) = Cx(t) + \bar{K}_j \bar{f}_j(t) \tag{2.13}$$

where $K_i$ and $\bar{K}_j$ are distribution vectors, and $f_i$ and $\bar{f}_j$ are scalar functions of time with the following meaning:

$K_i$ = $n \times 1$ design fault direction with $i = 1, 2, \cdots, r$;
$r$ = the number of fault directions;
$\bar{K}_j$ = $q \times 1$ unit vector associated with $\bar{f}_j(t)$;
$f_i(t)$ = actuator or component fault mode;
$\bar{f}_j(t)$ = sensor fault mode.

To reconstruct the states of the state vector $x(t)$ from measurable input and output signals, a full–order observer like that of Equations 2.8 and 2.9 can be used. A $p \times 1$ residual vector is generated from the difference between the actual and the estimated measurements having the form:

$$r = We = W(y(t) - \hat{y}(t)) \tag{2.14}$$

where $W$ is a $p \times m$ weighted matrix, and $e$ is the output estimation error $e = y - \hat{y}$.

The feedback gain matrix $H$ of the observer can be chosen in such a way that the residual vector $r$ has certain directional properties at the occurrence of a certain fault. If, for example, an actuator or component fault occurs, then the error model for the state estimation error $\epsilon = x - \hat{x}$ becomes:

$$\dot{\epsilon} = (A - HC)\epsilon + h_i f_i \tag{2.15}$$
$$r = WC\epsilon \tag{2.16}$$

but, if a sensor fault occurs, then

$$\dot{\epsilon}_j = (A - HC)\epsilon_j + h_j \bar{f}_j \tag{2.17}$$
$$r = WC\epsilon_j + \bar{K}_j \bar{f}_j \tag{2.18}$$

where $h_j$ is the $j$th column of the detection filter gain matrix.

As can be seen, the actuator or component fault residual of the detection filter comes to lie in the direction of $WCK_i$, but the sensor fault residual can only be made to lie in a plane. In conclusion, the important information for the fault detection is in the direction or plane of the residual, rather than in its time function, i.e., no knowledge is required about the fault mode $f_i$.

It is a very appealing feature of the Fault Detection Filter approach that the residual direction of the fault is not affected by the fault mode. However, if unknown inputs, including parameter variations, have to be considered, then the observer should be designed in such a way that its residuals become decoupled with respect to disturbances. The difficulty to distinguish the effects of faults from the effects of disturbances acting on the system can be seen just looking at Equation 2.11. Therefore, in a robust fault detection filter scheme, the remaining problem is how to choose the matrices $H$ and $W$ in such a way that the entries in the transfer function matrix between the residual and the disturbance are zeroed.

Applications of this approach can be found in [Frank, 1990; Isermann, 1984; Meserole, 1981; Patton *et al.*, 1987; Patton *et al.*, 1992]. An approach that takes into account the effects of disturbances (parameter variations or measurement noise) by means of precise modeling can be found in [Wilbers and Speyer, 1989].

#### 2.3.3.4 Parameter Estimation Approach

It may sometimes happen that a mathematical model cannot be obtained due to the lack of knowledge about the structure of the system or due to its nonlinear behavior. In such a case, *Parameter Estimation algorithms* can be used to obtain an approximation of the model's structure. This parameter estimation approach is an alternative to the above described methods that are based on state estimation. This technique makes use of the fact that faults in a dynamic system are reflected in its physical parameters as, for example, mass, resistance, inductance, etc. The idea behind this approach is to detect the faults via estimation of the parameters of the model that should change once a fault has occurred [Isermann, 1984]. Thus, a parametric model of the system structure must be proposed. This model is represented by means of mathematical functions, the parameters of which, originally unknown, are estimated through optimization algorithms. The whole process works as follows:

i) A parametric model of the system must be proposed, and the relationships between those parameters and the physical parameters must be determined. There exist some methods for such a proposition and its validation, the most widely used are [Ljung, 1987; Zhu and Backx, 1993]:

- **ARX.** Auto Regressive External Input. It is the simplest model and uses the linear regressive method, the parameters of which are adjusted by means of the least squares technique. The relationship between the error function and the estimated parameters must be linear.
- **Output error.** The function to optimize is not the equation error as in the least squares method, but the output error function that is normally minimized by means of a numerical method. The relationship between the error function and the estimated parameters can be non–linear.
- **Instrumental Variables.** This approach is based on correlation analysis. An instrumental vector is proposed in such a way that it is uncorrelated with the error vector, and correlated with the undisturbed components of the input/output data vector. This constitutes the convergence criteria for the minimization of the error function by means of the least squares method.
- **ARMAX.** Auto Regressive Moving Average External Input. Two models are identified with this method, one for the system's dynamics and another for the perturbations by using the linear regressive method, the parameters of which are adjusted by a moving average technique applied to an extended least squares method.
- **NARMAX.** The same as the ARMAX model, however, with non–linear capabilities.
- **Box–Jenkins.** This approach is especially useful in closed–loop situations. As in the ARMAX approach, two models are identified, one for the internal structure and one for the disturbances. The error function can be non–linear and is normally minimized by a numerical method.

ii) Identification of the model parameter vector using the inputs and outputs of the original system and a given criterion of fit.

iii) Determination of the physical parameter vector.

iv) Computation of the deviations vector by comparing the currently obtained parameters against those taken from the original model.

v) Decision on a fault by exploiting the relationships between faults and changes in the physical parameters.

This approach has primarily been used for detection of incipient faults in SISO and MISO systems. A detailed description can be found in [Isermann, 1984; Patton *et al.*, 1989; Frank, 1990; Zhu and Backx, 1993]. Examples of the application of this fault detection and troubleshooting approach are: the fault diagnosis of a DC–motor centrifugal pump and of a tubular heat exchanger, both in [Isermann, 1989], fault diagnosis of a nuclear reactor [Kitamura, 1989], and early warning on a glass tube process and on a single–stand rolling mill [Zhu and Backx, 1993].

#### 2.3.3.5 Limitations of Control–Derived Methodologies

All of the aforementioned control–derived methodologies for fault detection and isolation, and in general all quantitative model–based methodologies, suffer from a fundamental practical limitation: since the system model on which the redundancy is based is never known exactly, the actual system output $y$ will never match the model output $\hat{y}$, even when there are no faults present in the system. The consequence is that the residual $r = y - \hat{y}$ will be non–zero in general, forcing the use of thresholds and moving averages to prevent false alarms from occurring and to distinguish between different fault types. The problem with thresholds is that they not only reduce the sensitivity to faults, but also, that the appropriate threshold level varies with the input signal of the actual system and the magnitude and nature of the system disturbances. Choosing the threshold too small increases the false alarm rate, whereas choosing it too large reduces the net effect of fault detection. There is therefore a strong motivation for reducing the sensitivity of the residuals with respect to the modeling errors.

This robustness problem had been recognized early on, and several approaches to increase the robustness of fault detection and isolation schemes have been suggested with more or less success. [Zhu and Backx, 1993] give a detailed description of the available techniques for robust parameter estimation, while [Frank, 1990] summarizes the developed state estimator design techniques for robust residual generation. Other significant contributions to the robust observer design can be found in [Watanabe and Himmelblau, 1982; Massoumnia, 1986; Patton *et al.*, 1987, 1989; Ge and Fang, 1988; Wilbers and Speyer, 1989]. A systematic solution using the *Unknown Input Observer* scheme was provided by [Wünnenberg and Frank, 1987; Ge and Fang, 1988].

Other authors tackle this robustness problem from the parity space point of view. From this perspective, the residual of an estimator can be viewed as the most general parity function containing the complete set of redundancy relations. The underlying idea of robustness generation is that of using only those redundancy relations that are most reliable. This approach can be found in [Chow and Willsky, 1984; Lou *et al.*, 1986; Weiss *et al.*, 1995].

On the other hand, there are several approaches to increase the robustness by an optimal choice of the threshold [Emami–Naeini *et al.*, 1986], or by making the threshold adaptive to the inputs [Clark, 1989].

## 2.4 Qualitative Fault Detection

In the previous sections, modern quantitative techniques for fault detection and troubleshooting have been examined, and their advantages and disadvantages have been stated. Their main drawback is that all of them operate on a quantitative and precisely formulated plant model, and even though many of these techniques operate on linearized parametric models, the parameters of these models must be accurately known. Not only is this quantitative information often hard to come by, it may in fact be irrelevant. If human plant operators are asked what information they use to manually troubleshoot their plant, it turns out that they often rely on heuristic knowledge that may not be possible to capture in terms of crisp mathematical equations. In the aforementioned quantitative plant monitoring and troubleshooting schemes, it is very difficult, if not impossible, to incorporate into the models the *a priori* expertise, and the behavioral, uncertain, and heuristic knowledge, i.e., the volume of qualitative information about how the system is supposed to work that human plant operators possess and bring to bear when diagnosing their plant.

In complex real–life engineering problems, a lot of these types of knowledge will be needed in the model, since the monitored system could be highly non–linear or partly unknown. If this is the case, it may be preferable to use alternate troubleshooting techniques that offer the opportunity of including the available qualitative plant knowledge in their reasoning processes, and that will not break down when confronted with incomplete knowledge.

Evidently, this does not imply that precise quantitative information about some aspects of system behavior should be thrown away. A powerful FMS strategy should be able to exploit all of the available knowledge, both quantitative and qualitative, in an optimal fashion, without breaking down when some pieces of information are not available. Techniques that can only operate on qualitative knowledge are at least as limited in their applicability as the previously discussed purely quantitative schemes, and, in fact, may even be more limited, because they frequently lead to ambiguous conclusions that cannot be further discriminated on the basis of the qualitative knowledge alone.

Hence it shall be the foremost aim of this thesis to present a fault monitoring and analysis methodology with *adaptive granularity*, a technique that is able to exploit all the information presented to it in an optimal fashion. In the presence of complete information, the scheme will be competitive in terms of fault discrimination power (though not necessarily execution efficiency) with the best among the purely quantitative non–linear fault monitoring schemes. In the case of incomplete information, the performance of the scheme will gradually deteriorate without ever breaking down completely. If only qualitative information is provided, the methodology will be competitive in terms of fault discrimination power with the best among the purely qualitative fault monitoring techniques.

In this section, qualitative fault monitoring and analysis techniques are being discussed. However, the reader should bear in mind, that it has not ever been our intention to advocate purely qualitative techniques as a viable alternative to the proven quantitative methods discussed earlier in this chapter.

One of the major problems in automated fault monitoring system design is the possibility of the occurrence of unforeseen faults in a system. A fail–safe system must be able to cope with incomplete knowledge, and with redundant, or difficult to discriminate, or even inconsistent information [Davis, 1982, 1984]. Various approaches to qualitative fault detection and troubleshooting have been proposed that offer different solutions to these difficult problems. Among them, the more widely used, from a large–scale system perspective, are [Puccia and Levins, 1985]:

- Knowledge-based systems.
- Connectionist systems.
- Model-based systems.
- Qualitative simulation.

A fairly large volume of literature is available that describes the application of these qualitative methodologies (especially Expert Systems and Artificial

Neural Networks) to fault detection and troubleshooting. Some general surveys can be found in the books by [Boullart *et al.*, 1992] and [Krijgsman, 1993]. Summaries of general–purpose Expert Systems can be found in [Cuena, 1987; Giarratano and Riley, 1989; Waterman, 1986]. A description of Real–time Expert systems can be found in [Crespo, 1993]. Fuzzy Expert Systems are explained in detail in [Kandel, 1992]. The application of knowledge–based methodologies to fault detection and diagnosis is discussed in [Tzafestas *et al.*, 1987; Tzafestas, 1989, 1993]. The Connectionist approach (also known as subsymbolic approach), mostly represented through Neural Networks, is elaborated on in [Hopfield, 1982; Khanna, 1990; Kohonen, 1984; Kosko, 1992; Lippman, 1987]. The application of Neural Networks to control and diagnosis is well summarized in the book by [Miller *et al.*, 1990]. Fault diagnosis using Decision Trees and Model–Based Reasoners is fully explained in [Davis and Hamscher, 1988] as well as [Hassberger *et al.*, 1990]. Finally, the different approaches to Qualitative Simulation are elaborated in [de Kleer and Brown, 1982, 1984; de Kleer and Williams, 1987, 1989; Forbus, 1981, 1984, 1985; Forbus and de Kleer, 1993; Hayes, 1979, 1985a, 1985b; Kuipers, 1982, 1984, 1986, 1989a, 1989b]. Some good summaries of these Qualitative Simulation approaches can be found in [Bobrow, 1984; Fishwick, 1988, 1991; Hobbs and Moore, 1985; Weld and de Kleer, 1990].

Two main characteristics distinguishing the qualitative methods used for fault detection and troubleshooting are:

a) their management of uncertainty, and

b) their diagnostic reasoning approach, i.e., whether they employ *shallow* or *deep* diagnostic reasoning.

The management of uncertainty constitutes one of the most prominent features of qualitative methodologies and is responsible for the growing interest in their applications.

### 2.4.1 Management of Uncertainty

The management of the uncertainty introduced by all these methodologies has evolved from systems based on first order logic, Boolean algebra, and Bayesian conditional probabilities to systems based on Bayesian subjective inference, certainty factors, theories of evidence (credibility and plausibility), and possibilistic measures (fuzzy logic). For a good summary on uncertainty measures look at [López de Mántaras, 1990], and for advances in fuzzy logic look at [Klir and Yuan, 1995; López de Mántaras and Godó, 1993]. In the sequel, the major exponents of these theories are being outlined.

**Certainty Factors.** This approach was introduced in the MYCIN Expert System [Buchanan and Shortliffe, 1984]. Certainty factors are values between $-1$ (false) and $+1$ (true), or any other symmetric combination of values. These factors are used to deduce the certainty of dependent knowledge. The *AND* operator is defined as the minimum operator, and the *OR* operator is defined as the maximum operator. For example, if the certainty of a fact $A$ is given as 0.3 whereas the certainty of a fact $B$ is 0.7, and the rule to be applied to deduce fact $C$ is *IF A AND B THEN C*, then the certainty of fact $C$ is calculated as:

$$CF(C) = \min(0.3, 0.7) = 0.3 \tag{2.19}$$

If more rules support the same result $R$, the resulting certainty factor is calculated as:

$$C_R = \begin{cases} C_{R1} + C_{R2} - C_{R1}C_{R2} & if\ C_{R1}, C_{R2}\ are\ both\ positive \\ C_{R1} + C_{R2} + C_{R1}C_{R2} & if\ C_{R1}, C_{R2}\ are\ both\ negative \\ \frac{C_{R1}+C_{R2}}{1-\min(|C_{R1}|,|C_{R2}|)} & otherwise \end{cases} \tag{2.20}$$

This theory has no real mathematical foundations, but has proven quite useful in MYCIN and some other systems.

**Theory of Evidence.** This approach considers a set of propositions and assigns to each of them an interval

$$[Belief, Plausibility]$$

in which the degree of belief must lie. *Belief* measures the strength of the evidence in favor of a set of propositions, i.e., provides a measure of belief in some hypothesis given some evidence. It ranges from 0, indicating no evidence, to 1, denoting certainty. *Plausibility* also ranges from 0 to 1 and measures the extent to which evidence in favor of *NOT* $p$ leaves room for belief in $p$.

A weight $m(p)$ is assigned to a subset $p$ of the set $P$ of hypotheses, and measures its amount of belief. The subset $p$ is known as the subset of focal propositions [Dempster, 1968; Shafer, 1976]. The focal propositions are a representation of the available knowledge. Each focal proposition $F$ is defined as $F = \{p \in P \mid m(p) > 0\}$. To evaluate the truth of a proposition $p$ given the proposition $q$, $F$ and $m(p)$, two subsets of focal propositions should be considered: the subset of focal propositions that are not incompatible with $p$:

$$Pos(p) = \{q \in F \mid p \text{ AND } q \neq 0\} \tag{2.21}$$

and the subset of focal propositions that imply the truth of $p$:

$$Nec(p) = \{q \in F \mid q \rightarrow p = 1\} \tag{2.22}$$

These two subsets are used to define the two new measures of certainty: *belief* and *plausibility* in the following fashion:

$$Bl(p) = \sum \{m(q) \mid p \text{ AND } q \in Nec(p)\} \tag{2.23}$$

$$Pl(p) = \sum \{m(q) \in F \mid p \in Pos(p)\} \tag{2.24}$$

The *AND* and *OR* operators are defined in the following way:

$$Bl(p \text{ OR } q) \geq Bl(p) + Bl(q) - Bl(p \text{ AND } q) \tag{2.25}$$

$$Pl(p \text{ AND } q) \leq Pl(p) + Pl(q) - Pl(p \text{ OR } q) \tag{2.26}$$

The following relations hold between the *belief* in and the *plausibility* of a proposition $p$:

$$Bl(p) \leq Pl(p) \qquad , \qquad Pl(p) = 1 - Bl(\text{NOT } p) \tag{2.27}$$

The computational complexity of the belief and plausibility equations is exponential in the number of propositions. Hence this approach must be used cautiously, and the high computational demands of this approach may prevent it from being used in large engineering applications.

**Fuzzy Logic.** This approach was introduced as an extension to conventional set theory [Zadeh, 1978]. While traditional set theory defines set membership as a boolean predicate, fuzzy set theory allows the representation of set memberships as possibility distributions. Once the set membership has been redefined in this way, it is possible to define a reasoning system based on logics for combining possibility distributions. Thus, the degree of membership to a specific set is defined by a membership function $\mu$. Such a function has values in the range $[0, 1]$, defined in such a way that a value $u$:

$$\mu(u) = \begin{cases} 1 & \text{if } u \text{ is at the maximum value of } \mu \\ 0 & \text{if } u \text{ is not in } \mu \\ \text{a value in } [0,1] & \text{otherwise} \end{cases} \tag{2.28}$$

Every value of a variable is coupled to a fuzzy value via that membership function. Normally, membership functions are triangularly shaped, but trapezoidally and Gaussian–shaped membership functions can also be found. In Chapter 3, it will be shown how this fuzzy approach can be implemented using Gaussian–shaped membership functions.

Fuzzy reasoning for the operators $AND$ and $OR$ can be implemented in many ways. Following [Klir and Yuan, 1995], the $AND$ operator for two different facts or observations $p$ and $q$, is defined as the minimum operator $min(p, q)$, and the $OR$ operator is defined as the maximum operator $max(p, q)$. Following the "probabilistic" approach, the $AND$ operator is defined as $p \cdot q$, and the $OR$ operator is defined as $(p + q) - (p \cdot q)$. A third implementation of the $AND$ and $OR$ operators, known as the Lukasiewicz approach, defines the $AND$ operator as $max(p + q - 1, 0)$, and the $OR$ operator as $min(p + q, 1)$.

One of the advantages of the fuzzy logic approach is its theoretical foundation. Normal (non–fuzzy) sets can be consider as a special case of the fuzzy approach. Fuzzy logic has been applied in control and diagnosis systems for devices as diverse as aircrafts, trains, nuclear power plants, and washing machines.

The Artificial Intelligence community has not reached any consensus on which of the aforementioned methods is the best for expressing uncertainty in the knowledge used for reasoning. However, the present trend is towards the use of fuzzy logic. Many engineering applications have been developed, in which fuzzy logic has been successfully employed to express and handle uncertainty of knowledge.

## 2.4.2 Reasoning Approaches

### 2.4.2.1 Shallow Diagnostic Reasoning

Shallow diagnostic reasoning techniques are appropriate for evidence–driven systems as they are typically found in medical diagnosis. Shallow reasoning is based on shallow knowledge that consists of inference rules that can be fired on the basis of a set of pre–specified symptoms, establishing a static relationship between fault symptoms and system malfunctions. This means that shallow diagnostic reasoning can only be applied successfully, if a direct relationship between system behavior irregularities and system faults can be established. The production rules try to mimic the deductive process of the expert, linking symptoms to faults, but they normally do not explore why a certain anomaly might have occurred, and how future occurrences of the same fault might be prevented. This is what often happens in medicine, where the underlying mechanisms can rarely be described, and even where they can, they are in all likelihood uncontrollable anyway. Very often these rules are shortcuts of a deeper understanding of the problem domain, and loosely model the decision making process of how a human expert would model the domain [Tzafestas, 1989]. The classical example of a shallow expert system is MYCIN.

The main characteristics of the shallow reasoning approach are:

- Difficult knowledge acquisition.
- Unstructured knowledge requirements.
- Excessive number of rules.
- Knowledge base highly specialized to the individual process.
- Diagnosis success not guaranteed.

These disadvantages can be partly overcome by decomposing the problem into smaller problems, either in a hierarchical manner or according to unit operations.

Two very common ways of carrying out the fault diagnosis following a shallow diagnostic reasoning approach are:

i) **Fault Dictionaries.** They consist of a list of causes and effects. Diagnosis is performed by looking up the effects in a cause–effect lookup table to determine the most likely cause given the observed effects [Less, 1983].

ii) **Diagnostic Trees.** Also known as fault trees, they provide a simple and yet effective way for writing down the sequence of tests and conclusions needed to guide a diagnosis, while restricting the search to proceed along different diagnosis paths. Diagnostic trees are widely used off–line in chemical industries and nuclear power plants to discover possible combinations of small faults that may lead to an emergency [Hassberger *et al.*, 1990].

Since both methods use lookup tables, and since all potential faults have to be observed and characterized in advance, the resulting lookup table will almost always have a large number of entries. Thus the diagnosis procedure becomes difficult and time consuming for large–scale systems, and out of the question for on–line real–time applications.

Many of these disadvantages of the shallow reasoning approach are avoided in the deep reasoning approach, which makes use of deep models and is particularly suitable for diagnosing technological systems.

#### 2.4.2.2 Deep Diagnostic Reasoning

The deep reasoning approach is based predominantly on a structural model of the problem domain, the so–called *deep model*. Deep models are models that can derive their own behavior for a given set of parameters and excitations, and predict what should be the effects of changes in them. Methodologies that are based on deep knowledge attempt to compute the underlying governing principles of the process (or domain) explicitly, and the need to predict every possible fault scenario or to use precomputed rules is eliminated. This approach leads to qualitative tools that are able to handle a wider range of problem types and larger problem domains.

Three of the more widely used deep diagnostic reasoning techniques, from the point of view of fault detection and troubleshooting, are outlined below:

i) **Causal search and hypothesis testing technique.** It is used to trace process malfunctions to their source. Causality is usually represented by signed directed graphs (so–called digraphs), the nodes of which represent state variables, alarm conditions, or fault origins, whereas the branches represent the influence between nodes. With digraphs, the intensity of the fault, the time delays, and the probabilities of fault propagation along the branches can be represented. All the information contained in a digraph can then be expressed in the form of rules. The causal search method is known as hypothesis testing, since it follows the usual human diagnosis path, i.e., a cause for a system malfunction is postulated, the symptoms of the postulated fault are determined, and the result is compared with the system observations. Digraph–based qualitative diagnostic techniques have become popular, because little information is required to construct the digraph and carry on the diagnosis; however, qualitative simulation of the hypothesized behavior is required [Shiozaki *et al.*, 1985].

ii) **Constraint violation technique.** This technique belongs to the approaches of diagnostic reasoning from behavior to structure or, more accurately, from misbehavior to structural defect [Davis and Shrobe, 1983]. The constraints play the role of rules of behavior. Qualitative simulation is required to predict the values of the constraints that will then be compared against the observed values. The advantages of this approach are: a) it allows the system to deal with a wide range of faults, since it defines malfunction as anything that does not match the expected behavior, b) it allows a natural use of hierarchical descriptions, and c) it allows the system to yield symptom information about malfunctions. The main disadvantage is that, in a multicomponent system, the approach has been shown to be incapable of locating the fault.

iii) **Governing equations technique.** This approach is directly related to the previously explained constraint violation approach. It was originally developed for the purpose of locating faults in chemical processes, but it has successfully been applied to various situations where associations of each quantitative constraint equation of the system with a set of faults that suffice to cause violation of the constraint can be found [Kramer, 1986]. It requires a translation from quantitative constraint equations to qualitative cause–effect relationships, i.e., behavior rules.

Expert Systems and almost all qualitative fault detection techniques are based on shallow or deep reasoning approaches. Let us explain how these fault detection techniques work.

### 2.4.3 Knowledge–Based Systems

Knowledge–based systems for fault detection and troubleshooting are built by accumulating the experience of expert troubleshooters in the form of empirical or directed associations, rules that associate symptoms with underlying faults based on the experience with the system under study. Expert systems have been widely used to perform symptoms–guided diagnosis, and recently they are being used to perform supervisory and control tasks in real–time engineering systems. Such Expert Systems are composed of:

a) **Knowledge Base.** It may include facts, assumptions, and heuristic rules used to relate each condition–action pair with the facts and assumptions about a problem situation, constituting a single piece of problem–solving knowledge.

b) **Inference Engine.** It carries out the whole inference process. It normally includes the inference strategy (data driven or goal driven), the search strategy, and an uncertainty management strategy.

c) **Explanation Module.** It is used to translate the solution path that was followed by the inference engine to an explanation of why this path was followed, i.e., it explains why a solution is what it is.

d) **User Interface.** It serves to communicate with the user providing him or her with the results of the inference process, the explanation of how those results were obtained, and the confidence associated with them.

e) **Process Interface.** It serves to communicate with the process in a supervisory or control role, taking care of data handling, and interfaces to other programs for necessary calculations.

Diagnostic expert systems were first used in the medical domain, and were designed following the *shallow reasoning approach*, since the mechanisms that lead to a disease are usually difficult to describe and may even be completely unknown [Pople, 1982]. One of the earliest medical expert systems for the diagnosis of bacterial infections in blood is MYCIN, which uses production rules and backward chained inference. It has a modest–sized knowledge base containing 700 rules [Shortfile, 1976]. NEOMYCIN, as the name indicates, is a new version of MYCIN implemented using a distributed problem–solving methodology [Clansey and Letsinger, 1981]. EMYCIN standing for "Essential MYCIN," is the expert system shell (core) resulting from MYCIN when removing the knowledge base [Van Melle *et al.,* 1984], to be applied in other domains. Presently, there exist many expert systems for medical diagnostic applications, only some of which follow the shallow reasoning approach, whereas others make use of deep reasoning, such as AI/RHEUM for rheumatology, ANNA for heart arrythmias, CASNET for glaucoma diseases, CENTAUR for pulmonary function test interpretation, the EEG Analysis System for electro–encephalogram analysis [Tzafestas, 1989].

Expert Systems have advanced from providing symptom–guided diagnosis to performing supervisory and/or control functions. In this context, some of the most representative among the available Expert Systems shells for diagnosing faults in technological devices, systems, and processes (without real–time characteristics) are:

**DART.** Diagnostic Assistance Reference Tool, is a device–independent diagnostic system that works directly from design descriptions rather than symptom–directed fault rules [Genesereth, 1984]. This expert system is intended for use in conjunction with a tester that can manipulate and observe a malfunctioning device. The diagnostic engine receives from the tester a description of an observed malfunction, prescribes tests, and accepts the results, and finally locates the faulty components responsible for the fault. The DART system does not follow a shallow reasoning approach associating symptoms with possible diseases, i.e., MYCIN–like rules. Instead, it operates on deep information about the *intended structure* (the device subsystems and their interconnections) and the *expected behavior* (equations, rules, and procedures relating inputs and outputs of the device at hand). DART has been applied to electronic circuit troubleshooting and also in non–electronic devices such as the cooling system of a nuclear reactor.

**FOREST.** [Finin *et al.*, 1984] This system supplements the fault detection and isolation capabilities of current automatic test equipment diagnostic software. It makes use of experimental rules of thumb provided by experts, deep knowledge about circuit diagrams, and general troubleshooting principles. FOREST is encoded in PROLOG with rules using certainty factors, and offers a user–friendly explanation facility.

**LES.** Lockheed Expert System, is fully described in [Perkins and Laffey, 1984; Laffey *et al.*, 1986]. It is a shell similar to EMYCIN, but more powerful because it uses a deep–structural description of the device in its troubleshooting, and allows the operator to explicitly control the reasoning process. LES has been successfully applied to a large signal switching network.

**FIS.** Fault Isolation System [Pipitone, 1986, 1990]. It is written in LISP, and was designed primarily to diagnose analog systems, isolating faults at the level of amplifiers, power supplies, and other large components. The diagnosis is based on the computation of the fault probabilities of the individual modules after each test. FIS can be applied to fault diagnosis and isolation in systems containing mechanical, hydraulic, optical, and other types of large components.

**RELSHELL.** It is an expert system shell for fault diagnosis of multiparametric, gradually deteriorating technological systems. It uses a hybrid rule/frame based representation of knowledge, fuzzy arithmetic, and logical operations [Gazdik, 1987; Gazdik *et al.*, 1993].

Recently, Expert Systems are being used to supervise and/or control systems in which real–time considerations are involved. To this end, the expertise of process operators and control system designers has to be integrated with the time–varying information obtained from the process through measurements. When knowledge is evaluated, for example a rule, the *consequent* part is set true if the *antecedent* part was satisfied. The consequent part will be stored as a fact or new knowledge in the data base, and will remain there until it is explicitly removed. However, in supervision and/or control of complex engineering processes, the Expert System is dealing with *time–dependent information*, i.e., previously acquired and asserted knowledge can lose its validity at some point in time, after which itself as well as all the conclusions that were drawn on the basis of this knowledge should be deactivated [Laffey *et al.*, 1988]. Thus, a *Truth Maintenance System* is needed to keep track of the temporal information and of every inference step in the reasoning process, in order to preserve the consistency of the data base [Sarjoughian, 1995].

The use of *temporal logic* is an important issue in real–time Expert Systems. Time and its properties must be represented explicitly, either based on time points [Perkins and Austin, 1990] or on time intervals [Allen, 1984]. The reasoning processes associated with temporal knowledge are considerably more involved than those that only deal with static knowledge. This has led to a rethinking of the role and utility of Expert System technology in fault diagnosis, and has further advanced the use of deep knowledge in Expert System reasoning. Considerations about the real–time requirements of Expert Systems can be found in [Laffey *et al.*, 1988; Krijgsman, 1993]. Recent real–time Expert Systems shells architectures for fault detection are:

**DICE**. Direct Intelligent Control Expert System [Krijgsman, 1992], for control of highly nonlinear systems with unknown parameters. This shell is based on the RETE algorithm [Forgy, 1982], and permits fuzzy rules, direct feedback, model following, set point control, and adaptation.

**REAKT**. This object oriented shell [REAKT, 1990] is intended for multipurpose applications. Its main feature is that it is based on a temporal and concurrent blackboard architecture [Hayes–Roth, 1990] where classes, objects and knowledge bases are managed. This permits parallel communication between otherwise independent objects in the databases.

**G2**. G2 [Gensym, 1990] is one of the most common commercial shells for fault detection now available. It is an object–oriented shell with a very enhanced user interface that permits access to every variable and object of the system. The Expert System is used in combination with any continuous simulation technique in such a way that, when a malfunction occurs, the simulation triggers the Expert System, and a session is started to reason about the detected fault without stopping the simulation process.

Besides from Expert systems shells, there are lots of specific application fault detection real–time Expert Systems. Examples of them are: an Expert System with embedded Analytical Redundancy components and Automated Statistical Diagnosis[8] methods for fault detection and isolation in process plants [Fathi *et al.*, 1991]; a combination between an Expert System and a Neural Network for pressure early warning functions in dangerous liquids closed tanks [Rengaswamy and Venkatasubramanian, 1992]; an Expert System using Qualitative Process Theory for on–line reasoning about the time profiles (derivatives) of electric power plant variables [Konstantinov and Yoshida, 1993]; DMP: Diagnostic Model Procesor, an Expert System based methodology for thermal fault analysis of PWR nuclear reactors [Surgenor and Jofriet, 1992]; FDD: Fuzzy, Deep Knowledge–Based Fault Diagnosis System, for application to continuous stirred tank reactors [Terpstra *et al.*, 1992]; MFM: Multilevel Flow Modeling, an Expert System based on the G2 real–time shell that uses a description of topological and means–end relations between objects expressed in a graphical language, and intended for fault diagnosis of flowing liquid processes [Larsson, 1991].

In spite of being the most widely used qualitative fault detection technique, Expert Systems still present serious drawbacks. Among them, their domain dependence and sheer size are two of the most important ones. The general–purpose domain–independent Expert System shells do not contain the domain knowledge; in reality, gathering and encoding the domain–specific knowledge is usually much more formidable a task than writing a general–purpose inference engine. Moreover, Expert Systems for complex applications are usually

[8]Both techniques: Analytical Redundancy and Automated Statistical Diagnosis were explained in the Quantitative Fault Detection section of this same chapter.

designed from scratch, not even making use of one of the available shells for reasons of efficiency. Expert Systems suffer also from the often frightening size of their knowledge bases, and usually have problems when dealing with missing or inconsistent information.

### 2.4.4 Connectionist Systems

In contrast to Expert Systems, in which knowledge is expressed explicitly in a knowledge base and where this knowledge base is separated from the inference strategy, the connectionist approach presents a full integration of the knowledge and the knowledge processing technique. Artificial Neural Networks[9] are learning systems based on unstructured knowledge, usually operating within a numerical framework [Wells, 1992]. They are much simplified models of biological neurons trying to mimic their behavior.

The soma of an artificial neuron is often represented by a weighted sum of the input signals, followed by an appropriate non–linear function. This function is called the activation function and determines the output of the neuron. The activation function is usually related to a hard limiter or sigmoidal function. An artificial neuron of this kind having a binary threshold as output function is known as *Perceptron* [Minsky and Papert, 1969], a basic element of many Neural Network applications reported in the literature.

The combination of the input signal weights and the activation function is referred to as a *node*. A Neural Network can be defined as a set of interconnected nodes. The complete network is specified by three features [Krijgsman, 1993]:

i) **Network topology**. A Neural Network consists of a number of interconnected nodes and is usually organized into layers. Such networks can be characterized as either single–layer networks or multi–layer networks. A typical Neural Network has an input layer, an output layer, and an optimal number of hidden layers in between. These hidden layers are used for storing information, and their number is strongly application dependent. The connections of the nodes can be such that the information flows in one direction only (from input to output), or in both directions introducing feedback.

ii) **Control algorithm**. Determines the way in which the network is executed, and also dictates the communication between the network and

[9] Neural Networks from now on.

its environment. For the case of feedback networks with intermediate layers, a convergence is normally involved. The sequence of layers is executed until the convergence criteria are met.

iii) **Learning strategy**. The weights on the interconnections in the network are being modified in accordance with a pre–specified learning rule, unless they are known in advance (so–called network synthesis). Learning laws determine how the network adjusts its weights by using error functions of several kinds. According to [Wells, 1992], there are basically three learning strategies used in Neural Networks. They are:

- **Supervised Learning**: which occurs when the network is supplied with the correct input and output values, and the weight adjustments performed by the network are based on the error between the desired and the calculated output values, using one among a set of known *error minimization algorithms*, such as, for example, *backpropagation*, or the *Delta* rule.
- **Unsupervised Learning**: which occurs when the network is only provided with the input values, and the weight adjustments are based only on the input values and the current network output values following some *correlation rule* between the involved neurons. This type of learning is typical of associative memory and competitive architectures.
- **Reinforced Learning**: which occurs when the network is only provided with the input values, and the weight adjustments are based only on the input values and the current network output value following a global performance measure that is being fed back to the network. The weights are adjusted following this reinforcement signal. This strategy is somewhere in the middle between supervised and unsupervised learning.

The use of Neural Networks can be divided into two main phases: the learning phase and the operating phase. In the learning phase, which normally occurs while the network is off–line, the user provides the network with a number of input patterns and possibly some output patterns also, and the network adjusts its weights until convergence. During the operating phase, the network computes an output given any input pattern. If feedback is present, the network can adjust its weights during operation also. The time domain in which the network operates can be either continuous or discrete, depending on the network architecture and the application. Three of the most commonly used network architectures in fault detection and troubleshooting are:

- **Multilayer Feedforward Networks.** They have two or more layers linked by strictly feedforward connections, i.e., connections which pass signals forward through successive layers but never back to preceeding ones. Two early models with just two layers were the Perceptron and Adaline networks, that proved capable of performing simple linear classification tasks. Usually this kind of networks makes use of supervised learning algorithms, in particular *backpropagation*, which extends the original network capabilities to more complex non–linear problems. The success achieved with multilayer backpropagation networks in many application areas has made this the most widely used network architecture [Wells, 1992].

- **Recurrent Networks.** This networks permit feedback loops that circulate information back through the same or previous layers. Consequently, when input information is presented to the network, a circular process of neural activity *resonance* is caused wherein the same layers of neurons are activated repeatedly. Since each neuron is fully interconnected with all other neurons, the input signals to each neuron can be input signals to the network, or feedback signals from other neurons. This network allows continuous modification of the weights, characterizing it as an *adaptive network*. Examples of such architectures are: the single–layered Hopfield network [Hopfield, 1982] with functions of autoassociative memory, the two–layered ART and BAM networks, and the multi–layered Jordan and Elman networks [Kröse and Van der Smagt, 1991].

- **Kohonen Network.** Its structure is very similar to that of the Hopfield network. It is composed of a single layer with fully interconnected neurons. Its main characteristic is that it has two different sets of weights. The first one is used to control the interconnections between neurons (feedback information), whereas the second, which can be continuously modified, is used to process the input information. Usually, this kind of network uses unsupervised learning algorithms [Kohonen, 1984].

Neural Networks have been successfully applied to various application domains, such as: function approximation, identification of dynamic systems, and several control strategies including predictive, adaptive, supervisory, and fuzzy control. Recently, Neural Networks have also been applied to fault monitoring and troubleshooting of large–scale systems, for example, in nuclear power plants [Bartlett, 1991; Cordes, 1990; Jouse, 1989, 1992; Jouse and Williams, 1991], in chemical plants [Lee and Park, 1992; Rojas–Guzman and Kramer, 1992], in motion supervision [Kawato, 1990; Houk *et al.*, 1990], and airplane takeoff and landing maneuvers [Jorgensen and Schley, 1990]. General overviews can be found in [Foss and Johansen, 1992; Himmelblau, 1992; Schenker and Agarwal, 1992].

There exist many troubleshooting methodologies that combine Neural Networks with some of the other qualitative or quantitative techniques. In [Rengaswamy and Venkatasubramanian, 1992], an integrated framework for process monitoring, diagnosis, and control by means of the combination of knowledge–based and Neural Network systems is presented. In [Krijgsman, 1993], an advanced lookup table technique strongly related to the field of neural networks, the Cerebellar Model Articulation Controller (CMAC), is presented.

Neural Networks are not so widely used as Expert Systems for fault detection and troubleshooting purposes; however, they are gaining progressive importance due to their simulation capabilities. There are fields in with Neural Networks are already widely such as: Robotics [Sanderson, 1990], autonomous vehicles [Herman *et al.*, 1990], and motion and trajectory planning [Nguyen and Widrow, 1990]. The principal disadvantages of Neural Networks when applied to large–scale systems are that they do not provide much insight into their internal reasoning processes, and that they do not offer any self–assessment capabilities, i.e., they do not check whether what they propose is "reasonable" (fully consistent with the available information).

### 2.4.5 Model–Based Systems

These systems are a combination of qualitative simulation and some other techniques (for example, those explained in the shallow and deep diagnostic reasoning sections), depending on the way in which the model is derived [Feyock, 1987]. The model–based approaches are device independent. They use knowledge about structure and behavior, and provide methodical coverage, because the model–building process supplies a way of systematically enumerating the required knowledge. The main problems are how to create adequate models of complex non–linear behavior, and execution speed under real–time constraints. Few model–based systems offer abstraction capabilities that would allow the system to reach a conclusion faster than by searching through the entire search space.

When applied to large–scale systems, the combination of qualitative simulation and model–based approaches seems to be very promising and powerful. This methodology has a number of striking properties that make it attractive for use in a) fault detection, b) fault isolation, c) fault characterization, and d) fault diagnosis in an automated fault monitoring system.

In the model–based approaches, the knowledge can be captured using either deductive techniques employing available meta–knowledge and reasoning on the basis of first principles, or inductive techniques such as neural networks or inductive reasoners combined with reasoning on the basis of structure and/or behavior. Numerous such systems have been built in the 70s and 80s exploring a variety of problem domains such as electronic circuits [de Kleer, 1987; Davis and Hamscher, 1988], hydraulic systems [Pan, 1986], neurophysiology [First *et al.*, 1982], medicine [Patil *et al.*, 1981], aerospace [Scarl *et al.*, 1985, 1988; Enand and Scarl, 1989; Scarl, 1991a, 1991b, 1993], and dynamic systems [Dvorak and Kuipers, 1989]. The major challenge with all these approaches is making them adhere to real–time constraints.

Since the basis of the model–based approach is the interaction between observation and prediction, a fundamental assumption is that, if the model is correct, all the discrepancies between observation and prediction arise from defects in the studied system. The tasks of a model–based reasoner applied to diagnosis and troubleshooting can be summarized as [Davis and Hamscher, 1988]:

i) to determine which of the system's components could have failed in a way that accounts for all the discrepancies observed,

ii) to characterize failures in a way that they can be identified uniquely, i.e., distinguished unambiguously from other failures, and

iii) to learn the possible behavioral modes of the system to such an extent that one may be able to identify potential mishap before it ever occurs for the purpose of preventing potential faults from ever happening.

To achieve these goals, model–based diagnosis needs some or all of the following items:

- Observations of the system, normally consisting of measurements of its inputs and outputs.
- A description of the system's internal structure, normally given as a listing of its components and their interconnections.
- A description of the behavior of each component, and their relationships among each other.

The major obstacle in designing model–based systems is the modeling phase. Whereas the problem of how to apply model–based reasoning, once the model is available, has been solved to a satisfactory degree, the problem of modeling complex non–linear behaviors in a systematic and adequate fashion remains largely unsolved. Consequently, the other partner of the combined methodology, i.e., qualitative simulation, will have to provide some help for how to create models and how to select the appropriate one for the task at hand.

### 2.4.6 Qualitative Simulation

In recent years, several different methodologies for qualitative description of continuous–time processes have emerged. It all started with the "Naïve Physics" approach proposed by Hayes [Hayes, 1979, 1985a]. Its main purpose was to capture physical process knowledge and common–sense knowledge in a way that can be treated by computers. Since then, research in qualitative simulation has followed two different lines depending on the desired goal: in the first one, the objective is the qualitative prediction of behavior, whereas in the second, the objective is to provide a reasonable cause–effect explanation. The main approaches that evolved from the original Naïve Physics concept are:

i) **Common–Sense Reasoning.** Proposed by many authors, particularly by [Carbonell and Minton, 1985], common–sense reasoning tries to include common–sense knowledge in the simulation of physical systems.

ii) **Qualitative Process Theory.** Proposed by [Forbus, 1981; Forbus and Stevens, 1981], this approach provides a language for specifying and describing processes and their effects, working with the assumption that any change in a physical system can be expressed as a qualitative process. Refinements of this methodology are presented in [Forbus, 1984, 1985].

iii) **Qualitative Differential Equations.** Proposed by [Brauer and Nobel, 1969] and resumed by [Kuipers, 1982], this methodology provides a qualitative description of system behavior from a qualitative description of the system structure. This description is expressed by means of qualitative differential equations. Improved results using this technique are presented in [Kuipers, 1984, 1986, 1989a, 1989b].

iv) **Envisionment and the Qualitative Physics of Confluences.** Originally proposed by [de Kleer and Brown, 1982] and enhanced by [de

Kleer and Brown, 1984], this world view infers the qualitative behavior of a physical system by creating qualitative physical laws for each of its parts.

Since the "Naïve Physics Manifesto" [Hayes, 1979], several different techniques have been developed without significant success, except for those that combine qualitative simulation with other methodologies, such as for example, fault dictionaries. The main problem with qualitative simulation is its inherent ambiguity in reasoning. For these reasons it was not possible to adapt any of these methodologies to the problem of fault detection and troubleshooting in large–scale systems. The main difficulties were:

1. So far, it seems impossible to avoid the explosion of the manifold of feasible qualitative solutions. Consequently, these techniques can only be applied to very simple situations. The methods don't scale up properly, and are practically useless when dealing with large–scale systems.

2. Frequently, some of the solutions offered do not have any physical meaning. The self–assessment capabilities of these methods are thus limited.

3. The strongly symbolic nature of the qualitative models does not allow to represent the independent variable *time* in a quantitative fashion. It is unclear how such models can be coupled to quantitative models. Consequently, these methods do not lend themselves to a description at varying levels of granularity consistent with the knowledge available for different subsystems.

4. Although these models are capable of capturing uncertainty and informality with respect to the interrelations between system variables, they require a deep and explicit knowledge of the internal structure of the system to be modeled. Such knowledge may or may not be available.

5. All of these techniques suffer basically from the same shortcomings. Although each methodology uses a different terminology, all of them are in fact dealing with the same issues. They are not fundamentally different methods, only different dialects of one and the same basic methodology.

Summarizing, all these methodologies are excellently suited for emulating higher–level human reasoning processes by incorporating common–sense knowledge; however, none of them has the capacity of being useful for solving complex engineering problems.

What is required, is a methodology, based on the same principles as those mentioned earlier, that is capable of qualitatively describing and predicting the behavior of a physical system, yet without the inherent shortcomings of the aforementioned qualitative approaches; i.e., a method that is capable of interacting with quantitative subsystems (such a method would need to treat *time* as a quantitative rather than qualitative variable); that is able to convert quantitative variables into qualitative variables with minimal loss of information in the conversion processes; and finally, a method that can produce a good estimate of quantitative behavior from qualitative reasoning. In short, a qualitative simulation methodology is needed, whose knowledge representation structures, pattern recognition algorithms, and knowledge inferencing mechanisms are placed somewhere in between the pure qualitative and the pure quantitative modeling and simulation techniques.

On these grounds, *Fuzzy Inductive Reasoning*, a model–based approach for qualitative modeling and simulation, originally derived from General Systems Theory, is most promising, because it overcomes many of the earlier mentioned problems. The methodology will be fully explained in the next chapter.

Some of the principal advantages of Fuzzy Inductive Reasoning are the following:

1. Inductive reasoning allows qualitative models to treat time as a continuous (quantitative) variable. This is of primary importance if modeling and simulation of mixed quantitative and qualitative systems is to be attempted.

2. The technique can be applied to any system available to experimentation and observation. Inductive reasoning is fully based on behavior, thus, there is no need for knowing the internal structure of the system.

3. The methodology contains an inherent model validation mechanism inside its qualitative simulation engine preventing it from reaching conclusions that are not justifiable on the basis of the available facts.

4. Inductive reasoners operate internally in a qualitative fashion just like knowledge–based reasoners. Therefore, it is possible to apply meta–knowledge to improve the performance and quality of the inference engine, and it is also possible to trace back the reasoning process if desired.

Although *Fuzzy Inductive Reasoning* is the driving tool of the overall methodology advocated in this thesis, *Reconstruction Analysis*, a tool for subsystem identification and variable selection through causality analysis and refinement procedures, will also be used to meet the requirements imposed by the need to cope with large–scale systems.

# Chapter 3

# Fuzzy Inductive Reasoning

## 3.1 Introduction

Inductive Reasoning is a model–based approach for qualitative modeling and simulation originally derived from General Systems Theory. The research on inductive reasoning has had three different stages over the past 16 years. The basic idea was originally conceived by George J. Klir at the State University of New York at Binghamton, as part of a General Systems Problem Solver. The first implementation was carried out by one of Klir's students by the name of Hugo J. Uyttenhove, and was first published as his Ph.D. dissertation [Uyttenhove, 1978]. He named his inductive reasoning tool SAPS, which stands for Systems Approach Problem Solver. The earliest article in which the SAPS methodology was presented is [Uyttenhove, 1981].

In a second stage, the methodology was re–elaborated and reimplemented by François E. Cellier and three of his students at the University of Arizona at Tucson [Cellier and Yandell, 1987; Cellier, 1987]. They called the new methodology the Systems Approach Problem Solver II (SAPS–II). The first time SAPS–II was applied to a non–trivial engineering problem is presented in a Master Thesis of one of Cellier's students [Vesanterä, 1988]. One of the main features included by this research group was a fuzzy extension to the original tool. The fuzzy SAPS–II was first published in [Li and Cellier, 1990], and is fully explained in [Cellier, 1991a].

The third stage has been developed at the Universitat Politècnica de Catalunya UPC (Polytechnic University of Catalonia) at Barcelona, in a joint project between the Cybernetics Institute and the Control and Systems Engineering Department, in cooperation with the Electric and Computer Engineering Department of the University of Arizona. Although a lot of research had previously been done on inductive reasoning, at the beginning of this third stage, the SAPS–II methodology was still a somehow rudimentary and heuristic tool that had not yet been applied to any real–life problem. Thus, the main purpose of this third stage was that of improving the SAPS–II methodology and implementation in such a way that it could be applied to solve complex real–life problems. Three Ph.D. students, directed jointly by Françcois E. Cellier from the University of Arizona and Rafael M. Huber from the Cybernetics Institute at the UPC, were responsible for upgrading, improving, and applying SAPS–II to solving different engineering and biomedical problems. The credit for upgrading and improving the SAPS–II methodology is shared by the three Ph.D. students, and consequently a text similar to this chapter is present in all three dissertations. The three Ph.D. theses are [Nebot, 1994; Mugica, 1995] and the present thesis.

A number of papers on the improved SAPS–II methodology have already been published. Different defuzzification methods for inductive reasoning are presented and compared in [Mugica and Cellier, 1993]. A comparison between a crisp inductive reasoner and a fuzzy inductive reasoner can be found in [de Albornoz and Cellier, 1993a; de Albornoz and Cellier, 1994]. The limitations of predictability of behavior with inductive reasoning are stated in [Nebot *et al.*, 1994].

In this chapter, a full description of the Fuzzy Inductive Reasoning (FIR) methodology shall be provided. This will include a discussion of its origins from General Systems Theory, and precise details shall be given of how the tool works, of what have been the improvements made to the original tool that allow FIR to tackle, in a qualitative fashion, quantitative complex real–life problems. To this end, an application example shall also be included with two purposes in mind: to show all phases of the FIR methodology, and to demonstrate the practicality of combining quantitative and qualitative simulation.

## 3.2 Systems Problem Solving

All activities involved in the study of the properties characterizing a system and the problems that emanate from this characterization are now becoming identified with the general term of *system science*. System Science is an interdisciplinary science. For this reason, it should be viewed as a new dimension in science rather than as a new science. Its main objective is to provide potential users in various disciplines and problem areas with methodological tools for studying relational properties of various classes of systems and for solving systems problems, i.e., problems that deal with the relational aspects of systems [Klir, 1985a].

System problem solving is the discipline of systems science in charge of the development of methodological tools for all recognized types of systems problems. As far as the characteristics of some kind of system can be generalized, that system can be treated as a *general system*. Then, a methodology to solve problems related to that particular kind of system can be called a *General Systems Problem Solver* (GSPS). In Klir's own words: "a general systems problem solver is a conceptual framework through which types of systems problems are defined together with methodological tools for solving problems of these types." As mentioned earlier, the inductive reasoning technique was originally developed as part of Klir's General Systems Problem Solver called Systems Approach Problem Solver (SAPS).

For us, a system is a set of properties and relations between them that can be characterized. This characterization depends on the level of knowledge abstraction of the system, i.e., it depends on an epistemological hierarchy of systems.

### 3.2.1 Epistemological Levels of Systems

An epistemological hierarchy of systems must be considered if the development of any methodological tool for systems problem solving is to be attempted. Starting at level one, the amount of knowledge in the systems increases as the hierarchical ladder is climbed. The different epistemological levels of systems are derived from two simple notions:

a) A region in the universe must be defined where the system and the observer coexist and interact. A system in this context can be interpreted as a set of relations between some objects that belong to that region of the universe and in which the observer is interested.

b) A set of variables to represent the system has to be chosen and classified, normally, into input and output variables, which is a natural classification of variables: input variables depend on the environment and control the output variables.

A simplified representation of the five epistemological levels can be seen in Figure 3.1[1]. They are:

i) Level one: Source System.

ii) Level two: Data System.

iii) Level three: Generative System.

iv) Level four: Structure System.

v) Level five: Meta System.

**Source Systems.** They are at the lowest epistemological level. Also known as Dataless Systems, they represent the system as it is recognized by the observer. The amount of information present at this level represents the basic description of the problem in which the observer is interested, and consequently this information must include the variables that are relevant to the problem, the input/output relationships among these variables, and the states these variables can assume along their time–history. The number of states, or levels, that each variable can potentially assume is essentially problem dependent. It should be kept as low as possible without unacceptable loss of information. The lower level systems are contained in those that are at higher epistemological levels, thus the Source system is included in all of the higher level systems. The reason(s) for the observer to have chosen the system as such is not arbitrary and can be interpreted as the link between this system and its environment [Vesanterä, 1988].

**Data Systems.** Data Systems are located at the next epistemological level in the hierarchy, that is, at level 2. They include the Source System and additionally, the observed or measured time histories of all its variables, i.e., a Source System supplied with data. Data may be obtained by simulation of another system model, or by observation in system analysis problems, or may be imposed as desired states in system design problems. Since the Source Systems include the states the variables can assume along their time histories, the data supplied to the Data System must be represented in terms of those states, i.e., data become qualitatively classified into the states.

**Generative Systems.** Located at level 3, Generative Systems (also known as Behavior Systems) include, in addition to the knowledge of Source and Data Systems, the time–invariant relationships existent among their variables.

[1] This figure has been taken from [Klir, 1985a].

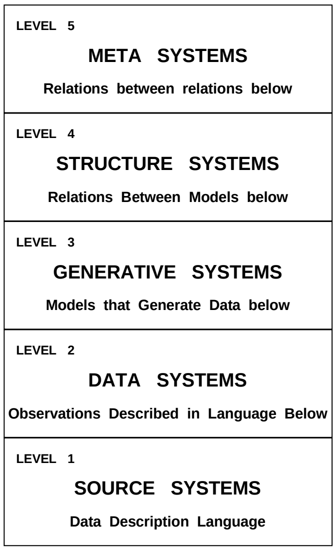


Figure 3.1: Klir's epistemological levels of systems.

These relationships are really *translation rules* that can be used to generate new states of the variables within the time span defined in the Data Model. The generation of new states for each variable is the feature that enables the inductive reasoning capabilities of the methodology as will be shown later in this chapter, when the qualitative modeling technique based on Optimal Mask Analysis is explained. On the same inductive reasoning capabilities of the generative systems is also based the design of a Fault Monitoring System built to detect misbehavior, transients, faults, and structural changes in a modeled system.

**Structure Systems.** Located at level 4, Structure Systems are defined in terms of a set of Generative Systems that can be viewed as a set of subsystems of the overall system. The knowledge included in this level, besides all the knowledge of the preceding levels, is composed of the causal and temporal relationships among the subsystems. In chapter 6, Reconstruction Analysis, a level 4 methodology used to obtain minimum sets of meaningful variables to be treated as subsystems, will be explained.

**Meta Systems.** Meta Systems constitute the fifth and final level of the epistemological hierarchy. They are composed of a set of Structure Systems that are required to share the same Source System, and a set of meta–rules describing relationships among these systems.

The Fuzzy Inductive Reasoning approach used in this dissertation is based on the aforementioned epistemological hierarchy of systems. Each of its modules has a correspondence with one of the epistemological levels, and basically, the methodology reaches level 3, i.e., the behavioral level. To be more precise, Fuzzy Inductive Reasoning involves levels 1, 2, and 3, and in some way also level 4, because one of its tools, namely the *Optimal Mask Analysis*, a tool for reasoning about the behavior of a system while deciding something about its structure that will be explained later in this chapter, bridges the gap between the epistemological levels 3 and 4, since it draws its inputs from level 3, but delivers outputs at level 4.

## 3.3 Fuzzy Inductive Reasoning Methodology

The Fuzzy Inductive Reasoning methodology has experienced lots of changes as it was stated at the beginning of this chapter. In this section, the up–to–date methodology is presented, i.e., the ultimate version of SAPS–II. At this moment, SAPS–II is a Fortran 77 coded system with interfaces to Matlab [MathWorks, 1992], CTRL–C [SCT, 1985], and ACSL [MGA, 1991].

Fuzzy Inductive Reasoning (FIR from now on) is a technique for constructing qualitative models that are represented by a special class of fuzzy Finite State Machines (FSMs), and that are used to qualitatively simulate the behavior of a given system. In inductive reasoning, *induction* stands for the fact that a set of rules can be derived from measurement data alone, rather than through any other type of knowledge or meta–knowledge; and *reasoning* stands for the fact that a set of rules, and as a consequence new knowledge, is obtained from previously available knowledge.

As already stated, the FIR methodology is entirely data–driven. Consequently, it does not require any knowledge of the internal structure of the system under investigation or any other knowledge except for time histories of input/output behavior. However, available *a priori* structural, expert, or experiential knowledge can be used to restrict the number of generated rules and the span of the search space. An inductive reasoning model, constructed on the basis of measurement data, qualitatively represents the input/output behavior of the modeled system in the vicinity of an operating point or an operating trajectory.

### 3.3.1 Definitions

Before starting to explain in detail how FIR works, it may be helpful to provide a definition of some terms that shall be frequently met in the context of the FIR methodology. The definition list is ordered following the natural order of the FIR methodology.

- **Fuzzy Recoding.** The process that converts a single quantitative value into a qualitative triple, also known as fuzzification.
- **Raw Data Matrix.** Is where the measurement data (observations) are stored. Each column of the raw data matrix represents one of the quantitative variables, and each row represents one time point.
- **Qualitative Data Model.** Is where the recoded triples corresponding to the measurement data are stored. The qualitative data model consists of three matrices of the same size as the raw data matrix, one holding the class values, the second containing the fuzzy membership values, and the third storing the side values.
- **Mask.** Is a matrix representation of a dynamic relationship among qualitative variables. Each column of the mask represents one qualitative variable, and each row represents one time point. The negative elements of the mask represent relative positions of inputs to the qualitative functional relationship (the so–called *m–inputs*), the positive element marks the position of the output of this relationship (the *m–output*), and zero elements denote forbidden connections among the qualitative variables.
- **Mask Candidate Matrix.** Is an ensemble of all possible masks from which the best is chosen by a mechanism of exhaustive search. Here, the negative elements denote positions of potential inputs to the mask. The meaning of the positive and zero elements is the same as in the case of the mask.
- **Complexity of a Mask.** Is the number of non–zero entries in a mask, i.e., the number of allowed relationships between qualitative variables in the mask.
- **Depth of a Mask.** Is the number of rows of the mask matrix. It determines the time span covered by the mask at any point in time.
- **Optimal Mask.** Is a matrix representation of the most plausible qualitative causal and temporal relationship among the recoded

variables, i.e., it constitutes the qualitative model. The optimal mask is used to obtain the set of rules called the *behavior matrix.*

- **Quality of a Mask.** Is a measure of the optimality of a mask, evaluated with respect to the maximization of its forecasting power. The Shannon entropy in combination with the complexity of the mask or the observation ratio, are the measures used to determine the uncertainty associated with forecasting a particular output state given any legal input state.

- **Mask History Matrix.** It consists of a horizontal concatenation of all best masks found for each level of allowed complexity. One of these masks is the optimal mask.

- **Optimal Mask Analysis.** The process that generates a qualitative model by analyzing the causal and temporal relationships present in the episodical behavior, i.e., the process that leads from the qualitative data model to the optimal mask.

- **Input/Output Matrix.** Is the static relationship formed by shifting the mask over the qualitative data model extracting those values that coincide with the $m$–inputs and $m$–output of the mask. The extracted values are written next to each other in one row. Each row is called a *state* of the system and consists of an *input state* and an *output state.*

- **Input State and Output State.** The input state denotes the vector of values of all the inputs belonging to the state, and the output state is the value of the only output of the state.

- **Behavior Matrix.** Is the input/output matrix with its relationships sorted in alphanumerical order. The behavior matrix is the ordered set of rules that, for any given input state, returns the output that is most likely to be observed.

- **Fuzzy Forecasting.** The process that carries on the qualitative simulation predicting the future behavior of the qualitative variables.

- **Regeneration.** The inverse process of the fuzzy recoding also known as defuzzification. It converts a qualitative triple into a single quantitative value.

## 3.3.2 Obtaining Data

The FIR methodology requires data (behavior trajectories in FIR terminology) for the inductive reasoner. To obtain this data, two important processes, not FIR processes but directly related with them, must be performed before any inferencing can be done. These processes have to do with the way in which information is produced, and with the way in which variables are selected. These processes concern (i) the excitation of the system, and (ii) the selection of variables.

### 3.3.2.1 Excitation of the System

In order for FIR to be able to learn the behavior of a certain system, this system must exhibit the highest possible number of behavioral patterns, i.e., all operating frequencies must be richly represented. The higher the number of behavioral patterns observed, the better shall be the characterization of the system, since FIR cannot predict what it has never observed. A cheap way of exciting a system is with *random binary noise*. In this way the complete dynamics of the system can be properly characterized; however, not all possible system states are captured since the random binary noise produces extreme dynamic conditions. To overcome this problem, an excitation strategy of mixing random binary noise inputs and random harmonic functions of long periods is normally used, depending on the application.

The system's excitation can turn out to be a very difficult task. If a model is available, as it usually happens in engineering, the excitation can be performed on the model without any danger to the real system. However, if the system cannot be modeled in an accurate way, which may happen because e.g. its internal mechanisms are not fully known, or if the system is a large–scale system too complex to be modeled in full, the excitation should be performed on the system itself. An additional problem occurs if the qualitative model is to be used for fault diagnosis. The quantitative model that is available may have been validated only for operation within the normal operating range of the real system, and may not reflect the behavior of the real plant when operated outside that region. Also, if the quantitative system model is intended for safety or training operations, it may not support modes of excitation that would drive the operating point of the model away from the region that the real plant is supposed to be operated in. In practice this means that, since a real system is not necessarily available for excitation in such a way that a proper characterization can be performed by FIR, the system may only be available for modeling and simulating in some but not all of its possible dynamic states.

#### 3.3.2.2 Variable Selection

The variable selection process contains two separate aspects. One has to do with the selection of the variables that are going to be used as outputs, i.e., the variables for which future values should be predicted. This selection must always be carried out by the modeler before the process of qualitative modeling can be attempted.

The other aspect has to do with the decision, which among the set of available variables should be considered as potential inputs. In a system of small or medium size, where the total number of variables is kept within reasonable limits, the FIR methodology can look at all variables deciding, which ones are strongly correlated with each other, and which others show only weak cross–correlations. A meaningful selection of input variables will include those variables that exhibit strong cross–correlations with the selected output variable, yet weak cross–correlations among each other.

In a large–scale system, the inductive reasoner cannot deal with the overwhelming number of variables inherent of such systems. Thus a preselection of a subset of prospectively good candidate variables must be performed. This task of selecting a minimum set of meaningful variables can be done in several ways. It can be done by the modeler if he or she possesses structural knowledge about the system. It can also be done by means of analyzing different subsets of variables in sequence using the *Optimal Mask Analysis* tool that will be explained later in this chapter [de Albornoz and Cellier, 1993b]. Finally, another way in which a minimum set of meaningful variables can be preselected is by means of *Reconstruction Analysis* [De Albornoz and Cellier, 1996], a methodological tool that will be presented in Chapter 5. The different methods will be compared to each other in Chapter 6.

### 3.3.3 Fuzzification

The FIR methodology rests basically on two methodological pillars: *Qualitative Modeling through Optimal Mask Analysis* and *Qualitative Simulation through Fuzzy Forecasting*. FIR also needs two converter modules, one that converts quantitative information into qualitative information, and another that performs the reverse operation. Thus the four main processes in

FIR are:

i) Fuzzification.

ii) Qualitative Modeling.

iii) Qualitative Simulation.

iv) Defuzzification.

All four of these functions have their foundations in General System Theory. The entire FIR methodology can be seen in Figure 3.2. The four tools are now described.

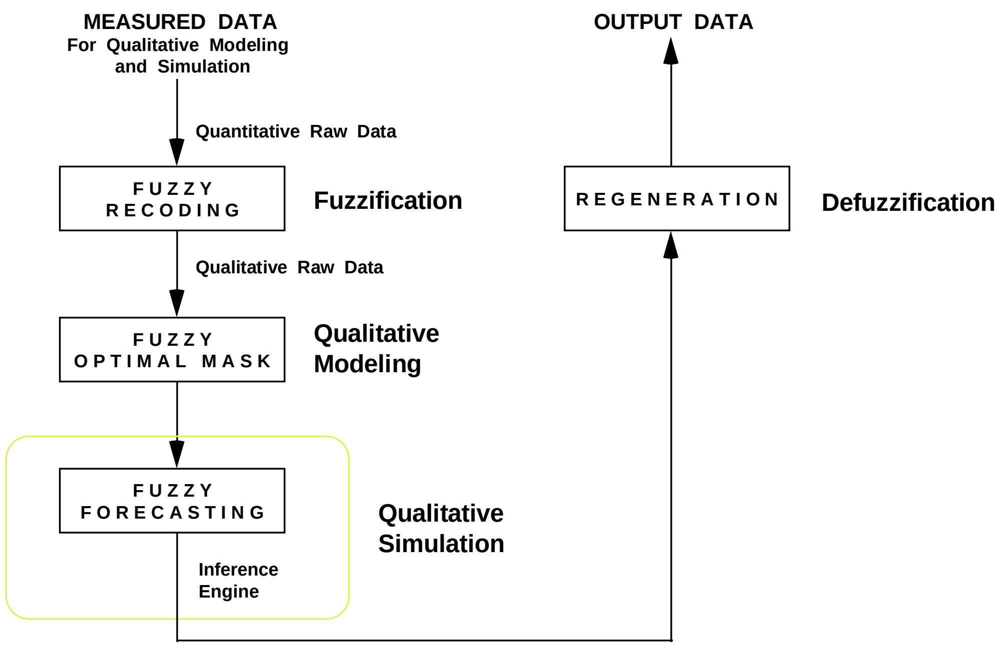


Figure 3.2: Fuzzy Inductive Reasoning process.

Inductive reasoners, like all other qualitative reasoners, base their reasoning on discrete (qualitative) variables. To this end, continuous input signals that constitute the measurement data must be converted to discrete values. In the language of fuzzy inductive reasoning, the discretization process is referred to as *Fuzzy Recoding*. In the fuzzy systems literature, this process is usually referred to as *fuzzification*.

Recoding denotes the process of converting a quantitative variable to a qualitative variable. In general, some information is lost in the process of recoding. Obviously, a temperature value of 23$^o$C contains more information than the value "normal." Fuzzy recoding avoids this problem by converting a single quantitative value into a qualitative triple that includes a discrete "class" value, a "fuzzy membership" value, and a "side" value[2].

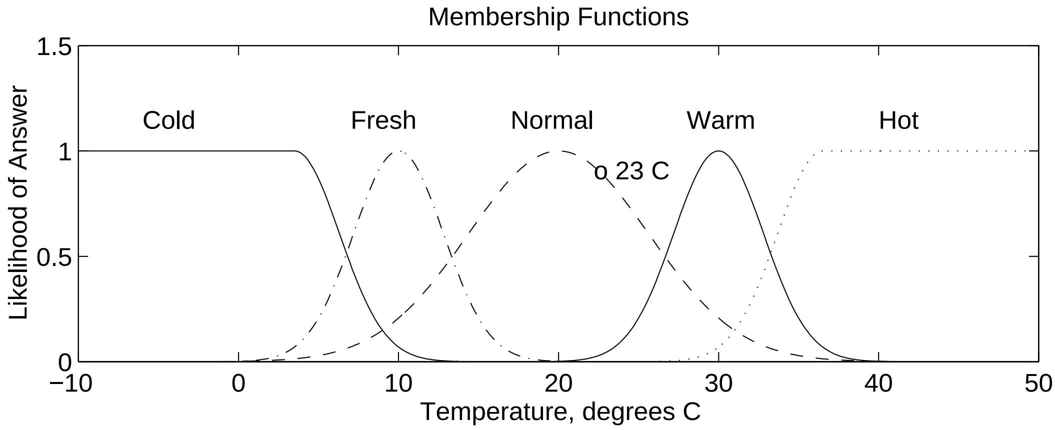


Figure 3.3: Typical membership functions used in the process of Fuzzy Recoding.

Figure 3.3 shows the fuzzy recoding of a variable called "environmental temperature." For example, a quantitative temperature of 23 degrees Centigrade is recoded into a qualitative value of 'normal' with a fuzzy membership function of 0.895 and a side function of 'right.' Thus, a single quantitative value is recoded into a qualitative triple. Any temperature with a quantitative value between 13$^o$C and 27$^o$C will be recoded into the qualitative value 'normal.' The fuzzy membership function denotes the value of the bell–shaped curve shown in Figure 3.3, always a value between 0.5 and 1.0. It was decided to use bell–shaped (normally distributed) fuzzy membership functions rather than the more commonly used triangular ones. These membership

[2] One of the current lines of research in FIR deals with the introduction of additional information about the qualitative derivative of the variables. Although fuzzy recoding does not lose any information up front, as shall be shown shortly, this does not guarantee that the total information available will be utilized in the process of fuzzy inferencing. It may make sense to introduce *redundancy* into the qualitative data model in order to reduce the risk of losing information later on in the inferencing process. To this end, *Causal Inductive Reasoning* (CIR), a new approach to inductive reasoning based on FIR, recodes quantitative values into qualitative *quadruples*. Each quadruple contains the same three pieces of information as used in FIR, plus a qualitative derivative value that indicates whether the recoded variable is currently increasing, decreasing, or staying at about the same level [Cellier and López, 1995]. However, this line of research was not pursued by the author of this thesis.

functions have a value of 1.0 at the arithmetic mean value $mu_i$ of any two neighboring *landmarks* (the borderlines that separate two qualitative classes), and a value of 0.5 at the landmarks themselves.

One problem remains to be discussed: How normal is normal? Obviously, qualitative terms are somewhat subjective, which makes the concept of landmarks a treacherous one. Is it really true that an environmental temperature of 26.9°C is "normal," whereas an environmental temperature of 27.1°C is "warm"? Fuzzy measures were introduced to inductive reasoning as a technique to deal with the uncertainty of landmarks. Instead of saying that the environmental temperature is "normal" for values below 27°C and "warm" for values above 27°C, a fuzzy measure allows us to specify that, as we pass the value 27°C in the positive direction, the answer "normal" becomes less and less likely, whereas the answer "warm" becomes more and more likely [Li and Cellier, 1990].

The membership functions can be easily calculated using the equation:

$$Memb_i = \exp\left(-\tau_i \cdot (x - \mu_i)^2\right) \tag{3.1}$$

where:

| | | |
|---|---|---|
| $x$ | = | continuous variable to be recoded; |
| $\mu_i$ | = | algebraic mean between two neighboring landmarks; |
| $\tau_i$ | = | is determined such that the membership function $Memb_i$ degrades to a value of 0.5 at the landmarks. |

The first and last membership functions are treated a little differently. Their shape ($\tau_i$) value is the same as for their immediate neighbors, and they are semi–open. Contrary to other fuzzy approaches, the tails of the membership functions ($Memb_i < 0.5$) are ignored in the method described in this thesis. The decision to ignore the tails of the membership functions is related to the selection of the fuzzy inferencing technique, and is justified in [Mugica and Cellier, 1993].

The side function indicates whether the quantitative value is to the left or to the right of the maximum of the fuzzy membership function. In this way, the class and side values can be used to process the available information in a qualitative fashion, whereas the fuzzy membership functions allow to preserve more quantitative information in the reasoning process. The qualitative analysis allows to generate quickly a rough qualitative response, whereas the fuzzy membership functions can then be used to smoothly interpolate between neighboring qualitative class values. In this way, an inductive reasoning model is able to preserve some numerical information about the system under study, information that enables the modeler to undo the process of fuzzy recoding, thereby making a prediction in quantitative terms. By adding the side value information to the qualitative triple, the complete quantitative information is preserved in a concise and easily reversible format without need for multiple class assignments, as long as no fuzzy membership functions with a flat top (like trapezoidal membership functions) are being used. Thus, the qualitative triple contains the same information as the original quantitative variable. Therefore, our approach to fuzzification can indeed be thought of as a process of recodification.

The recoded data can be thought of as the Data Model of the epistemological hierarchy of systems previously defined, and can be represented through three $n_{rec} \times n_{var}$ matrices, where $n_{rec}$ is the number of data recordings collected in the covered time span, and $n_{var}$ is the number of variables present in the model. This is a matrix representation of the *episodical behavior* (time-history) of the system, where the input variables are normally located in the leftmost columns, and time increases from top to bottom of the three matrices. In FIR terminology, the three matrices together are called the *Qualitative Data Model.*

#### 3.3.3.1 Qualitative Levels and Landmarks

At this point, the following questions can be raised. How many discrete levels should be selected for each state variable, and where should the borderlines (named *landmarks* in FIR terminology) be drawn that separate neighboring regions from each other?

From statistical considerations, it is known that, in any class analysis, each legal discrete state should be recorded at least five times [Law and Kelton, 1990]. Thus, a relation exists between the total number of legal states and the number of data points required to base the modeling effort upon:

$$n_{\text{rec}} \geq 5 \cdot n_{\text{leg}} = 5 \cdot \prod_{\forall i} k_i \tag{3.2}$$

where:

| | | |
|---|---|---|
| $n_{\mathrm{rec}}$ | = | total number of recordings, i.e., total number of observed states; |
| $n_{\mathrm{leg}}$ | = | total number of distinct legal states; |
| $i$ | = | index that loops over all variables in the state; |
| $k_i$ | = | number of levels that the $i^{\mathrm{th}}$ variable can assume. |

If the number of variables is given (as it usually happens in small– to medium–sized systems), the number of recordings is predetermined, and it is assumed that all variables are recoded into the same number of levels, then the optimum number of levels, $n_{\mathrm{lev}}$, of all variables can be found from the following equation:

$$n_{\mathrm{lev}} = \mathrm{round}\ (\ \sqrt[n_{\mathrm{var}}]{n_{\mathrm{rec}}/5}\ ) \tag{3.3}$$

For reasons of symmetry, an odd number of levels is often preferred over an even number of levels. Abnormal states ("too low," "too high," and "much too low," "much too high") are grouped symmetrically about the "normal" state.

The number of levels chosen for each variable is very important for several reasons. This number influences directly the computational complexity of the inference stage. Traditional fuzzy systems usually require between seven and 13 classes for each variable [Aliev *et al.*, 1992; Maiers and Sherif, 1985]. An exhaustive search in such a high–dimensional discrete search space would be very expensive, and the number of classes should therefore be reduced, if possible, to help speed up the optimization. It was shown in [Mugica and Cellier, 1993] that the selected fuzzy inferencing technique makes it possible to reduce the number of levels down to usually three or five, a number confirmed by several practical applications [Vesanterä, 1988; de Albornoz and Cellier, 1993a; de Albornoz and Cellier, 1993b; Nebot, 1994; Mugica, 1995].

Once the number of levels of each variable has been selected, the landmarks must be chosen to separate neighboring regions from each other. There are several ways to find a meaningful set of landmarks. Most frequently used is an approach in which each class is assigned the same number of recoded points. This method is based on the idea that the expressiveness (or information content) of the model will be maximized if each level is observed equally often. In order to distribute the observed trajectory values of each variable equally among the various levels, they are sorted into ascending order, the sorted vector is then split into $n_{\mathrm{lev}}$ segments of equal length, and the landmarks are chosen anywhere between the extreme values of neighboring segments, e.g., as the arithmetic mean values of neighboring observed data points in different segments.

In some cases, an equal distribution of data points among levels may not be suitable, however. For example, the designation of environmental temperature values as "cold," "moderate," or "hot," for example, does not depend on available measurement data, but on common sense. Thus, sometimes it may be more appropriate to ask one or several "experts" where the borders should be drawn between neighboring classes, rather than determine these values in an automated fashion.

Also, the assumption that each variable should be recoded into the same number of classes does not always hold. For example, it frequently happens that one or several of the measurement variables are binary signals. It evidently makes no sense to recode a binary signal into three levels.

The SAPS–II function to perform fuzzy recoding on the raw data model using the Matlab interface is called *recode*:

$$[c, m, s] = \text{recode } (meas(:, i), 'fuzzy', from, to)$$

where the input variables stand for:

| | | |
|---|---|---|
| $meas(:, i)$ | = | is the raw data matrix including all rows and one column corresponding to all possible states of one of the variables to be recoded; |
| $'fuzzy'$ | = | tells SAPS–II that the fuzzy option has been selected; |
| $from$ | = | indicates the previously computed landmarks; |
| $to$ | = | indicates the corresponding names for the qualitative levels ('1,' '2,' '3,' etc). |

The output variables stand for:

| | | |
|---|---|---|
| $c$ | = | is the vector of recoded class values of state $i$; |
| $m$ | = | is the vector of recoded membership function values of state $i$; |
| $s$ | = | is the vector of recoded side values of state $i$. |

### 3.3.4 Qualitative Modeling

By now, the quantitative trajectory behavior (the *raw data model*) has been recoded into a qualitative episodical behavior, and has been stored in a *qualitative data model* (consisting of three separate matrices as explained in the preceding section). Each column of the qualitative data model represents one of the observed variables, and each row represents one time point, i.e., one recording of all variables, or one recorded state. The class values are in the sets of legal levels that the variables can assume. In the current implementation of FIR, they are all positive integers, usually ranging from '1' to '3,' '1' to '5,' or '1' to '7,' since SAPS–II employs integers instead of symbolic values to represent qualitative classes (levels). In an inductive reasoning system coded in LISP or Prolog, symbolic names would probably be preferred, whereas in an inductive reasoning system coded in a predominantly numeric software such as Matlab or CTRL–C, integers will be the representation of choice. From a practical point of view, it really does not matter, which of the two representations is being used, since one can easily be mapped into the other.

#### 3.3.4.1 Masks as Qualitative Models

How does the episodical behavior support the identification of a model of a given system for the purpose of forecasting its future behavior for any given input stream?

The input/output (trajectory) behavior of the system to be modeled has been recorded, recoded into a qualitative (episodical) behavior, and is now available for modeling. It is assumed that all inputs to the real system and a set of measurable outputs have been recorded. The trajectory behavior can thus be separated into a set of input trajectories, $u_i$, concatenated from the right with a set of output trajectories, $y_i$, as shown in the following example containing two inputs $u_1$ and $u_2$, and three outputs $y_1$, $y_2$, and $y_3$:

$$\begin{array}{ll} \textit{time} & \quad u_1 \quad\;\; u_2 \quad\;\; y_1 \quad\;\; y_2 \quad\;\; y_3 \\ \begin{array}{l} 0.0 \\ \delta t \\ 2\cdot\delta t \\ 3\cdot\delta t \\ \vdots \\ (n_{rec}-1)\cdot\delta t \end{array} & \begin{pmatrix} \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{pmatrix} \end{array} \tag{3.4}$$

In order to avoid possible ambiguities, it is defined that the terms "inputs" and "outputs," when used in this dissertation without further qualifier, shall always refer to the input and output variables of the subsystem to be modeled by the qualitative reasoner.

In the process of modeling, it is desired to discover finite automata relations among the recoded variables that make the resulting state transition matrices as deterministic as possible. If such a relationship is found for every output variable, the behavior of the system can be forecast by iterating through the state transition matrices. The more deterministic the state transition matrices are, the better is the certainty that the future behavior will be predicted correctly.

A possible relation among the qualitative variables for this example could be of the form:

$$y_1(t) = \tilde{\mathrm{f}}(y_3(t-2\delta t), u_2(t-\delta t), y_1(t-\delta t), u_1(t)) \tag{3.5}$$

Equation (3.5) can be represented as follows:

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} u_1 & u_2 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & -2 & -3 & 0 & 0 \\ -4 & 0 & +1 & 0 & 0 \end{pmatrix} \end{array} \tag{3.6}$$

The negative elements in this matrix are referred to as $m$–inputs. $M$–inputs denote inputs of the qualitative functional relationship. They can be either inputs or outputs of the subsystem to be modeled, and they can represent different time instants. The above example contains four $m$–inputs. The sequence in which they are enumerated is immaterial. They are usually enumerated from left to right and top to bottom. The positive value denotes the $m$–output. In the above example, the first $m$–input, $i_1$, corresponds to the output variable $y_3$ two sampling intervals back: $y_3(t-2\delta t)$, the second $m$–input refers to the input variable $u_2$ one sampling interval in the past: $u_2(t-\delta t)$, the third $m$–input corresponds to the output variable $y_1$ one sampling interval in the past: $y_1(t-\delta t)$, the fourth $m$–input refers to the input variable $u_1$ at the current sampling interval: $u_1(t)$, and the only $m$–output corresponds to the output variable $y_1$ at the current sampling interval: $y_1(t)$. This is exactly what Equation (3.6) represents.

In inductive reasoning, such a representation is called a *mask*. A mask is a matrix representation of a translation rule relative to a given Data Model, i.e., it is a matrix representation of the Generative or Behavior Model of the system. In FIR terminology, a mask is the qualitative model of the given system and denotes the dynamic relationship among its qualitative variables. A mask has the same number of columns as the episodical behavior to which it should be applied, that is the number of input and output variables $n_{var}$, and it has a certain number of rows. The number of rows of the mask matrix is called the *depth* of the mask. It is related to the number of sampling intervals that the mask covers. The mask represented by Equation (3.6) has a depth of three.

The mask can be used to flatten a dynamic relationship out into a static relationship. It can be shifted over the qualitative data model, the selected $m$–inputs and $m$–output can be extracted from the qualitative data, and they can be written next to each other in one row of the so–called *input/output matrix*. Figure 3.4 illustrates this process. After the mask has been applied to the qualitative data, the formerly dynamic episodical behavior has become static, i.e., the relationships are now contained within single rows.

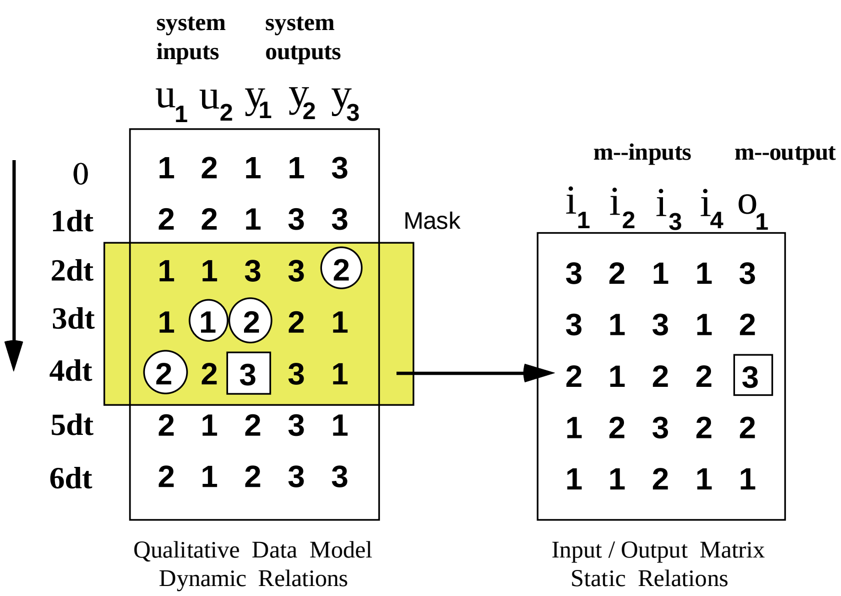


Figure 3.4: Flattening dynamic relationships through masking.

Each row of the input/output matrix is called a *state* of the system. The state consists of an *input state* and an *output state*. The input state denotes the vector of values of all the $m$–inputs belonging to the state, and the output state is the value of the single $m$–output of the state. The set of all possible states is referred to as the set of *legal states* of the qualitative model. Thus, a model with $n_{var}$ number of variables, each of which being recoded into $k$ levels, has $k^{n_{var}}$ legal states.

Note that, for reasons of causality, the mask's only $m$–output must always be located in the bottom row of the mask, corresponding to the current sampling interval $t$, which means that, as long as the $m$–input variables are available, the $m$–output variable can be obtained, which gives the mask generative capabilities. Taking the example of Equations (3.5) and (3.6), the state of the $m$–output $y_1$ at the next sampling interval: $y_1(t + \delta t)$, represented by the relationship:

$$y_1(t + \delta t) \;=\; \tilde{\mathrm{f}}(y_3(t - \delta t), u_2(t), y_1(t), u_1(t + \delta t)) \tag{3.7}$$

which can be interpreted as the mask:

$$\begin{array}{ll} {}_{t}\backslash^{x} & \begin{array}{ccccc} u_1 & u_2 & y_1 & y_2 & y_3 \end{array} \\ \begin{array}{l} t - \delta t \\ t \\ t + \delta t \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & -2 & -3 & 0 & 0 \\ -4 & 0 & +1 & 0 & 0 \end{pmatrix} \end{array} \tag{3.8}$$

can be generated or *forecast* if the $m$–input variables, one sampling interval into the future, are known. In this example, $y_1$ is not the only output variable of the system, $y_2$ and $y_3$ are also outputs and must be obtained as well; however, FIR does not permit causal dependencies between systems outputs to avoid algebraic loops. For that reason, $y_2(t)$ and $y_3(t)$ in Mask (3.6), and $y_2(t + \delta t)$ and $y_3(t + \delta t)$ in Mask (3.8) are zero, which means forbidden connections, or forbidden dependencies. The variables $y_2$ and $y_3$ should be computed using a different mask for each of them. The mask generating $y_2$ should include zeros in the bottom row for the variables $y_1$ and $y_3$, and the one computing $y_3$ should include zeros in the same row for $y_1$ and $y_2$.

Until now, we only used the class values in the computation of the input/output matrix. However, also the membership values are useful. The fuzzy membership value associated with the class value of a variable is a measure of the *confidence* that we have in the correctness of its assignment. An entire observation (input/output state) contains one or several input variables and one output variable. Each of them has a class value and a fuzzy membership value associated with it, and somehow, we need to declare what confidence value we assign to the observation as a whole. The confidence of an observation $Conf_{obs}$ is defined as the joint membership of all the inputs and the single output associated with that observation [Li and Cellier, 1990].

In FIR, the joint membership of $i$ membership functions is defined as the smallest individual membership[3]:

$$Memb_{\text{joint}} = \bigcap_{\forall i} Memb_i = \inf_{\forall i}(Memb_i) \stackrel{\text{def}}{=} \min_{\forall i}(Memb_i) \tag{3.9}$$

Thus the confidence of a particular observation is defined as:

$$Conf_{\text{obs}} \stackrel{\text{def}}{=} \min_{\forall i}(Memb_i) \tag{3.10}$$

In FIR, the function:

$$[IOmat, Conf] = \text{fiomodel }(class, Memb, mask)$$

is used to compute the input/output matrix together with its associated vector of observation confidence values from the class and membership matrices of the qualitative data model and a given mask.

The same input/output state (or observation) may have been observed several times in the past. As the same input/output state is observed again and again, the confidence in the correctness of that state grows. Thus, FIR defines the confidence of the $i^{\text{th}}$ input/output state, $Conf_{\text{IO\_state}}(i)$, as the sum of the confidences of its observations[4], i.e., all input/output states that coincide in their class values:

$$Conf_{\text{IO\_state}}(i) \stackrel{\text{def}}{=} \sum Conf_{\text{obs}}(i) \tag{3.11}$$

The FIR function:

$$[Beh, Conf_{IO}] = \text{fbehavior }(IOmat, Conf)$$

[3]Some fuzzy approaches define the joint membership value as the product of the membership values of its members.

[4]Some fuzzy approaches define the accumulated confidence value of a state as the largest of the individual confidence values of its observations.

is used to compute an ordered list of all input/output states (the *behavior* of the input/output matrix) together with the vector of accumulated confidence values in each of the observed input/output states.

#### 3.3.4.2 Sampling Interval

It has been discussed in the Fuzzification section, how many measurements (data points in time) are needed to characterize a system. However, it has not been discussed yet how long the time intervals between those measurements should be, i.e., how the time distance between two logged entries of the trajectory behavior (sampling interval), $\delta t$, is chosen in practice. $\delta t$ must be selected carefully, because its value will strongly influence the mask selection process. Determining a good value for this parameter in a systematic way is currently an object of research, and is one of the open problems that remain to be tackled, and that will be stated in Chapter 8. However, experience has shown that the mask should cover the largest time constant (the slowest mode) to be captured by the model.

If the trajectory behavior stems from measurement data, a Bode diagram of the system to be modeled can be obtained. This enables to determine the eigenfrequencies of the system, and in particular, the smallest and largest eigenfrequencies. The smallest eigenfrequency $\omega_{low}$ is the smallest frequency, at which the tangential behavior of the amplitude of the Bode diagram changes by $-20$ dB/decade, and the largest eigenvalue $\omega_{high}$ is the highest frequency where this happens. The largest time constant, $t_{settling}$, and the shortest time constant, $t_{fast}$, of the system can then be computed as follows:

$$t_{settling} = \frac{2\pi}{\omega_{low}} \quad ; \quad t_{fast} = \frac{2\pi}{\omega_{high}} \tag{3.12}$$

In order to satisfy the sampling theorem, the sampling rate should be chosen as:

$$\delta t = \frac{t_{fast}}{2} \tag{3.13}$$

The mask will cover the slowest time constant $t_{settling}$, if the mask depth is chosen as an integer approximation of the ratio between the largest and smallest time constants to be captured by the model plus one:

$$depth = \text{int}\left(\, 2\, \frac{t_{settling}}{t_{fast}}\right) + 1 \tag{3.14}$$

However, this ratio should not be much larger than three or four. Otherwise, the inductive reasoner won't work very well, since the computing effort grows factorially with the size of the mask. Multiple frequency resolution in inductive reasoning is still an area of open research; however, some results relating to this problem have already been obtained, and shall be presented in due course.

#### 3.3.4.3 Optimal Mask Analysis

How is the mask found that, within the framework of all allowable masks, represents the most deterministic state transition matrix, and consequently, ensures optimal predictiveness of the model?

The inductive reasoner operates exclusively on the aforementioned qualitative triples and reasons about qualitative spatial and temporal relations. The most plausible qualitative causal relationship among the recoded variables is called the *fuzzy optimal mask*. This finite state machine represents a temporal qualitative model, and is used to obtain a set of rules called the *behavior matrix* of the system. The optimal mask is obtained by performing an exhaustive comparison between all possible masks with respect to the maximization of their forecasting power. The *Shannon entropy measure* is used to determine the uncertainty associated with forecasting a particular output state, given any input state. An *observation ratio* is introduced as a measure of mask complexity, disfavoring the selection of unnecessarily (unjustifiably) complex models. Let us explain one by one all these concepts.

**The Mask Candidate Matrix**

In FIR, the concept of a *mask candidate matrix* has been introduced. A mask candidate matrix is an ensemble of all possible masks from which the best is chosen by a mechanism of exhaustive search. The mask candidate matrix contains $(-1)$ elements where the mask has potential $m$–inputs, a $(+1)$ element where the mask has its $m$–output, and $(0)$ elements to denote forbidden connections. Thus, a good mask candidate matrix to predict $y_1(t)$ in the previously introduced five–variable example might be:

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} u_1 & u_2 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & +1 & 0 & 0 \end{pmatrix} \end{array} \tag{3.15}$$

Similar mask candidate matrices can be written for the other two system outputs, $y_2$ and $y_3$.

Notice that each mask candidate matrix contains only a single $m$–output, and that usually all connections are permitted except for those that include the other outputs of the system at the current time $t$. By not permitting a direct interaction among the outputs at the time of prediction, the forecasting algorithm is prevented from entering into algebraic loops, since otherwise it could e.g. happen that:

$$\begin{aligned} y_3(t) &= f_3\,(y_1(t)) \\ y_1(t) &= f_1\,(y_3(t)) \end{aligned} \tag{3.16}$$

The potential inputs in a mask candidate matrix do not need to be numbered since their function is solely to point out, which are the $m$–input variables that may possibly affect the behavior of the $m$–output variable of the mask[5].

The mask candidate matrix allows to encode meta–knowledge into the qualitative model. If some knowledge of the system structure is available *a priori*, it can be represented in the mask candidate matrix through enforced or forbidden connections between variables.[6] For example, in the mask candidate matrix represented in Mask (3.15), if we already know that the $m$–output $y_1(t)$ may not depend on the $m$–inputs $u_1(t-2\delta t)$, $u_2(t-\delta t)$, $u_2(t)$, and $y_3(t-\delta t)$, the following zero elements should be introduced into the mask candidate matrix:

[5]It would make sense to introduce an additional value, e.g. $(-2)$, to denote *required inputs*. In this case, only masks would be evaluated that contain the variables identified by this marker in the mask candidate matrix, reducing the overall search effort. However, this feature has not yet been implemented.

[6]While the forbidden connections feature is already available, the enforced connections feature is not.

$$
\begin{array}{l c}
{}_t\backslash^x & \begin{array}{ccccc} u_1 & u_2 & y_1 & y_2 & y_3 \end{array} \\
\begin{array}{l} t-2\delta t \\ t-\delta t \\ t \end{array} & \begin{pmatrix} 0 & -1 & -1 & -1 & -1 \\ -1 & 0 & -1 & -1 & 0 \\ -1 & 0 & +1 & 0 & 0 \end{pmatrix}
\end{array}
\tag{3.17}
$$

Evidently, any 0 (or $-2$) value introduced in the mask candidate matrix shall reduce the search effort, because it reduces the number of masks that are compatible with the mask candidate matrix.

In FIR, the FOPTMASK routine can be used to determine the optimal mask from a qualitative data model, a mask candidate matrix, and a parameter that limits the maximum tolerated mask complexity, i.e., the largest number of non–zero elements that the mask may contain. FOPTMASK searches through all legal masks of complexity two, i.e., all masks with a single permitted $m$–input and finds the best one; it then proceeds by searching through all legal masks of complexity three, i.e., all masks with two $m$–inputs and finds the best of those; and it continues in the same manner until the maximum complexity has been reached. In all practical examples, the quality of the masks will first grow with increasing complexity, then reach a maximum, and then decay rapidly. For most applications, a maximum complexity value of somewhere between four and seven is adequate.

### The Mask Quality

Each of the possible masks is compared to the others with respect to its potential merit. The optimality of the mask is evaluated with respect to the maximization of the forecasting power of the input/output matrix.

The Shannon entropy measure is used to determine the uncertainty associated with forecasting a particular output state given any legal input state. The Shannon entropy relative to one input state is calculated from the equation:

$$H_i = \sum_{\forall o} \mathrm{p}(o|i) \cdot \log_2 \mathrm{p}(o|i) \tag{3.18}$$

where $\mathrm{p}(o|i)$ is the conditional probability of a certain output state $o$ to occur, given that the input state $i$ has already occurred. Of course, these probabilities are not really known. They must be estimated. In FIR, probabilities are approximated through measures of *relative confidence*. Given a particular input state, several different output states may have been observed one or several times in the past associated with this input state. Each of these input/output states has an accumulated confidence value associated with it as shown in Equation 3.11. Thus, we can write:

$$\mathrm{p}(o|i) \approx \frac{Conf_{IO}(i,o)}{\sum_{(o)} Conf_I(i,o)} \tag{3.19}$$

i.e., as the confidence of the given input/output state divided by the sum of the confidence values of all input/output states that share the same input state. This quotient is evidently a real–valued number in the range $[0.0, 1.0]$, and can be interpreted as a probability.

The overall entropy of the mask is then calculated as the sum:

$$H_m = -\sum_{\forall i} \mathrm{p}(i) \cdot H_i \tag{3.20}$$

where $\mathrm{p}(i)$ is the probability of that input state to occur. Again, the exact probability is not known, and hence needs to be estimated. This time, the estimation is done differently. The probability of a given input state is approximated through the *relative observation frequency* of that input state:

$$\mathrm{p}(i) \approx \frac{\#\mathrm{obs}(i)}{\#\mathrm{all\_obs}} \tag{3.21}$$

i.e., as the number of previous observations of that particular input state divided by the total number of all previous observations.

The highest possible entropy $H_{\max}$ is obtained when all probabilities are equal, and a zero entropy is encountered for relationships that are totally deterministic.

A normalized overall entropy reduction $H_r$ is defined as:

$$H_r = 1.0 - \frac{H_m}{H_{\max}} \tag{3.22}$$

$H_r$ is obviously a real number in the range $[0.0, 1.0]$, where higher values usually indicate improved forecasting power. The optimal mask among a set of mask candidates is defined as the one with the highest entropy reduction.

Application of the Shannon entropy to a confidence measure is a somewhat questionable undertaking on theoretical grounds, since the Shannon entropy was derived in the context of probabilistic measures only. For this reason, some scientists prefer to replace the Shannon entropy by other types of performance indices [Klir, 1989; Shafer, 1976], which have been derived in the context of the particular measure chosen. However, from a practical point of view, numerous simulation experiments have shown that the Shannon entropy works satisfactorily also in the context of fuzzy measures if those confidences have been reinterpreted as conditional probabilities, as it has been done in our case as shown in Equation 3.19.

One problem still remains. The size of the input/output matrix increases as the complexity of the mask grows, and consequently, the number of legal states of the model grows quickly. Since the total number of observed states remains constant, the frequency of observation of each state shrinks rapidly, and so does the predictiveness of the model. The entropy reduction measure does not account for this problem. With increasing complexity, $H_r$ simply keeps growing. Very soon, a situation is encountered where every state that has ever been observed has been observed precisely once. Thus all observed state transitions are totally deterministic, and $H_r$ assumes a value of 1.0, which means that the forecasting power over a single step is maximized, however, the predictiveness of the model over several steps will nevertheless be poor, since already the next predicted state will, in all likelihood, never have been observed before, and that means the end of forecasting. Therefore, it seems not practical to use the Shannon entropy exclusively in the performance index that evaluates the quality of a mask.

Two different approaches to overcome this problem have been implemented in FIR. On the one hand, using the complexity of the mask in the performance index, and on the other hand, using the *observation ratio* in the performance index. Which of them is selected, is left up to the modeler and is carried out by changing a single global parameter in the Optimal Mask Analysis module. The two approaches are now explained.

**Complexity of the Mask Approach** [Uyttenhove, 1978]. To find the mask that best represents the relationship existing among the qualitative variables under consideration, i.e., the optimal mask, FIR goes through an exhaustive search that is carried out in levels. It first computes the qualities of the masks of lowest complexity (the ones with the smallest number of non–zero elements), chooses the best at that level and steps to the next complexity level. In this same fashion, all possible masks at all levels, up to a maximum complexity level specified by the modeler, will have their quality factors evaluated. In order to compare masks at different levels, a mask complexity weight $C_m$ is defined as:

$$C_m = \frac{n_{\text{var}} \cdot depth \cdot compl}{depth_{\max}} \tag{3.23}$$

where:

| | | |
|---|---|---|
| $n_{\text{var}}$ | = | number of variables; |
| $depth$ | = | current depth of the mask; |
| $compl$ | = | complexity of the mask (the number of non–zero entries in the mask); |
| $depth_{\max}$ | = | the maximum possible depth the mask could have (the depth of the chosen mask candidate). |

Finally, the quality measure $Q_m$ is defined in terms of the normalized overall entropy reduction $H_r$ and the mask complexity weight $C_m$ as:

$$Q_m = \frac{H_r}{C_m} \tag{3.24}$$

The mask with the highest quality factor found in the search is the *optimal mask.*

**Observation Ratio Approach** [Li and Cellier, 1990]. It was mentioned earlier that, from a statistical point of view, every state should be observed at least five times [Law and Kelton, 1990]. Therefore, an *observation ratio*, $O_r$, is introduced as an additional contributor to the overall quality measure, which reduces the mask quality if there exist states that have been observed less often than five times. This observation ratio is defined as:

$$O_r = \frac{5 \cdot n_{5\times} + 4 \cdot n_{4\times} + 3 \cdot n_{3\times} + 2 \cdot n_{2\times} + n_{1\times}}{5 \cdot n_{\text{leg}}} \tag{3.25}$$

where:

$n_{\rm leg}$ = number of legal input states;
$n_{1\times}$ = number of input states observed only once;
$n_{2\times}$ = number of input states observed twice;
$n_{3\times}$ = number of input states observed thrice;
$n_{4\times}$ = number of input states observed four times;
$n_{5\times}$ = number of input states observed five times or more.

If every legal input state has been observed at least five times, $O_r$ is equal to 1.0. If no input state has been observed at all (no data are available), $O_r$ is equal to 0.0. Thus, $O_r$ can also be used as a quality measure.

The overall *quality of a mask*, $Q_m$, is then defined as the product of its uncertainty reduction measure, $H_r$, and its observation ratio, $O_r$:

$$Q_m = H_r \cdot O_r \tag{3.26}$$

The *optimal mask* is then the mask with the largest $Q_m$ value.

In FIR, the function used to perform an Optimal Mask Analysis is known as *foptmask*. This function returns the overall best mask found in the optimization, the Shannon entropies of the best masks for every considered complexity, the corresponding uncertainty reduction measures, and the quality measures of these suboptimal masks. Finally, *foptmask* also returns the *mask history matrix*, a matrix that consists of a horizontal concatenation of the best masks at each complexity level. One of these masks is the optimal mask, which, for reasons of convenience, is also returned separately. The *foptmask* function has the following syntax:

$$[mask, H_m, H_r, Q_m, m_{\rm his}] = \text{foptmask}\ (class, Memb, mask_{\rm cand}, cplx)$$

where the input variables are:

$class$ = is the matrix of class values;
$Memb$ = is the matrix of fuzzy membership values;
$mask_{\rm cand}$ = is a mask candidate matrix;
$cplx$ = is a number that indicates the maximum tolerated complexity.

and the output variables are:

| | | |
|---|---|---|
| $mask$ | = | is the optimal mask; |
| $H_m$ | = | is a row vector that contains the Shannon entropies of the best masks for every considered complexity; |
| $H_r$ | = | is a row vector containing the corresponding uncertainty reduction measures; |
| $Q_m$ | = | is a row vector listing the quality measures of the suboptimal masks; |
| $m_{\text{his}}$ | = | is the mask history matrix. |

### 3.3.5 Qualitative Simulation

This process is used to infer class, side, and membership values of the output variables from class, side, and membership values of the input variables through a generated set of rules. This inference can be performed efficiently, since the search for optimal inference rules is limited to a discrete search space. The set of rules is obtained by applying the optimal mask to the qualitative data model resulting in the input/output matrix, that once ordered, becomes the *behavior matrix* of the system, which tells us for each input state, which output is most likely to be observed. The process of qualitative simulation through fuzzy inductive reasoning is called *Fuzzy Forecasting* and can be viewed as a temporal inference engine.

#### 3.3.5.1 Fuzzy Forecasting

Once the optimal mask has been determined, it can be applied to the given qualitative data model resulting in an input/output matrix. Since the input/output matrix contains functional relationships within single rows, the rows of the input/output matrix can be sorted in alphanumerical order. The result of this operation is called the *behavior matrix* of the system. The behavior matrix is a finite state machine. For each input state, it shows which output is most likely to be observed.

Forecasting has now become a straightforward procedure for the class and side values. The optimal mask is simply shifted further down over the qualitative input data records as shown in Figure 3.4, the values of the $m-$ inputs are read out from the mask, and the behavior matrix is used to look for the most likely match among the previously recorded observations of input/output behavior, determining the future class and side values of the $m$–output, which can then be copied back into the qualitative data model.

This process works in the following way. The membership and side functions of the new input state are compared with those of all previous recordings of the same input state contained in the behavior matrix. The one input state with the most similar membership and side functions is identified. For this purpose, a normalized quantitative value:

$$p_i = class_i + side_i * (1 - Memb_i) \tag{3.27}$$

is computed for every element of the new input state, and the $p_i$ values are stored in a vector:

$$\mathbf{p} = [p_1, p_2, \ldots, p_k] \tag{3.28}$$

where $k$ denotes the number of input variables in the input state. Normalized quantitative values are then also computed for all previous recordings of the same input state:

$$p_{ij} = class_{ij} + side_{ij} * (1 - Memb_{ij}) \tag{3.29}$$

where the index $i$ denotes the $i^{\text{th}}$ input variable, and the index $j$ denotes the $j^{\text{th}}$ previous recording of the same input state. Also the $p_{ij}$ values are concatenated into a vector:

$$\mathbf{p_j} = [p_{1j}, p_{2j}, \ldots, p_{kj}] \tag{3.30}$$

Finally, the $\mathcal{L}_2$ norms of the difference between the $\mathbf{p}$ vector of the new input state and the $\mathbf{p_j}$ vectors of all previous recordings of the same input state are computed:

$$d_j = \|\mathbf{p} - \mathbf{p_j}\|_2 \tag{3.31}$$

and the previous recording with the smallest $\mathcal{L}_2$ norm is identified. Its *class* and *side* values are then used as forecasts for the *class* and *side* values of the current state.

In fuzzy forecasting, it is essential that, together with the qualitative class and side values, also a fuzzy membership value is forecast. Thus, fuzzy forecasting predicts an entire qualitative triple from which a quantitative variable can be regenerated whenever needed.

Forecasting of the new membership function is done a little differently. It is carried out by using the *Five–Nearest–Neighbors (5NN) fuzzy inferencing method* presented in [Mugica and Cellier, 1993].[7] Here, the five previous recordings with the smallest $\mathcal{L}_2$ norms, i.e., the five nearest neighbors with respect to their input space, are used (if at least five such recordings are found in the behavior matrix), and a distance–weighted average of their fuzzy membership functions is computed and used as the new forecast for the fuzzy membership function of the current state.

From Equation 3.31, the sum of the distances of the five nearest neighbors is:

$$s_d = \sum_{j=1}^{5} d_j \tag{3.32}$$

and the relative distances are given by:

$$d_{rel_j} = \frac{d_j}{s_d} \tag{3.33}$$

By applying this algorithm, the five nearest neighbors can be determined while simultaneously computing their distance functions. Absolute weights are computed as follows [Cellier *et al.*, 1996b]:[8]

[7] Two other fuzzy inferencing methods have been implemented in FIR, they are: the *Mean–of–Maxima (MOM)*, and the *Center–of–Area (COA)*. Both of them are described in [Mugica, 1995], and a comparative study of their application to a linear system is presented in [Mugica and Cellier, 1993].

[8] The equation that was used during most of this dissertation for the computation of the absolute weights was:

$$w_{\mathrm{abs}_j} = \frac{d^2_{\max} - d^2_j}{d_{\max \cdot \mathrm{d_j}}}$$

$$w_{\mathrm{abs}_j} = \frac{1.0}{d_{\mathrm{rel}_j}} \tag{3.34}$$

The absolute weights are real–valued numbers in the range $[0.0, 1.0]$. Using the sum of the five absolute weights:

$$s_w = \sum_{j=1}^{5} w_{\mathrm{abs}_j} \tag{3.35}$$

it is possible to compute relative weights:

$$w_{\mathrm{rel}_j} = \frac{w_{\mathrm{abs}_j}}{s_w} \tag{3.36}$$

Also the relative weights are real–valued numbers in the range $[0.0, 1.0]$. However, their sum is always equal to 1.0. It is therefore possible to interpret the relative weights as percentages. Using this idea, the membership function of the new output can be computed as a weighted sum of the membership functions of the outputs of the previously observed five nearest neighbors:

$$Memb_{out_{new}} = \sum_{\forall j} w_{rel_j} \cdot Memb_{out_j} \tag{3.37}$$

The fuzzy forecasting function usually gives a more accurate forecast than the previously used probabilistic forecasting function [Cellier and Yandell, 1987], as was shown in [de Albornoz and Cellier 1993a]. These results will be presented in Chapter 4. The fuzzy forecasting function of FIR is known as FFORECAST, and takes the syntactic form:

$$[ff_c, ff_m, ff_s] = \text{fforecast } (f_c, f_M, f_s, mask, row_j)$$

where the input variables are:

$f_c$ = is the class history of all previous points;
$f_M$ = is the membership history of all previous points;
$f_s$ = is the side history of all previous points;
$mask$ = is the selected optimal mask;
$row_j$ = pointer that separates past history from behavior to be predicted.

and the output variables are:

$ff_c$ = is the matrix of forecast class values;
$ff_M$ = is the matrix of forecast membership function values;
$ff_s$ = is the matrix of forecast side values.

### 3.3.6 Defuzzification

From the predicted qualitative triples, quantitative estimates of the output variables of the qualitative simulations can then be regenerated. In the fuzzy literature, this process is usually referred to as *defuzzification*. In inductive reasoning terminology, the process is called *regeneration*. A quasi–continuous signal can be obtained from the predicted class, side, and membership values of the qualitative output triples by reversing the process of fuzzy recoding. These values can subsequently be used as input variables to other quantitative models in a mixed quantitative/qualitative simulation environment, as will be shown by means of an example at the end of this chapter. The *regenerate* function of FIR takes the form:

$$r = \text{regenerate }(ff_c, ff_M, ff_s, from, to)$$

where the input parameters are:

$ff_c$ = is the vector of predicted class values;
$ff_M$ = is the vector of predicted membership function values;
$ff_s$ = is the vector of predicted side values;
$to$ = indicates the previously computed landmarks;
$from$ = indicates the corresponding names for the qualitative levels ('1,' '2,' '3,' etc).

and the single output parameter, $r$, is the regenerated quantitative value. Notice that the *from* and *to* parameters have exchanged the roles they had in the *recode* function.

The way in which fuzziness is treated in this dissertation is different from any of the traditionally used techniques. Traditional fuzzifiers do not use the side value. They replace this information by multiple recodings, associating with each quantitative value several qualitative tuples consisting of a class value and its associated membership function value. The fuzzy membership value is used as a *measure of plausibility* of its associated class value. In forecasting, they do not forecast a membership value of the output. Instead, multiple class values of the output variable are forecast, one for each of the feasible class values of the inputs. Defuzzification is then accomplished by computing a weighted average of the forecast class values of the outputs by multiplying them with the values of the membership functions of the associated inputs. Several different defuzzification techniques are quoted in the literature that vary in how the weighted average is computed precisely. Most common are the *Center–of–Area* technique and the *Mean–of–Maxima* approach.

Our approach to fuzzy reasoning offers a clear separation between the fuzzy inferencing stage and the defuzzification stage, whereas these two stages are usually lumped together into one by the more conventional approaches to fuzzy reasoning. The reason for this deviation from the commonly used techniques of fuzzy reasoning is that our approach offers a much better smoothing than the traditional techniques in a data–rich environment. This makes it possible to get away with a smaller number of classes (three to five versus seven to 13 using one of the conventional techniques), which dramatically speeds up the process of qualitative modeling by reducing the associated discrete search space. This is particularly important in the case of multi–input systems.

### 3.3.7 Capabilities of FIR

Now that the methodological pillars of the FIR methodology have been explained in some detail, it may be useful to discuss what can be done with the tool, what types of utilities does the tool offer.

The qualitative modeling engine, implemented by means of the *foptmask* function, allows to learn the behavioral patterns of a system by mere observation. Input/output data streams are fed to the modeling engine, which in turn generates a finite state machine, represented in a compact form through the optimal mask, that captures the causal relationship between the various input signals and the output signal of the system.

The qualitative simulation engine, implemented by means of the *fforecast* function, allows to predict future system behavior given the previously obtained qualitative model. Whereas qualitative modeling represents an *inductive identification* problem (given inputs and outputs, what is the system connecting them?), qualitative simulation represents a *deductive forecasting* problem (given a perceived system model and a set of future input values, what are the corresponding future outputs?).

In our own dialect of fuzzy reasoning, the fuzzy forecasting function predicts future values of the class, side, and membership of the output variables by comparing the class, side, and membership values of the input variables of the *current input record* (the *testing data record* as it would be called in the Neural Network literature) with the corresponding quantities associated with inputs in the *experience data base* (the *training data set*, as it would be called in the Neural Network literature), determining the five records that exhibit the most similar input patterns. The membership value of the current output variable is then computed as a weighted sum of the membership values of the five nearest neighbors, using the nearness of the five neighbors in the input space as a measure of importance when determining the new output.

The two (inductive and deductive) reasoning engines are interfaced to real–world signals through the fuzzification and defuzzification modules, implemented by means of the *recode* and *regenerate* functions, that convert quantitative signals to their qualitative counterparts and vice–versa.

The approach works amazingly well, better so than any of the Ph.D. students that contributed to its development and implementation had expected. Contrary to most other modeling tools, such as the NARMAX approach or neural networks, that presuppose a model structure, thereby effectively parametrizing the identification problem, which is then solved by parameter optimization, FIR does not make any assumption about the system structure at all, and therefore, is less constrained than other modeling approaches. The optimal mask analysis captures the knowledge contained in the input/output data streams reliably and effectively.

The heart of the methodology is really the qualitative modeling engine. The other three modules are more or less straightforward. Success or failure of the approach hinges upon the ability of FIR to extract the knowledge that is implicitly contained in the available observations, and make it explicit. Hence the central feature of FIR is its flexibility and effectiveness in learning behavioral patterns of a system from observations of input/output data streams.

#### 3.3.7.1 Learning Abilities

One of the main characteristics of inductive reasoning is its *learning capability*. The behavior matrix is created and modified in such a way that, for a certain set of inputs, a desired set of outputs can be obtained. All meaningful modeling and simulation schemes must be accompanied by learning capabilities, in order to enable them to cope with a changing world.

Learning can take place during either the *modeling phase* (*off–line learning*, *synthesis*) or during the *simulation phase* (*on–line learning*, *adaptation*). Both types of learning are essential. Off–line learning permits to acquire a new world view, whereas on–line learning enables us to stay up–to–date in an ever changing world. FIR is capable of implementing both of these types of learning.

On a somewhat different scale, learning can take place either *inductively* (learning by acquiring new pieces of information, i.e., new observations) or *deductively* (learning by concluding new facts from previously available knowledge). According to [Sarjoughian, 1995] and from a systems–theoretic perspective, *inductive learning* is identical with learning in the modeling phase: given a set of inputs and an output of a system, inductive learning concludes something about the system connecting these inputs with the single output. *Deductive learning* is identical with learning during the simulation phase: given a set of inputs and a system structure, deductive learning concludes something about the outputs of the system. Finally, *abductive learning* relates to learning control behavior: given a system structure and a set of desired outputs, abductive learning concludes something about the inputs needed to drive the system to the desired outputs. Again, the FIR methodology is capable of implementing all of the above concepts.

Yet another view of learning is possible. Following the learning classification proposed in [Cortés *et al.*, 1993], it can be stated that the FIR methodology is capable of the following three types of learning:

**Learning from observation.** If no other knowledge is available, the FIR methodology is able to learn the behavior of a system by inducing a system structure from observations of its input/output behavior. Learning from observation is synonymous with inductive learning. It can take place both off–line (during the modeling phase) and on–line (during the simulation phase). During the modeling phase, temporal causal relations are discovered among the observed input/output variables in such a way that the system is properly characterized, i.e., the observed outputs can be explained in a reasonably deterministic fashion from the observed inputs. During the simulation phase, newly made observations can be added to the experience data base on the fly, enhancing the reliability of future predictions, or newly made observations can replace previous observations stored in the experience data base enabling the system to cope with time–varying behavior of the system under study.

**Learning from experience.** Learning from experience is a deductive form of learning. New rules are generated by making predictions using the previously induced set of rules and then adding the predictions to the rule base (behavior matrix). Also this type of learning can be applied either in an off–line or in an on–line mode. During the modeling phase, it may be desirable to enlarge the experience data base by making predictions that fall in between observations. This is a form of interpolation. In the fuzzy literature, this technique is called *fuzzy region completion* [Sudkamp and Hammel, 1994]. The technique is particularly useful if not enough measurement data are available. During the simulation phase, it is also possible to add the newly made predictions to the experience data base. However, this can be dangerous since simulation represents some form of extrapolation. Whereas *extrapolation in time* (simulation) is perfectly feasible, qualitative models do not contain sufficient internal validity to support *extrapolation in space* as well. By adding predictions made during simulation to the experience data base, it is possible to fool the system into believing it still makes *reasonable predictions* (i.e., predictions that are justifiable on the basis of the available observations), where, in reality, the system only keeps reiterating on its own previous assumptions.

**Reinforced learning.** The learning process includes a reinforcement measure that describes how good the problem has been solved [Cellier *et al.*, 1996a]. In the FIR methodology, a confidence value is associated with each prediction made. This is essential for preventing the inductive reasoner from reaching conclusions that are not justifiable on the basis of observed behavior. Consequently, such a feature ought to accompany *any* decently robust inductive modeling and simulation scheme. However, the same feature can also be used for reinforced learning. As for the previous two types of learning, also reinforcement can be applied in either off–line or on–line fashion. Often, the goal is to come up with the best possible predictions. However, in other situations, the goal is to come up with the best possible model, i.e., a time–invariant model that best explains the behavior of the system under study for all times to come. In the former case, all available facts (observations) should be used. In the latter case, stray observations (caused either by external disturbances or by unmodeled system dynamics) may weaken the model and should best be eliminated. Reinforced learning can be applied in an off–line fashion by revisiting the observations already made after the model structure has been determined, predicting outputs for the inputs in the training data set and eliminating all input/output pairs from the experience data base that are inconsistent with the predictions made by the model. Reinforced learning is however more commonly applied in an on–line fashion. Rather than adding *all* newly made predictions to the experience data base, only those predictions are added that have a high confidence value. This helps alleviate to some extent the previously mentioned dangers of extrapolation, of designing a system that indulges in self–fulfilling prophecies.

In spite of all the aforementioned advantages, the FIR methodology has also a serious drawback. It is not currently capable of dealing with a large number of behavior trajectories. Particularly, *Optimal Mask Analysis*, the advocated technique for building qualitative models, works fine only if the number of input and output variables is kept within reasonable limits. The technique should not be applied to more than 10 variables at a time. Even 10 variables will make the modeling process sluggish and the experience data base bulky. In most situations dealt with in the past, we have limited the number of variables to no more than five or six. This limitation is in direct conflict with our desire to be able to deal with large–scale systems. Therefore, another technique will be needed to select minimum sets of significant variables grouped as subsystems. To this end, we shall make use of *Reconstruction Analysis*.

## 3.4 FIR Implementation

In order to show how the implementation of FIR is accomplished, a simple linear example with one input and three outputs is introduced. This system is represented by the following differential and algebraic equations:

(3.38)

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{b} \cdot u = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -3 & -4 \end{pmatrix} \cdot \mathbf{x} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot u$$

(3.39)

$$\mathbf{y} \;=\; \mathbf{C}\cdot\mathbf{x}+\mathbf{d}\cdot u \;=\; \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\cdot\mathbf{x}+\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}\cdot u$$

The Bode diagrams for the three output variables can be computed directly in Matlab [MathWorks, 1992a] or ACSL [MGA, 1991]. From those diagrams the bandwidth of the system can be obtained. It is $\omega_{3dB} \approx 1 sec^{-1}$. Therefore, from Equation (3.12), the settling time is $t_{settling} \approx 6\ sec$, and consequently, according to Equation (3.13), the communication interval is $\delta t \approx 3\ sec$. Then, a mask that covers the slowest time constant of the system ($t_{settling}$) must be selected. From Equation 3.14, the depth of the mask must be set to three.

In order to observe all possible frequencies of this system in an optimal manner, the system has been simulated directly in Matlab applying binary random noise to the input signal, trying to excite the system in such a way that all possible states of its output variables are shown. Since the example is very simple, it has been decided to recode each of the output states into just three levels, namely level '1,' level '2,' and level '3.' The input is already binary. Thus, the number of legal states of the qualitative data model can be computed according to Equation (3.2) as:

$$n_{\text{leg}} = \prod_{\forall i} k_i = 2\cdot 3\cdot 3\cdot 3 = 54 \tag{3.40}$$

and therefore, the required number of recordings, also in accordance with Equation (3.2), is:

$$n_{\text{rec}} = 5\cdot n_{\text{leg}} = 270 \tag{3.41}$$

This means that the FIR methodology needs at least 270 samples to characterize the system, i.e., to learn its behavior. Since this is not a real system, but only a mathematical model, the "system" will be simulated in Matlab across 900 seconds of simulated time, equivalent to 300 sampling intervals (or 301 samples), from which 270 will be used to obtain the qualitative model, and the remaining 31 will be used to compare the FIR predictions with the real values.

The Matlab code used to simulate the system is:

```
>>  a = [ 0 1 0 ; 0 0 1 ; -2 -3 -4 ];
>>  b = [ 0 ; 0 ; 1 ];
>>  c = eye(a);
>>  d = zeros(b);
>>  t = 0:3:900;
>>  rand('seed',1);
>>  u = round(rand(301,1));
>>  x0 = zeros(3,1);
>>  [y,x] = lsim(a,b,c,d,u,t,x0);
```

From this simulation, 301 data records, each containing values for the input and output variables, are obtained. Thus, a raw data matrix of 301 rows (simulated data records) and four columns (one input and three outputs) can be formed:

```
>>  meas = [ u , y ];
```

The next step consists of computing the landmarks that separate neighboring classes. Remember that it had been decided to recode the system into three qualitative classes, which means that there will be four landmarks. To obtain the landmarks, the raw data matrix *meas* will first be sorted for simplicity. The Matlab code used for the computation of the landmarks is as follows:

```
>>  m = meas;
>>  veps = 0.000001*ones(1,4);
>>  for i=2:4,
>>    [mi,indx] = sort(meas(:,i));
>>    m(:,i) = mi;
>>  end
>>  LM = [ m(1,:)
>>         0.5*(m(100,:) + m(101,:))
>>         0.5*(m(200,:) + m(201,:))
>>         m(301,:)+veps ];
```

As it can be seen, landmarks are only obtained for the three output variables. This is because the input variable is already binary, and hence does not need to be recoded. The $LM$ matrix containing four rows (the number of landmarks for each variable) and four columns (the number of variables) is:

$$LM = \begin{pmatrix} 0 & -0.0600 & -0.1700 & -0.1200 \\ 0 & 0.1351 & -0.0341 & -0.0431 \\ 1.0000 & 0.3661 & 0.0354 & 0.0413 \\ 1.0000 & 0.5746 & 0.1855 & 0.1304 \end{pmatrix} \tag{3.42}$$

At this point, the variables need to be recoded into the three classes '1,' '2,' and '3' defined by the above landmarks. As in the case of the computation of the landmarks, the input variable will not be recoded. The Matlab code to perform the recoding process is:

```
>>  class1 = meas;
>>  Memb1 = ones(meas);
>>  side1 = ones(meas);
>>  for j=1:301,
>>      if class1(j,1) == 0
>>         side1(j,1) = 1;
>>      else
>>         side1(j,1) = -1;
>>      end
>>  end
>>  to = 1:3;
>>  for i=2:4,
>>    from = [ LM(1:3,i) , LM(2:4,i) ]';
>>    [c,m,s] = recode(meas(:,i),'fuzzy ',from,to);
>>    class1(:,i) = c;  Memb1(:,i) = m;  side1(:,i) = s;
>>  end
```

In this way, a continuous trajectory behavior is recoded into an episodical trajectory behavior. The variables $class1$, $Memb1$, and $side1$ contain the class, membership, and side values of the four variables at each sampling point, respectively. The parameter *from* is a $2 \times 3$ matrix that encodes the landmarks of a single variable. Each column represents one class, whereby the top row contains the lower end and the bottom row the upper end of the range characterizing that class. The parameter *to* is a row vector providing the names (numbers) that represent each of the recoded classes.

Once the recoding process is completed, FIR is able to look for an optimal mask that will represent the static relationship among the input and output variables. It was already mentioned that a mask should only include one value to be forecast, however, this example has three outputs that need to be predicted. Consequently, FIR needs to look for three different optimal masks, one for each output. The first step is to separate the necessary data to carry out the Optimal Mask Analysis, and this is accomplished in the following way:

```
>>  cclass = class1(1:270,:);
>>  MMemb = Memb1(1:270,:);
>>  sside = side1(1:270,:);
```

Then, three different mask candidates must be proposed. Remember that relationships between outputs at the time of forecasting are forbidden. The Matlab code generating the mask candidate matrices is:

```
>>  mcan1 = -ones(3,4);
>>  mcan1(3,2:4) = [ 1 0 0 ];
>>  mcan2 = mcan1;
>>  mcan2(3,2:4) = [ 0 1 0 ];
>>  mcan3 = mcan1
>>  mcan3(3,2:4) = [ 0 0 1 ];
```

The mask candidate matrices are, for the first output variable $y_1$:

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \\ -1 & +1 & 0 & 0 \end{pmatrix} \end{array} \tag{3.43}$$

for the second output variable $y_2$:

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \\ -1 & 0 & +1 & 0 \end{pmatrix} \end{array} \tag{3.44}$$

and for the third output variable $y_3$:

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \\ -1 & 0 & 0 & +1 \end{pmatrix} \end{array} \tag{3.45}$$

The Matlab code to carry out the Optimal Mask Analysis is:

```
>>  [mask1,hm1,hr1,q1,mhis1] = foptmask(cclass,MMemb,mcan1,5)
>>  [Q,indx] = sort(-q1);
>>  m1a = mhis1(:,4*(indx(1)-1)+1:4*indx(1));
>>  m1b = mhis1(:,4*(indx(2)-1)+1:4*indx(2));
>>  m1c = mhis1(:,4*(indx(3)-1)+1:4*indx(3));
```

```
>>  m1 = [m1a,m1b,m1c];
>>  [mask2,hm2,hr2,q2,mhis2] = foptmask(cclass,MMemb,mcan2,5)
>>  [Q,indx]  = sort(-q2);
>>  m2a = mhis2(:,4*(indx(1)-1)+1:4*indx(1));
>>  m2b = mhis2(:,4*(indx(2)-1)+1:4*indx(2));
>>  m2c = mhis2(:,4*(indx(3)-1)+1:4*indx(3));
>>  m2 = [m2a,m2b,m2c];
>>  [mask3,hl3,hr3,q3,mhis3] = foptmask(cclass,MMemb,mcan3,5)
>>  [Q,indx]  = sort(-q3);
>>  m3a = mhis3(:,4*(indx(1)-1)+1:4*indx(1));
>>  m3b = mhis3(:,4*(indx(2)-1)+1:4*indx(2));
>>  m3c = mhis3(:,4*(indx(3)-1)+1:4*indx(3));
>>  m3 = [m3a,m3b,m3c];
```

Notice that the maximum allowed complexity has been set to five, which means that each mask is allowed to have a maximum of five non–zero entries. The matrices $m_{ia}$, $m_{ib}$, and $m_{ic}$ correspond to the best, the second best, and the third best masks among the suboptimal masks of different complexity levels. The optimal masks, $m_{ia}$, found for each output variable are:

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} 0 & -1 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ -3 & +1 & 0 & 0 \end{pmatrix} \end{array} \tag{3.46}$$

for the first output variable, $y_1$,

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} 0 & 0 & -1 & 0 \\ -2 & 0 & 0 & 0 \\ -3 & 0 & +1 & 0 \end{pmatrix} \end{array} \tag{3.47}$$

for the second output variable, $y_2$, and

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ -4 & 0 & 0 & +1 \end{pmatrix} \end{array} \tag{3.48}$$

for the third output variable, $y_3$.

At this point, FIR is prepared to carry out the qualitative simulation across the 31 sampling points that had not been used to characterize the system, i.e., from sampling point 271 to sampling point 301. The forecasting procedure operates in the following way. FIR loops over the 31 steps of the forecast. In each step, the *fforecast* routine is called thrice, using in a row the three optimal masks, $m_{ia}$, to forecast one value only. At the end of the step, the new row (the forecast) is concatenated to the qualitative data from below, and the step counter is incremented. Notice that forecasting is always first attempted with the optimal mask, $m_{ia}$. However, if no value can be predicted with that mask because the input state has never been observed before, the forecast procedure returns the qualitative data unchanged, i.e., the number of rows upon output is the same as upon input, which is tested in the code. If this has happened, the code switches to the next mask, $m_{ib}$, which is usually of lower complexity, or even to the mask $m_{ic}$. If none of the three masks is able to predict the next step, a random number is drawn as an "approximation" to prevent the forecasting process from coming to a halt. The forecasting code in Matlab looks as follows:

```
>>  inpt = class1(271:301,1);
>>  m1a = m1(:,1:4); m1b = m1(:,5:8); m1c = m1(:,9:12);
>>  m2a = m2(:,1:4); m2b = m2(:,5:8); m2c = m2(:,9:12);
>>  m3a = m3(:,1:4); m3b = m3(:,5:8); m3c = m3(:,9:12);
>>  c = cclass;
>>  m = MMemb;
>>  s = sside;
>>  [row,col] = size(cclass);
>>  [n,b] = size(inpt);
>>  for i=1:n,
>>    i,
>>    in  = inpt(i);
>>    fc  = [in,0,0,0];
>>    fcc = [c;fc];
>>    mc  = [1 0.75 0.75 0.75];
>>    mcc = [m; mc];
>>    sc  = [1 1 1 1];
>>    scc = [s; sc];
>>    [ff1,memb11,side11] = fforecast(fcc,mcc,scc,
>>                                    m1a,row+i-1);
>>    [rf,cf] = size(ff1);
>>    if rf ~= row + i,
>>      [ff1,memb11,side11] = fforecast(fcc,mcc,scc,
>>                                      m1b,row+i-1);
>>      [rf,cf] = size(ff1);
>>      if rf ~= row + i,
>>        [ff1,memb11,side11] = fforecast(fcc,mcc,scc,
>>                                        m1c,row+i-1);
>>        [rf,cf] = size(ff1);
```

```
>>        if rf ~= row + i,
>>          ff1 = [ff1;round(rand(1,4))];
>>        end,
>>      end,
>>    end,
>>    [ff2,memb22,side22] = fforecast(fcc,mcc,scc,
>>                                    m2a,row+i-1);
>>    [rf,cf] = size(ff2);
>>    if rf ~= row + i,
>>      [ff2,memb22,side22] = fforecast(fcc,mcc,scc,
>>                                      m2b,row+i-1);
>>      [rf,cf] = size(ff2);
>>      if rf ~= row + i,
>>        [ff2,memb22,side22] = fforecast(fcc,mcc,scc,
>>                                        m2c,row+i-1);
>>        [rf,cf] = size(ff2);
>>        if rf ~= row + i,
>>          ff2 = [ff2;round(rand(1,4))];
>>        end,
>>      end,
>>    end,
>>    [ff3,memb33,side33] = fforecast(fcc,mcc,scc,
>>                                    m3a,row+i-1);
>>    [rf,cf] = size(ff3);
>>    if rf ~= row + i,
>>      [ff3,memb33,side33] = fforecast(fcc,mcc,scc,
>>                                      m3b,row+i-1);
>>      [rf,cf] = size(ff3);
>>      if rf ~= row + i,
>>        [ff3,memb33,side33] = fforecast(fcc,mcc,scc,
>>                                        m3c,row+i-1);
>>        [rf,cf] = size(ff3);
>>        if rf ~= row + i,
>>          ff3 = [ff3;round(rand(1,4))];
>>        end,
>>      end,
>>    end,
>>    ff = [in,ff1(row+i,2),ff2(row+i,3),ff3(row+i,4)];
>>    mf = [1,memb11(row+i,2),memb22(row+i,3),memb33(row+i,4)];
>>    sf = [1,side11(row+i,2),side22(row+i,3),side33(row+i,4)];
>>    c = [c;ff];
>>    m = [m;mf];
>>    s = [s;sf];
>>  end
```

At this moment, the 31 points have been forecast without problem, and are stored in the vectors $c$, $m$, and $s$ which stand for class, membership, and side, respectively. A qualitative comparison between the predicted class values and the previously recoded but not used class values (from 270 to 301) of the quantitative simulation (the “true” values) can be performed, providing a

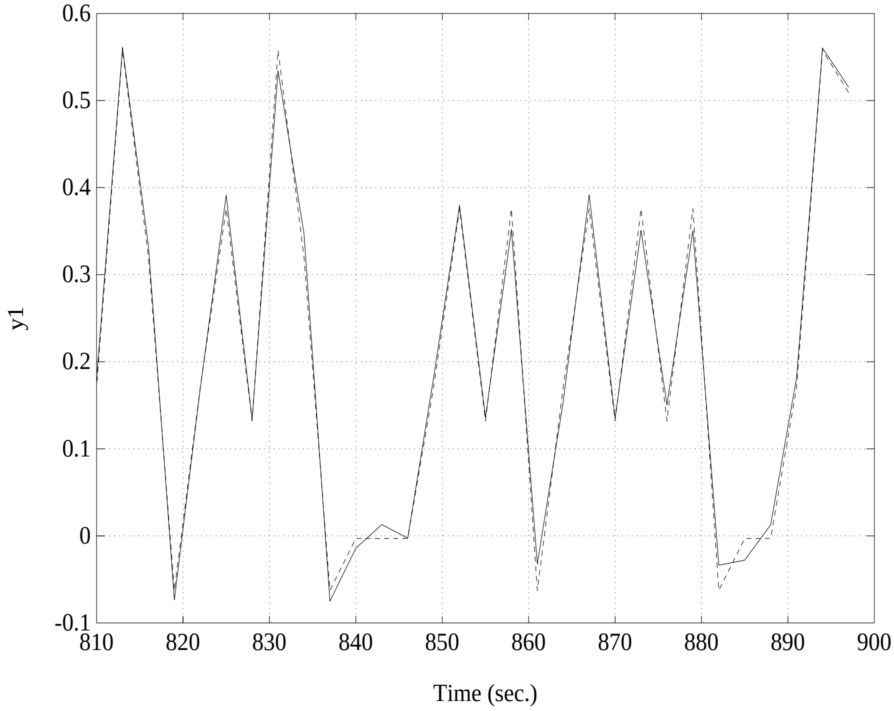


Figure 3.5: Comparison between real and regenerated behavior of output variable $y_1$.

measure of accuracy of the qualitative forecast:

```
>>  frcst = c;
>>  fmemb = m;
>>  fside = s;
>>  frcdat1 = c(271:300,2);
>>  Mfrcdat1 = m(271:300,2);
>>  sfrcdat1 =s(271:300,2);
>>  frcdat2 = c(271:300,3);
>>  Mfrcdat2 = m(271:300,3);
>>  sfrcdat2 = s(271:300,3);
>>  frcdat3 = c(271:300,4);
>>  Mfrcdat3 = m(271:300,4);
>>  sfrcdat3 = s(271:300,4);
>>  simdat = class1(271:300,:);
>>  Msimdat = Memb1(271:300,:);
>>  ssimdat = side1(271:300,:);
>>  frcdat = frcst(271:300,:);
>>  Mfrcdat = fmemb(271:300,:);
>>  sfrcdat = fside(271:300,:);
>>  error = simdat - frcdat;
>>  [ simdat , frcdat , error ]
```

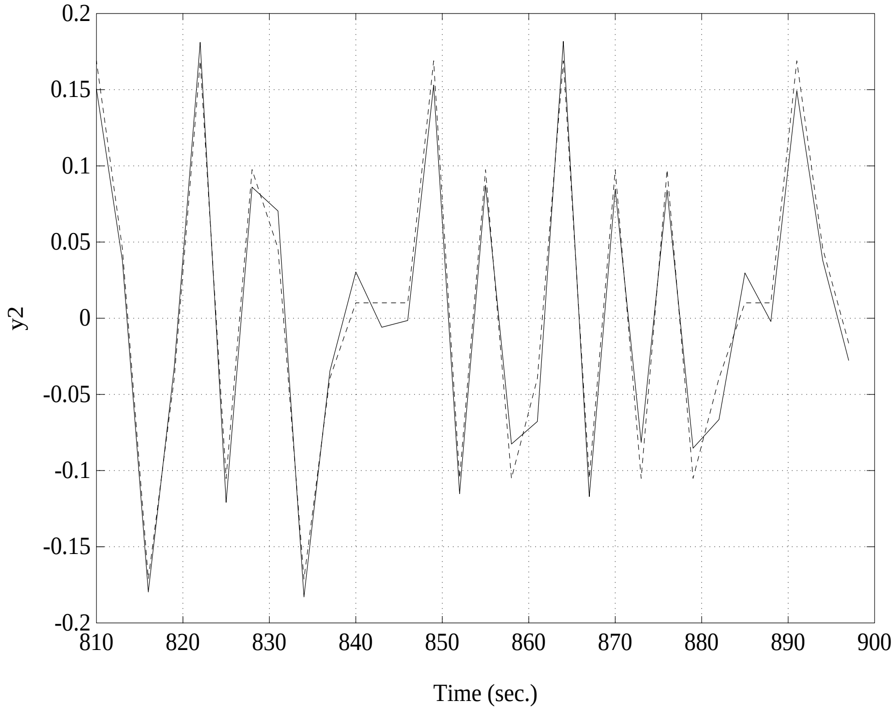


Figure 3.6: Comparison between real and regenerated behavior of output variable $y_2$.

The result of this qualitative comparison is that the forecast thirty data points do not contain even a single class error. The comparison itself is not presented here to save space. However, in more involved applications, the number of errors that can be expected oscillates between 3% and 12% [de Albornoz and Cellier 1993a].

Finally, if the comparison needs to be done quantitatively instead of qualitatively, a regeneration of the predicted qualitative triples should be done. The regeneration process is the inverse operation of the recoding process and works in a very similar way. The Matlab code for the regeneration process is:

```
>> from = 1:3;
>> to = [ LM(1:3,2) , LM(2:4,2) ]';
>> rvar = regenerate(frcdat1,Mfrcdat1,sfrcdat1,from,to);
>> rvar1 = rvar(:);
>> from = 1:3;
>> to = [ LM(1:3,3) , LM(2:4,3) ]';
>> rvar = regenerate(frcdat2,Mfrcdat2,sfrcdat2,from,to);
>> rvar2 = rvar(:);
>> from = 1:3;
>> to = [ LM(1:3,4) , LM(2:4,4) ]';
```

```
>> rvar = regenerate(frcdat3,Mfrcdat3,sfrcdat3,from,to);
>> rvar3 = rvar(:);
```

A graphical comparison of the original behavior (continuous line) with the regenerated behavior (dashed line) is presented in Figures 3.5, 3.6, and 3.7 corresponding to the output variables $y_1$, $y_2$, and $y_3$. It can be seen that the differences between the real and predicted curves are quite small.

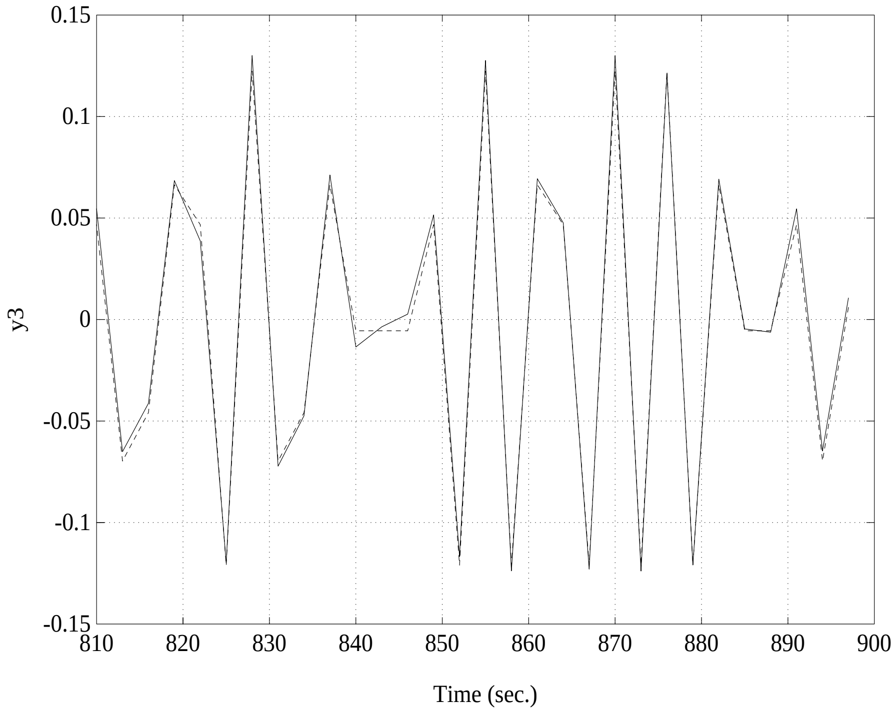


Figure 3.7: Comparison between real and regenerated behavior of output variable $y_3$.

## 3.5 Combining Quantitative and Qualitative Simulation

Combinations of quantitative and qualitative simulation have been attempted during the past years by several research groups, and various approaches have been developed, most of which are referenced in Chapter 2. In this section, FIR is applied to a problem of mixed simulation.

We start out with providing some definitions related to the task at hand[9]. Qualitative simulation can be defined as evaluating the behavior of a system in qualitative terms [Cellier, 1991b]. To this end, the states that the system can be in are lumped together to a finite (discrete) set. Qualitative variables are variables that assume qualitative values. Variables of a dynamical system are functions of time. The behavior of a dynamical system is a description of the values of its variables over time. The behavior of quantitative variables has been defined as *trajectory behavior*, whereas the behavior of qualitative variables has been coined *episodical behavior*. Qualitative simulation can thus be defined as a process of inferring the episodical behavior of a qualitative dynamical system or model.

Qualitative variables are frequently interpreted as an ordered set without distance measure [Babbie, 1989]. It is correct that 'warm' is "larger" (warmer) than 'cold,' and that 'hot' is "larger" (warmer) than 'warm.' Yet, it is not true that

$$\text{`warm'} - \text{`cold'} = \text{`hot'} - \text{`warm'}$$

or, even more absurdly, that

$$\text{`hot'} = 2 * \text{`warm'} - \text{`cold'}$$

Operators such as '$-$' and '$*$' are not defined for qualitative variables.

Time, in a qualitative simulation, is also frequently treated as a qualitative variable. It is then possible to determine whether one event happens before or after another event, but it is not possible to specify when precisely a particular event takes place.

The purpose of most qualitative simulation attempts is to enumerate, in qualitative terms, all possible episodical behaviors of a given system under all feasible experimental conditions. This is in direct contrast to quantitative simulations that usually content themselves with generating one single trajectory behavior of a given system under one single set of experimental conditions.

[9] Some of the terms introduced here have already been met in earlier chapters, but their meaning had always been clear from the context, so it was not necessary to introduce them formally.

### 3.5.1 Mixed Models

In the light of what has been explained above, it seems questionable whether mixed quantitative and qualitative models are feasible at all. How should a mixed quantitative and qualitative simulation deal with the fact that the quantitative subsystems treat the independent variable, *time*, as a quantitative variable, whereas the qualitative subsystems treat the same variable qualitatively? When does a particular qualitative event occur in terms of quantitative time? How are the explicit experimental conditions that are needed by the quantitative subsystems accounted for in the qualitative subsystems?

Quite obviously, a number of incompatibility issues exist between quantitative and qualitative subsystems that must be settled before mixed simulations can be attempted. In a mixed simulation, also the qualitative subsystems must treat time as a quantitative variable. Furthermore, the purpose of qualitative models in the context of mixed simulations is revised. It is no longer their aim to enumerate episodical behaviors. Instead, also the qualitative models are now used to determine a single episodical behavior in response to a single set of qualitative experimental conditions.

Do so revised qualitative models make sense? It is certainly illegitimate to request that, because human aircraft pilots are unable to solve Riccati equations in their heads to determine an optimal flight path, autopilots shouldn't tackle this problem either. It is not sufficient to justify the existence of qualitative models by human inadequacies to deal with quantitative information.

Two good reasons for dealing with information in qualitative ways are the following:

i) Quantitative details about a (sub)system may not be available. For example, in biomedical applications and in large–autonomy systems, while some mechanical properties of the system under study may be well understood and can easily be described by differential equation models, the effects of some interactions on the behavior of the system may be poorly understood and consequently, cannot be easily quantified. A mixed model could be used to describe those portions of the overall system that are well understood by quantitative differential equation models, while other aspects that are less well understood may still be representable in qualitative terms [Nebot, 1994].

ii) Quantitative details may limit the robustness of a (sub)system to react to previously unknown experimental conditions. For example, while a human pilot is unable to compute an optimal flight path, he or she can control the airplane in a much more robust fashion than any of today's autopilots. Optimality in behavior can be traded for robustness. A fuzzy controller and a qualitative Fault Monitoring System are examples of qualitative subsystems that are designed to deal with a larger class of experimental conditions in suboptimal ways [Mugica, 1995].

Mixed quantitative and qualitative models may be used to address either or both of the above applications. However, in order to do so, it is necessary to devise qualitative modeling and simulation capabilities that are compatible with their quantitative counterparts and that can be used to represent qualitative subsystems, such as those mentioned above, appropriately and in terms of knowledge available to the system designer at the time of modeling.

The FIR methodology can be applied to perform such a task. The qualitative subsystems can be modeled by means of fuzzy optimal masks and their episodical behavior can be simulated by means of fuzzy forecasting. The FIR methodology had originally been implemented as toolboxes of Matlab and CTRL–C. However, for the purpose of mixed quantitative and qualitative simulation, a subset of its modules, namely the fuzzy recoding, fuzzy forecasting, and signal regeneration modules have also been made available as an application library of the Advanced Continuous Simulation Language ACSL [MGA, 1991]. This is the software that is being used in mixed quantitative and qualitative simulation runs.

Suppose for example, that the system $S$ shown in Figure 3.8 must be simulated. The system is composed of four subsystems of which $S_1$, $S_3$, and $S_4$ are structurally known subsystems that can be modeled and simulated by means of algebraic and differential equations without much of an effort, whereas subsystem $S_2$ is of a different kind. It may belong to one of two types:

a) the subsystem is basically unknown, and only its interactions with the other subsystems are known, or

b) the subsystem represents a very large and complex plant, the precise details of which are of no immediate concern to the problem at hand, and therefore, a detailed quantitative model would be too costly to develop and/or run.

Hence it may make sense to treat subsystem $S_2$ as a qualitative subsystem and describe it by means of a qualitative modeling and simulation technique.

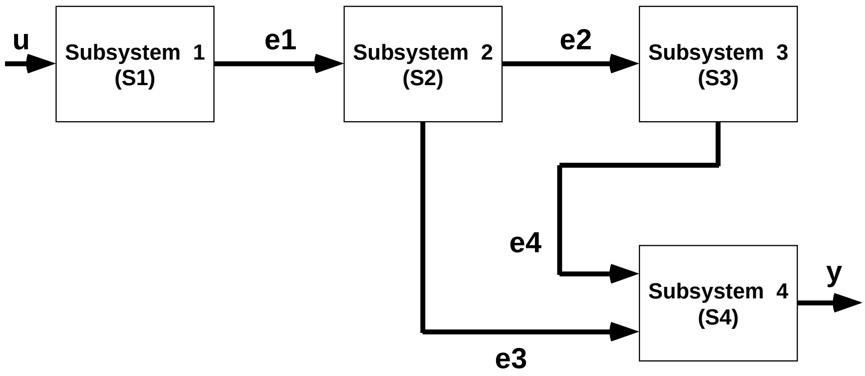


Figure 3.8: System $S$, its subsystems, and the relations between them.

Independently of the used technique, in a mixed simulation environment, qualitative simulation approaches need converters from quantitative values to qualitative values and vice–versa. Thus subsystem $S_2$ should be connected to the quantitative subsystems through two such converters, one for its input variable, $e_1$, and another for its output variables, $e_2$ and $e_3$.

If the qualitative simulation approach is FIR, the input to subsystem $S_2$ should be passed through the fuzzification module converting the quantitative input variable $e_1$, to the qualitative triple $e_1^*$, and the two outputs of $S_2$ should be processed by the defuzzification module converting the qualitative triples $e_2^*$ and $e_3^*$ to the quantitative output variables $e_2$ and $e_3$. The subsystem $S_2$ itself is represented by an optimal mask, to be qualitatively simulated by means of fuzzy forecasting. Figure 3.9 presents an augmented block diagram, in which the converters and the FIR implementation of the qualitative subsystem $S_2$ are schematically shown.

### 3.5.2 Mixed Simulation of a Hydraulic Control System

In the remainder of this chapter, an example will be presented that demonstrates the process of mixed quantitative and qualitative simulation

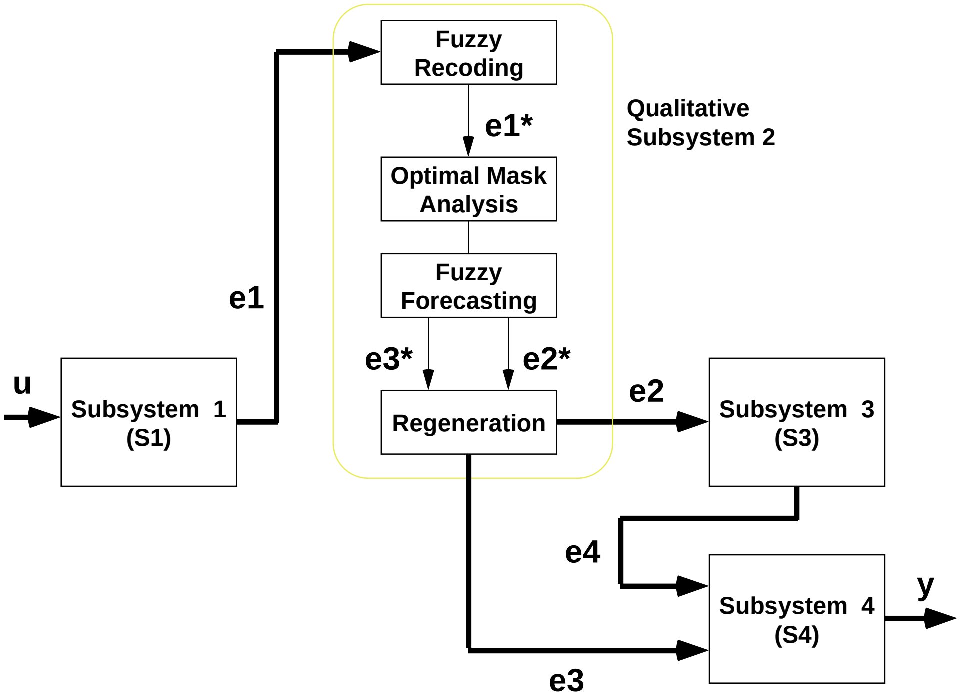


Figure 3.9: Subsystem $S_2$ transformed to a qualitative subsystem.

using Fuzzy Inductive Reasoning. The example was chosen simple enough to be presented in full, yet complex enough to demonstrate the generality and validity of the approach. However, it is not suggested that the chosen example represent a meaningful application of mixed quantitative and qualitative simulation. The example was chosen to prove the concept and to clearly demonstrate the approach, not as a realistic and meaningful application of the proposed technique.

Figure 3.10 shows a hydraulic motor with a four–way servo valve. The flows from the high–pressure line into the servo valve and from the servo valve back into the low–pressure line are turbulent. Consequently, the relation between flow and pressure is quadratic:

$$\begin{aligned} q_1 &= k(x_0 + x)\sqrt{P_S - p_1} \\ q_2 &= k(x_0 - x)\sqrt{p_1 - P_0} \\ q_3 &= k(x_0 + x)\sqrt{p_2 - P_0} \\ q_4 &= k(x_0 - x)\sqrt{P_S - p_2} \end{aligned} \tag{3.49}$$

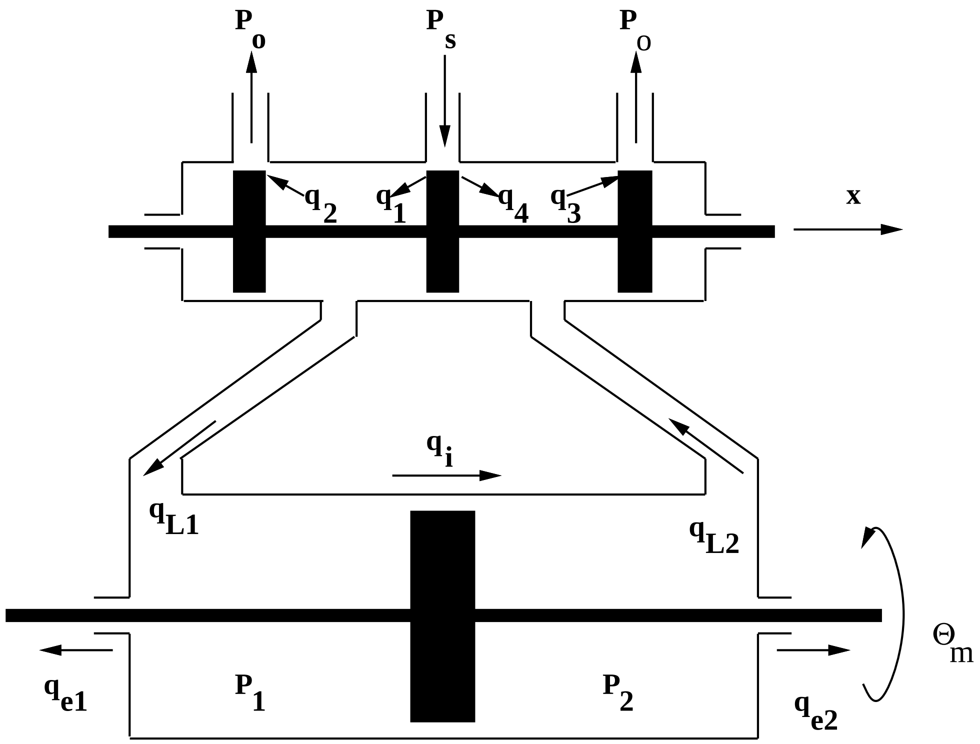


Figure 3.10: Hydraulic motor with a four–way servo valve.

The chosen parameter values are:

$$\begin{aligned}
P_S &= 0.137 \times 10^8 \text{ N m}^{-2}; \\
P_0 &= 1.0132 \times 10^5 \text{ N m}^{-2},\ x_0 = 0.05 \text{ m}; \\
k &= 0.248 \times 10^{-6} \text{ kg}^{-1/2} \text{ m}^{5/2}.
\end{aligned}$$

The change in the chamber pressures is proportional to the effective flows in the two chambers

$$\begin{aligned}
\dot{p}_1 &= c_1(q_{L1} - q_i - q_{e1} - q_{ind}) \\
\dot{p}_2 &= c_1(q_{ind} + q_i - q_{e2} - q_{L2})
\end{aligned} \tag{3.50}$$

with $c_1 = 5.857 \times 10^{13}$ kg m$^{-4}$ sec$^{-2}$. The internal leakage flow, $q_i$, and the external leakage flows, $q_{e1}$ and $q_{e2}$, can be computed as

$$q_i = c_i \cdot p_L = c_i(p_1 - p_2)$$

$$\begin{aligned} q_{e1} &= c_e \cdot p_1 \\ q_{e2} &= c_e \cdot p_2 \end{aligned} \tag{3.51}$$

where

$c_i = 0.737 \times 10^{-13}$ kg$^{-1}$ m$^4$ sec, and
$c_e = 0.737 \times 10^{-12}$ kg$^{-1}$ m$^4$ sec.

The induced leakage, $q_{ind}$, is proportional to the angular velocity of the hydraulic motor, $\omega_m$

$$q_{ind} = \psi \cdot \omega_m \tag{3.52}$$

with $\psi = 0.575 \times 10^{-5}$ m$^3$, and the torque produced by the hydraulic motor is proportional to the load pressure, $p_L$

$$T_m = \psi \cdot p_L = \psi(p_1 - p_2) \tag{3.53}$$

The mechanical side of the motor has an inertia, $J_m$, of 0.08 kg m$^2$, and a viscous friction, $\rho$, of 1.5 kg m$^2$ sec$^{-1}$.

The hydraulic motor is embedded in the control circuitry shown on Figure 3.11. In the mixed quantitative and qualitative simulation, the mechanical and electrical parts of the control system will be represented by differential equation models, whereas the hydraulic part will be represented by a fuzzy inductive reasoning model.

For this purpose, it was assumed that no knowledge exists that would permit a description of the hydraulic equations by means of a differential equation model. All that is known is that the mechanical torque, $T_m$, of the hydraulic motor somehow depends on the control signal, $u$, and the angular velocity, $\omega_m$.

For validation purposes, the mixed simulation results will be compared with previously obtained purely quantitative simulation results. The purely quantitative simulation of the overall system was simulated over 2.5 seconds. A binary random input signal was applied to the input of the system, $\theta_{set}$.

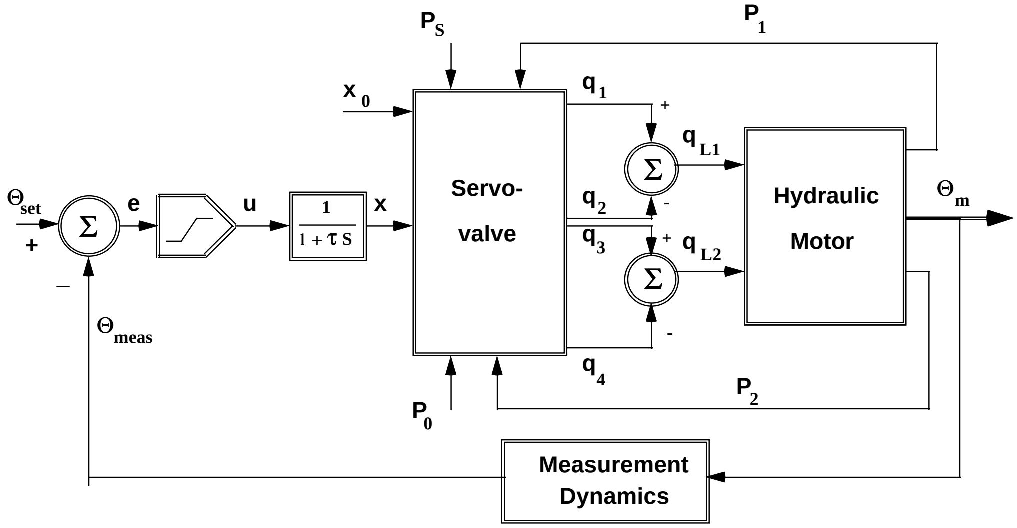


Figure 3.11: Hydraulic motor position control circuit.

The values of the control signal, $u$, the angular velocity, $\omega_m = \dot{\theta}_m$, and the torque, $T_m$, from the first 2.25 seconds of the quantitative simulation were recoded to generate the fuzzy inductive model of the hydraulic motor.

The values of the last 0.25 seconds of quantitative simulation were stored for validation purposes. Validation is accomplished by comparing the simulation results of the new mixed model with those of the purely quantitative model, which is being used in the place of "measurement data."

#### 3.5.2.1 Building the Fuzzy Inductive Model

As was described in the previous sections of this chapter and was shown in the linear system example, the fuzzy inductive model is constructed in two steps. In the first step, the quantitative data are recoded, and in the second step, the fuzzy optimal mask is determined from the recoded data.

**Fuzzy Recoding of the Hydraulics**

The first question to be addressed in the recoding process is the selection of the appropriate sampling rate (communication interval) for the continuous variables to be recorded (either from measurements or, as in this example, from a quantitative simulation study). In the given example, this value can be deduced from the longest time constant to be considered (i.e., the inverse of the slowest eigenvalue of the Jacobian). The eigenvalue is at $-20$, and therefore, the longest time constant is 0.05 seconds. In accordance with Equations (3.13) and 3.14, the three variables $u$, $\omega_m$, and $T_m$ must thus be sampled once every 0.025 seconds, and a mask of depth three must be chosen.

Unfortunately, fuzzy inductive forecasting predicts only one value of $T_m$ per sampling interval. Thus, the mixed qualitative and quantitative simulation behaves like a sampled–data control system with a sampling rate of 0.025 seconds. Thereby, the stability of the control system is lost because the sampling rate is too slow to keep up with the changes in the system. From a control system perspective, it is necessary to sample the variables considerably faster. An ACSL program was coded to study different sampling rates in order to obtain a stable control performance. This program introduces into the quantitative simulation a delay in the computation of the torque. The largest delay time that could be introduced without losing stability of the control system was identified. It was determined that the longest tolerable delay is 0.0025 seconds. Consequently, the mask depth must be increased from three to 21, i.e., the depth of the mask covers the slowest time constant, while $\delta t$ satisfies the stability requirements of the control system.

The next step is to find the number of discrete levels into which each of these variables will be recoded. For the given example, it was decided that all three variables can be sufficiently well characterized by three levels. A discretization of the variables in this manner implies, in accordance with Equation (3.2), that the number of legal states is 27 ($3 \times 3 \times 3$).

As explained before, it is desirable to record each state at least five times. Consequently, a minimum of 130 recordings, corresponding to a total simulation time of 0.325 seconds, is needed. However, due to the mismatch between the sampling rate required by fuzzy forecasting and the actually used sampling rate that is required due to the control characteristics, considerably more data are needed. It was decided to choose a total simulation time of 2.5 seconds with 2.25 seconds being used for model identification, and the remaining 0.25 seconds being used for validation. This provides the optimal mask module with 900 recordings used for model identification, while fuzzy forecasting is carried out across the final 100 steps.

While the qualitative subsystem is analyzed in off–line mode, as during the recoding and the Optimal Mask Analysis, the Matlab FIR modules can be used. However, once the qualitative subsystem is connected to the quantitative subsystems for the purpose of mixed simulation, the ACSL FIR modules should be used instead.

### Fuzzy Optimal Mask of the Hydraulics

With the data recoded as described above, it is possible to construct a qualitative model of the hydraulics by means of the Optimal Mask Analysis procedure. To combine the qualitative and quantitative simulation models, it is necessary to solve the dynamic stability problem while covering with the mask the longest time constant to be captured in the qualitative model. This means that, as mandated by control theory, the sampling interval $\delta t$ is chosen to be 0.0025 seconds. Consequently, the mask depth must be selected equal to 21. Even a search through all possible masks of complexities up to seven only would be painfully slow. Therefore, the following approach was taken. From the point of view of fuzzy reasoning, a mask depth of three is usually sufficient. Consequently, it was decided to consider only inputs in the first, the 11$^{\text{th}}$, and the 21$^{\text{st}}$ row of the mask, blocking all other rows out by setting the corresponding elements of the mask candidate matrix to 0. In this way, the search can proceed quickly, and yet, the resulting "optimal" mask will still be very close to the truly optimal mask. Thus, the following mask candidate matrix of depth 21 was chosen:

$$
\begin{array}{l}
{}_t\backslash^x \\
t-20\delta t \\
t-19\delta t \\
\vdots \\
t-11\delta t \\
t-10\delta t \\
t-9\delta t \\
\vdots \\
t-\delta t \\
t
\end{array}
\begin{array}{c}
\begin{array}{ccc} u & \omega_m & T_m \end{array} \\
\begin{pmatrix}
-1 & -1 & -1 \\
0 & 0 & 0 \\
\vdots & \vdots & \vdots \\
0 & 0 & 0 \\
-1 & -1 & -1 \\
0 & 0 & 0 \\
\vdots & \vdots & \vdots \\
0 & 0 & 0 \\
-1 & -1 & +1
\end{pmatrix}
\end{array}
\tag{3.54}
$$

Consequently, the mechanical torque, $T_m$, at time $t$ may depend on the current values of $u$ and $\omega_m$, as well as on past values of $u$, $\omega_m$, and $T_m$ at times $t - 0.025$ seconds and $t - 0.05$ seconds.

The fuzzy optimal mask is obtained using once again the *foptmask* function coded in Matlab that makes use of the previously recoded data. The following optimal mask has been found for this example:

$$
\begin{array}{l}
{}_t\backslash^x \\
t-20\delta t \\
t-19\delta t \\
\vdots \\
t-11\delta t \\
t-10\delta t \\
t-9\delta t \\
\vdots \\
t-\delta t \\
t
\end{array}
\begin{array}{c}
\begin{array}{ccc} u & \omega_m & T_m \end{array} \\
\begin{pmatrix}
0 & -1 & -2 \\
0 & 0 & 0 \\
\vdots & \vdots & \vdots \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\vdots & \vdots & \vdots \\
0 & 0 & 0 \\
-3 & 0 & +1
\end{pmatrix}
\end{array}
\tag{3.55}
$$

In other words:

$$
T_m(t) = \tilde{\mathrm{f}}(\omega_m(t-0.05), T_m(t-0.05), u(t)) \tag{3.56}
$$

#### 3.5.2.2 Fuzzy Forecasting and Signal Regeneration

Once the optimal mask has been determined and before it can be integrated into the mixed simulation, its prediction capability must be checked. For this purpose, the *fforecast* function of Matlab will be used. The point is to compare the values of $T_m$ obtained from the quantitative simulation with those obtained with the forecast and regeneration FIR modules, i.e., those obtained by means of qualitative simulation. As mentioned before, the first 900 rows of the qualitative data model were used as past history data to compute the optimal mask. Fuzzy forecasting is being used to predict new qualitative triples for $T_m$, but only for the last 100 rows of the qualitative data model. From the predicted qualitative triples, quantitative values are then regenerated.

Figure 3.12 compares the true "measured" values of $T_m$ obtained from the purely quantitative simulation (solid line) with the forecast and regenerated values obtained from Fuzzy Inductive Reasoning (dashed line). The results are encouraging. Quite obviously, the optimal mask contains sufficient information about the behavior of the hydraulic subsystem to be used as a valid replacement of the true quantitative differential equation model, although the chosen recoding scheme was extremely crude using three levels for each variable only. Notice that the fuzzy inductive reasoning model was constructed solely on the basis of measurement data. No insight into the functioning of the hydraulic subsystem was required other than the knowledge that the torque, $T_m$, dynamically depends on the control signal, $u$, and the angular velocity, $\omega_m$.

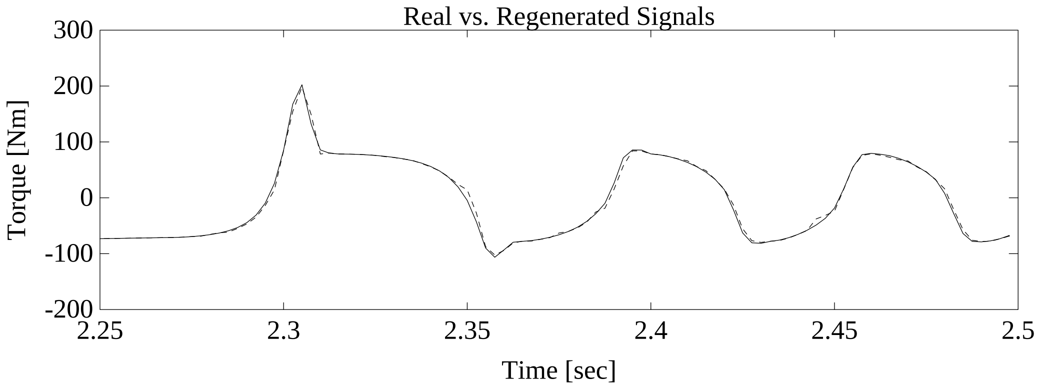


Figure 3.12: Comparison between real and regenerated torque behavior.

### 3.5.3 Mixed Modeling and Simulation

Once the prediction capability has been demonstrated, the fuzzy inductive reasoning model can be used to replace the former differential equation model of the hydraulic subsystem in an on–line mixed simulation, where the electrical and mechanical subsystems are still modeled using differential equations, whereas the hydraulic subsystem is modeled using a fuzzy optimal mask. The mixed model is shown on Figure 3.13.

The quantitative control signal, $u$, is converted to a qualitative triple, $u^*$, using fuzzy recoding. Also the quantitative angular velocity, $\omega_m$, of the hydraulic motor is converted to a qualitative triple, $\omega_m^*$. From these two qualitative signals, a qualitative triple of the torque of the hydraulic motor, $T_m^*$, is computed by means of fuzzy forecasting. This qualitative signal is then converted back to a quantitative signal, $T_m$, using fuzzy signal regeneration.

Forecasting was restricted to the last 100 sampling intervals, i.e., to the time span from 2.25 to 2.5 seconds. Figure 3.14 compares the angular position, $\theta_m$, of the hydraulic motor from the purely quantitative simulation (solid line) with that of the mixed quantitative and qualitative simulation (dashed line). As was to be expected, the mixed model behaves like a sampled–data control system. The mixed simulation exhibits an oscillation amplitude that is slightly larger and an oscillation frequency that is slightly smaller than those shown by

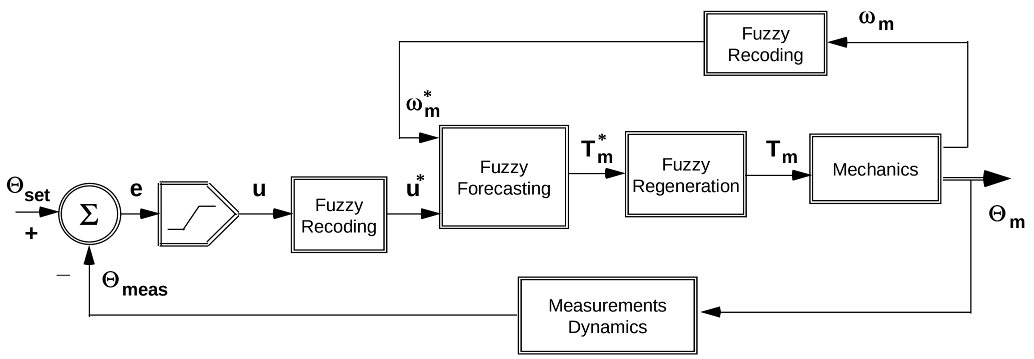


Figure 3.13: Mixed model of the hydraulic system.

the purely quantitative simulation. Surprisingly, the damping characteristics of the mixed model are slightly better than those of the purely quantitative model.

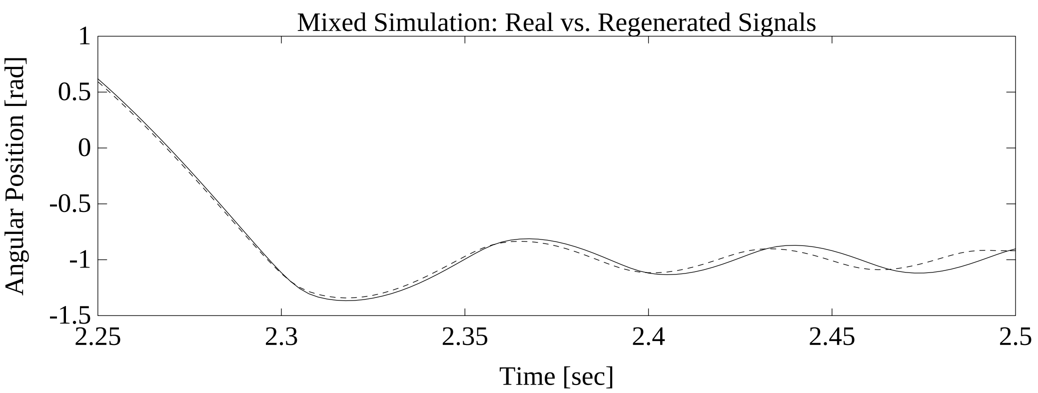


Figure 3.14: Comparison between real and regenerated closed–loop behavior.

## 3.6 Conclusions

The examples demonstrate the validity of the chosen approach. Mixed simulations are similar in effect to sampled–data system simulations. *Fuzzy recoding* takes the place of analog–to–digital converters, and *fuzzy signal regeneration* takes the place of digital–to–analog converters. However, this is where the similarity ends. Sampled–data systems operate on a fairly accurate representation of the digital signals. Typical converters are 12–bit converters, corresponding to discretized signals with 4096 discrete levels. In contrast, the fuzzy inductive reasoning model employed in the above examples recoded all three variables into qualitative variables with the three levels 'small,' 'medium,' and 'large.' The quantitative information is retained in the fuzzy membership functions that accompany the qualitative signals. Due to the small number of discrete levels, the resulting finite state machine is extremely simple. Fuzzy membership forecasting has been shown to be very effective in inferring quantitative information about the system under investigation in qualitative terms.

Some of the principal advantages of Fuzzy Inductive Reasoning are the following:

1. Inductive reasoning allows qualitative models to treat time as a continuous (quantitative) variable. This is of primary importance if modeling and simulation of mixed quantitative and qualitative systems is to be attempted.

2. The technique can be applied to any system available to experimentation and observation. Inductive reasoning is fully based on behavior, thus, there is no need for knowing the internal structure of the system.

3. The methodology contains an inherent model validation mechanism inside its qualitative simulation engine that prevents it from reaching conclusions that are not justifiable on the basis of the available facts.

4. Inductive reasoners operate internally in a qualitative fashion just like knowledge–based reasoners. Therefore, it is possible to apply meta–knowledge to improve the performance and quality of the inference engine, and it is also possible to trace back the reasoning process if desired.

In this dissertation, mixed quantitative and qualitative simulation is applied to fault monitoring and analysis in large–scale systems. The plant itself is represented by a purely quantitative differential equation model (since no real large–scale systems were at our disposal to experiment with), whereas the fault monitor is a qualitative model that predicts system behavior in parallel with the quantitative plant model, and reasons about the integrity of the plant and its controllers. The subsequent chapters of this thesis shall demonstrate how this is done.

# Chapter 4

# Building a Qualitative Fault Monitoring System

## 4.1 Introduction

In fault monitoring of large–scale systems, any fault, transient, or malfunction, i.e., any misbehavior, can be seen as a *structural change*. The differences between such a system and a "true" variable structure system (VSS)[1] can be stated from two different perspectives; on the one hand, from the point of view of the normal modes the system can be operating in; and on the other hand, taking into account the purpose of the Fault Monitoring System (FMS).

In large–scale systems such as those defined in Section 1.1, there is usually only one normal mode in which the system is operating at any one time, and consequently, all unexpected (unscheduled) structural changes encountered are related to malfunctions, transients, and/or accidents taking place. The purpose of a supervision and decision support system is to detect the anomalous behavior of the system, or the paths that will lead to it, and take appropriate actions to correct the problem, providing the supervisory control with sufficient information to deal with a developing emergency efficiently and effectively, or to prevent it from happening altogether.

[1] cf. Section 1.1.3 for a definition of variable structure system.

In "true" variable structure systems, the transition from one structural mode to another is not related to an emergency, but represents a normal event that will happen regularly during system operation. The purpose of the supervisory controller when applied to a VSS, is to provide the mode transition controller (usually either a hardwired controller or a software controller) with sufficient information to guarantee a smooth transition between different operational modes.

In both cases, the information obtained by the system that monitors structural changes must be provided to the control system on–line and very fast. The mode transition must be detected and diagnosed as quickly as possible. Most physical plants are characterized by an upper time limit before which the controller must react to a structural change that took place in the plant. The example of an electrical grid with several interconnected power plants can be used to show how fast the control system must react to a structural change. Suppose that the controller of a hydraulic turbine (in one of the power plants) will naturally react to a short circuit (a structural change) in the electrical net by opening up the sluice to increase the power of the turbine to compensate for the sudden jump in perceived load. Whereas this control action is entirely appropriate in the case of a true load increase, it is not appropriate in the case of a short circuit. The fault monitoring system has two seconds time to discover that a structural change took place and what the nature of it is, before the growing angular velocity of the turbine will trip the emergency shut–down system. A properly functioning fault monitor should react faster than the emergency shut–down system, detect that something abnormal has happened, discover that a short circuit has occurred, determine where in the net the short circuit is located, isolate the short circuit by notifying affected neighboring power stations of the problem, throw off the isolated load, and notify power stations in the vicinity that are not directly affected by the short circuit of the expected transient behavior (a scheduled structural change). If the fault monitoring system fails to react faster than the emergency shut–down system, neighboring power stations will experience an unexpected additional sudden load increase. Since one power station got shut down, the others have to take over its customer load. This load increase may be viewed by the neighboring power stations as a new fault, and a wave of emergency shut–downs may result in the potential black–out of an entire region.

The methodology presented here is capable of diagnosing faults in large–scale systems and structural changes in variable structure systems alike. The design of a decentralized hierarchical fuzzy inductive reasoning architecture for fault detection and diagnosis will be presented in this chapter. The advantages and disadvantages of this Fault Monitoring System will be demonstrated by means of some prototypical implementation of such a system in its different operating modes, applied to a fairly simple model of a Boeing 747 aircraft at cruise flight (this chapter), and a sophisticated large–scale model of a Boiling Water Nuclear Reactor at 100% power (Chapter 7), as well as for diagnosing structural changes in two VSSs, namely an interconnected water–tank example, and an electrical switching circuit example (this chapter).

## 4.2 Fault Detection Through Inductive Reasoning

Fault detection through inductive reasoning can be decomposed into two different phases, a) off–line and b) on–line. The off–line phase includes the processes that must be carried out before the qualitative inference process can be attempted, i.e., the fuzzification and qualitative modeling processes described in Chapter 3, as well as the characterization of the faults the FMS should be able to recognize. The main tasks of the off–line phase are:

i) Selection of input/output variables.

ii) Causal and temporal grouping of variables.

iii) Qualitative modeling.

iv) Hierarchical arrangement of qualitative models of subsystems.

The only condition for carrying out these off–line processes is that knowledge about the system's behavior must be available.

In the on–line phase, the FMS is coupled to a quantitative dynamic system or model in order to accomplish the tasks for which it has been designed. These tasks depend on the system under study, and can be those of detecting anomalous behavior in dynamic systems, or detecting structural changes in VSS or both. The processes of the on–line phase[2] are those described in Chapter 2:

- Detection.

[2]Chapter 2 provides a full description and complete references of each one of these processes.

- Isolation.
- Characterization.
- Identification.
- Propagation through the hierarchy.
- Diagnosis.
- Analysis.

### 4.2.1 Selection and Causal Grouping of Variables

The process of properly identifying the variables to reason with and the subsystems making use of these variables, is one of the most important and difficult problems to be solved on the way of designing a supervision and decision support system. To solve this problem, the Optimal Mask Analysis technique, described in Chapter 3, has been implemented to identify clusters of causally related variables that can be isolated as subsystems in a model hierarchy.[3] Structural knowledge of the physical system can also help in this endeavor, but the identified subsystems do not necessarily coincide with physical subsystems. The variables making up one reasoning subsystem are selected by the Reconstruction Analysis and/or Optimal Masks Analysis methodologies on the basis of similarities in their frequency characteristics and causal relations rather than on the basis of geometric topology.

Optimal Mask Analysis has already been explained in Chapter 3; however, its use for selecting a minimum set of meaningful variables has not been treated yet. This selection is made by means of a comparison between different possible qualitative relationships, each one with a different combination of variables, i.e., a comparison between different optimal masks, one for each combination of variables. The mask with the best quality not just represents the best qualitative model, but the strongest relationship between input and output variables from the point of view of their forecasting power. The purpose is then that of finding strong qualitative (non–linear) correlations between inputs and outputs, and weak correlations among inputs, in such a way that the resulting relations for the outputs will be as deterministic as possible while avoiding unnecessary redundancy among highly correlated inputs.

[3]It has already been mentioned that when the system under consideration is a large–scale system, the Optimal Masks Analysis technique should be combined with the Reconstruction Analysis methodology that will be explained in Chapter 5.

If the chosen set of variables is not representative enough of the subsystem to be characterized, the resulting optimal masks will have few interactions among their variables, which in turn will lead to poor propagation of information up the hierarchical ladder (the necessity for a hierarchical arrangement of the selected sets of variables will be explained in the next section). The same can be observed if the chosen variables are too strongly correlated, since, in this case, the complexity of the search space is enhanced without significantly augmenting the amount of available information.

Also, the subsystems should complement each other in an optimal manner. Subsystems that are too independent of each other show few interactions between them, so that the higher hierarchical levels of the overall architecture do not contribute significantly to the reasoning process, but simply accumulate and propagate further the findings of subordinate reasoners. On the other hand, a duplication of reasoning capabilities within different subsystems located at the same hierarchical level simply increases the complexity of the search space of the supervisory reasoner without providing it with additional information that would justify the enhancement of its complexity.

It may happen that the system under study is composed of lots of variables, thus, the number of possible subsets of variables to be tried is too high. It seems not practical to perform an exhaustive test of all possible combinations of variables. Consequently, the Optimal Masks Analysis is not the appropriate tool for selecting a minimum set of meaningful variables to reason with, in this case. To solve this problem, the Reconstruction Analysis methodology will be introduced in Chapter 5. In this way, if the system to be fault monitored is a large–scale system, Reconstruction Analysis will be in charge of selecting the variables, whereas Optimal Mask Analysis will be in charge of obtaining the qualitative models relating those variables to each other.

### 4.2.2 Hierarchical Fault Monitoring

Since Optimal Mask Analysis inherently involves *time*, its causality analysis is necessarily temporal. This means that the identified clusters of meaningful variables (subsystems) are temporally related to each other. The subsystems should be hierarchically arranged. Each hierarchical level can be composed of several subsystems except for the highest level, which can include one subsystem only. Since each subsystem is modeled by an optimal mask, the hierarchy of subsystems is in fact modeled as a hierarchy of fuzzy inductive reasoners (FIRs).

Hierarchization of inductive reasoners can proceed in two ways. In a *top–down* approach, one starts with an overall system output and determines an optimal mask that estimates its value from a set of selected input variables. If any of these "input" variables are not true inputs of the system but are only intermediary variables that may not be accessible at run time, these "inputs" are declared as new outputs of subsystems, and a new optimal mask is determined for each of them relating that variable to other variables, until all "inputs" have been reduced to true system inputs. In a *bottom–up* approach, one starts with individual subsystems at the lowest level, determining optimal masks for them. The output variables of each subsystem are then propagated up to the next stage of the reasoner to determine the qualitative behavior of the composite system located at the next higher hierarchical level.

In a FMS, both approaches may be combined. Modeling the behavior of physical variables is usually better done in a top–down approach, i.e., when trying to obtain a qualitative model that explains a physical variable for which data are available in the training data set, it is better to start out with this variable as the first output and work oneself backward to the system inputs. However, the final output of the fault monitor is not normally a physical variable. It is a global indicator that tells whether or not a fault has occurred. It is usually computed as a logical function of similar but local indicators generated by individual subsystems. Each of these indicator functions, in turn, is computed from differences between an observed physical variable and a predicted estimate of the same physical variable produced by a yet lower–level inductive reasoner. Consequently, it is better to build the higher–level reasoners that evaluate the indicator functions using a bottom–up approach, whereas the lower–level reasoners that predict the behavior of physical variables are best constructed using a top–down approach.

The highest FIR in the hierarchy is known as the *executive FIR*. The executive FIR uses as inputs the output signals of the subsystem FIRs, and its output is one of the "true" outputs of the system. The role of the higher–level subsystem FIRs is that of *sensor fusion* [Luo and Kay, 1989; Pau, 1989], i.e., they concentrate the information available through the large number of sensors to a much smaller number of signals that the executive FIR can be expected to handle effectively and efficiently. The executive FIR will report its findings while pointing out which of the subsystem FIRs is most closely related to the problem. Then, it will turn to that FIR to receive more detailed information and to continue with the fault characterization. Each of the subsystem FIRs operates on differences between the observed and expected behaviors of physical variables. The expected behaviors are calculated by yet lower–level FIRs from the physical plant inputs.

### 4.2.3 Detection

The FIR methodology predicts the system behavior in qualitative terms. This behavior is then compared against the real values obtained from either a quantitative model of the system or from the physical system itself. As the prediction is based on the recent past behavior of the system, it is somewhat adaptive to slow changes in system parameters or a slow drift in the steady–state, but a fault, transient, or structural change is immediately detected since the behavior of the system can no longer be predicted with the fuzzy optimal mask that had been determined for the previously active system structure. Figure 4.1 depicts the fault detection scheme using FIR.

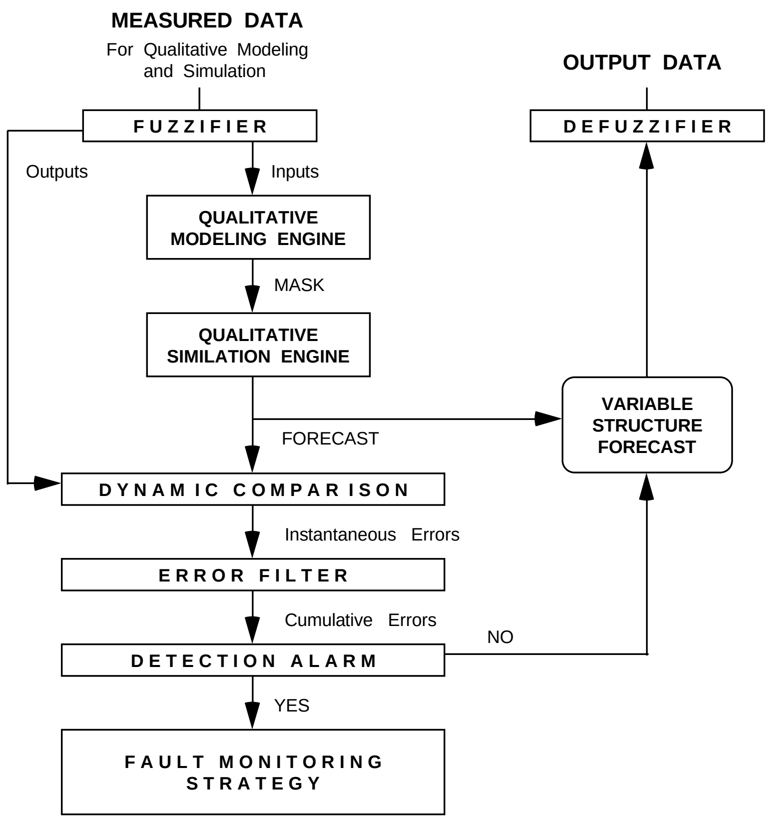


Figure 4.1: Fault detection using FIR.

The detection of a structural change is made through a *Dynamic Comparison Module* that continuously compares the forecasting results obtained by the inductive reasoner (the qualitative simulation results) against the real measurement data obtained from either a quantitative model or a physical system. Since the fuzzy optimal mask operates on qualitative information, it is to be expected that the forecast is not perfect. For a typical technical application, we may expect a forecasting error of somewhere in the order of 3–12% [de Albornoz and Cellier, 1993a; de Albornoz and Cellier, 1994] in the FIR predictions. Evidently, it will sometimes happen that a prediction is kind of poor even if the correct model is in use, or it may also happen that a prediction is right on the mark even though the incorrect model is being used. In both cases, an alarm basing its decision on instantaneous errors alone would be destined to make mistakes.

In order to overcome this problem, the instantaneous error vector is first being "refined" by sending it through an *Error Filter Module*. The error filter accumulates errors over $k$ steps, i.e., it generates a moving average of the errors accumulated over the most recent $k$ steps. The *Alarm Module* then bases its decision on the comparison between the number of accumulated errors and a certain threshold specified by the modeler. In this way, local aberrations or accidental hits can be filtered out and will not influence the decision making process.

Figure 4.2 shows an example of fault detection using the aforementioned process. The Dynamic Comparison Module generates an instantaneous error for each optimal mask output variable at each time interval. The instantaneous errors are stored in a matrix where the number of rows is $t + n\Delta t$ and the number of columns is the number of optimal masks in the hierarchy. In this example there are five optimal masks in the hierarchy. A moving average error filter is shifted further down computing, for each optimal mask, the sum of the instantaneous errors it covers. These instantaneous errors are stored in a cumulative errors matrix. If any of these values surpass the threshold of the alarm module, then the alarm is triggered immediately for those particular masks, i.e., the faulty subsystems are identified.

### 4.2.4 Isolation and Characterization

Once the alarm has been triggered because an anomalous behavior has been detected, the FMS proceeds to the next stages of automated fault monitoring, i.e., isolation and characterization. The isolation step is carried out by the executive FIR that knows which among its input variables (output variables from other subsystems) are responsible for triggering the alarm. Tracing back these variables into lower levels of the hierarchical ladder, the executive

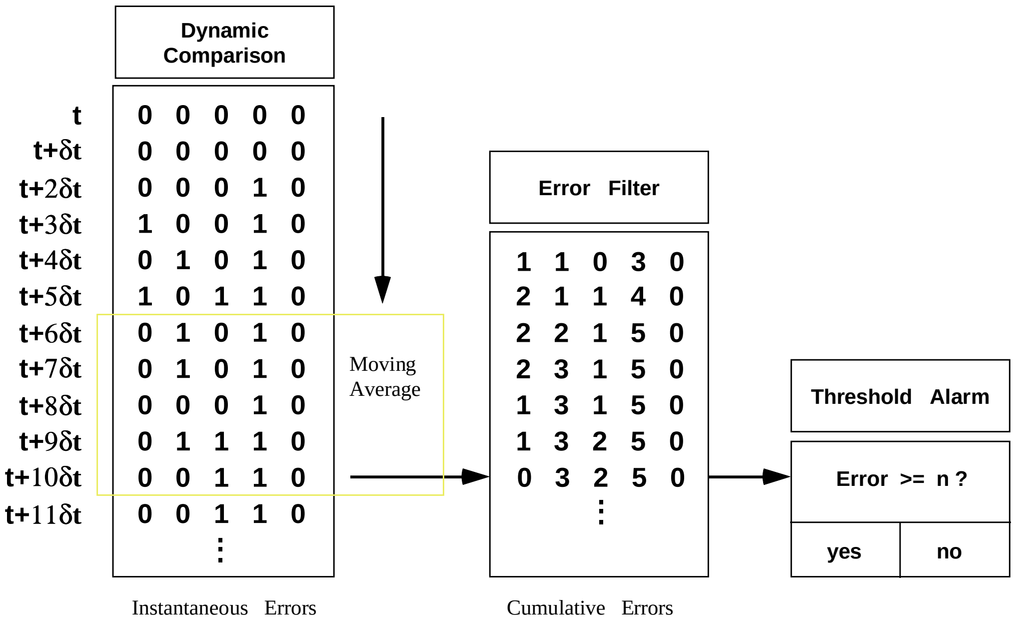


Figure 4.2: Fault detection scheme.

FIR determines which subsystem or subsystems have failed. In this way, the detected anomaly is restricted to a set of few candidate subsystems that may have caused it.

The purpose of the characterization process is to classify the anomalous behavior in order to simplify the hypothesis formulation process of the diagnosis, i.e., to find out what kind of failure has been detected in order to reduce the number of possible causes. The characterization is done by comparing the behavior obtained from the set of variables that compose the affected subsystems in the real physical system or quantitative model against the behavior obtained from previously characterized qualitative models of those subsystems under different conditions of anomalous behavior.

Up to this point, the anomalous behavior, transient, or fault, (from the point of view of qualitative fault monitoring a structural change) has been detected, isolated, and characterized. In the next section, different strategies will be presented for tackling the diagnosis and analysis processes, and also for continuing the fault monitoring duties after a fault has taken place, that is, after a structural change has occurred.

### 4.2.5 Diagnosis and Analysis

The fault diagnosis and fault analysis can be performed off–line using any available quantitative or qualitative tool, or on–line using FIR. To carry out an on–line diagnosis and analysis, FIR must continue with the prediction of system behavior after the structural change has taken place.

To this end, qualitative models that represent faulty subsystems must be swapped for other qualitative models that have been determined under the newly active system structure, i.e., the optimal masks used before the structural change took place must be substituted by new optimal masks that represent the new situation created after the structural change. These new fuzzy optimal masks that fit the new situation include the necessary information for the diagnosis and analysis of the structural change. If the optimal masks used before the structural change are still capable of forecasting the behavior of the system after the perceived change has occurred, i.e., after the error alarm has been triggered, then it can be concluded that no structural change has taken place, or that the change is so small and smooth that it cannot be considered a fault, but represents a small operational disturbance only. In this case, the alarm threshold has probably been set at too low a value.

There are two reasons for the FMS to continue with the prediction of system behavior after the change has been detected; the first one is that the necessary elements to perform the diagnosis and analysis of the structural change are included in the new qualitative model, i.e., the qualitative model itself represents the result of the diagnosis, and with that qualitative model, an analysis process can be carried out. The second reason is that the role of the FMS does not terminate once an anomalous behavior has been detected, but continues into the new post–accident situation.

Since the inductive reasoner is not capable of predicting the behavior of a system that it has never observed before, a successful prediction of the system behavior after the change is only possible if one of the following three assumptions holds:

a) The inductive reasoner switches from the *prediction mode*, in which it has been operating before the structural change occurred, to a **training mode** in which new incoming data should be observed in order to generate a new qualitative model that fits the new situation.

b) All possible structural modes have been previously observed and characterized by FIR in such a way that a qualitative model for each one is available in a **Qualitative Model Library**, avoiding the necessity for a new training phase.

c) All possible structural modes have been previously observed and characterized by FIR in such a way that a qualitative model for each one is used in parallel to **Forecast All Possible Structures**, avoiding the necessity for a new training phase, and for a search in a qualitative model library.

In the next subsections, these three assumptions will be explored in full. However, since the first assumption *Back to Training Mode* presents severe drawbacks in a real–time control environment, it will not be implemented in any practical application.

#### 4.2.5.1 Back to Training Mode

FIR detects that a structural change has taken place since it cannot predict the behavior of the system any longer. Since no other structural mode is known to the reasoner, i.e., no other structural mode has been previously observed and consequently modeled, the inductive reasoner switches from the *prediction mode*, in which it has been operating before the structural change occurred, to a *training mode*, in which it observes new incoming data for a sufficiently long period of time to characterize a new structural mode, i.e., to generate a new set of fuzzy optimal masks. Only after this phase has been successfully completed will the inductive reasoner switch back to its prediction mode and resume its original duties. Figure 4.3 shows a schematic diagram of this assumption including the qualitative simulation, the dynamic comparison, and the identification procedures.

This assumption has severe drawbacks in a real–time control environment. It requires a time period after a structural change has taken place for determining the new qualitative model to be used, and consequently, for performing the diagnosis and analysis of the detected structural change. During this time period, the supervisory control is disabled for all practical purposes. However, it is exactly this time period when the transient takes place, and when knowledge of what is going on would be most valuable to damp out the transition shock and to steer the system smoothly into its new mode of operation.

Consequently, this assumption makes little sense as the fault monitoring strategy of a continuous FMS. It can only be used for building:

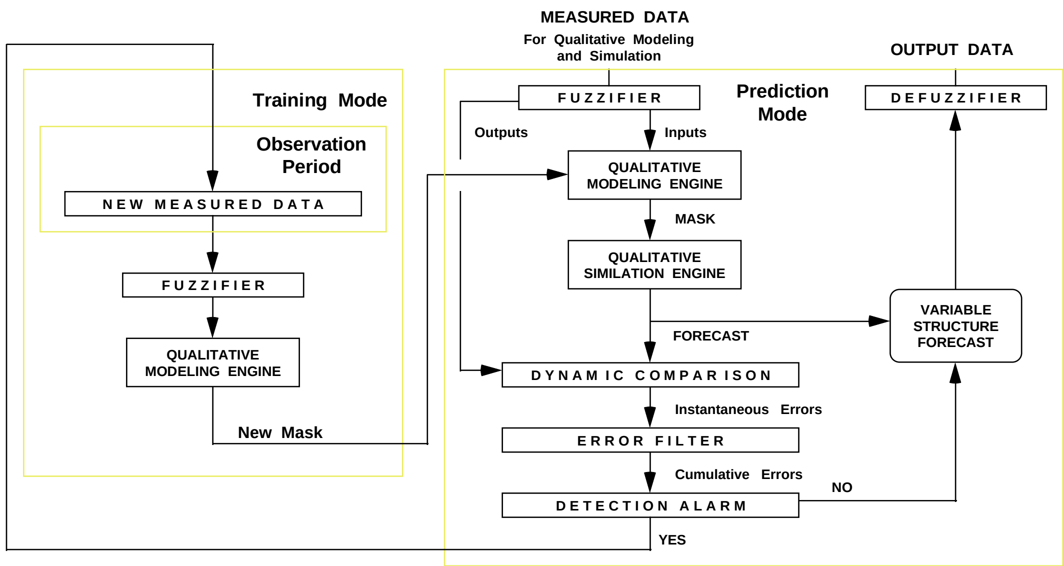


Figure 4.3: “Back to Training Mode” scenario.

- A semi–continuous FMS with functions for early warning. Some potential problems can be discovered before they become emergencies or even accidents.
- A semi–continuous transient discovery system for quick detection of an evolving anomaly capable of pointing out which of the subsystems is causing the problem, but incapable of diagnosing and analyzing the precise nature of the detected anomaly.
- An off–line anomaly characterization and identification system to be used with other tools for post–accident analysis.

#### 4.2.5.2 Qualitative Model Library

In this assumption, the system should have previously been observed by the inductive reasoner in all its structural modes (or at least in the most common ones), and different hierarchies of fuzzy optimal masks, one hierarchy for each mode, should have been stored away in a *Qualitative Model Library* for later reuse. Once a structural change is detected, isolated, and characterized, the model library is searched for another hierarchy of fuzzy optimal masks that leads to qualitative behavior that is consistent with the real system behavior observed after the change.

Each of a set of qualitative models in the library will be tried during some specified time, in order to find the one that best fits the new situation. Each of these qualitative models represents a particular type of structure, and because an abnormal behavior is considered a structural change, this information can be used to conclude what anomaly has happened, i.e., to discriminate between different types of accidents (diagnosis), and to decide upon an appropriate corrective action to be taken (analysis). In this way the continuity of the fault monitoring tasks after a structural change is also guaranteed. Figure 4.4 shows the schematic diagram of this assumption. Notice that the upper modules of the FMS work in the same way they did in the previous *Back to Training Mode* assumption; what really changes is the fault monitoring strategy.

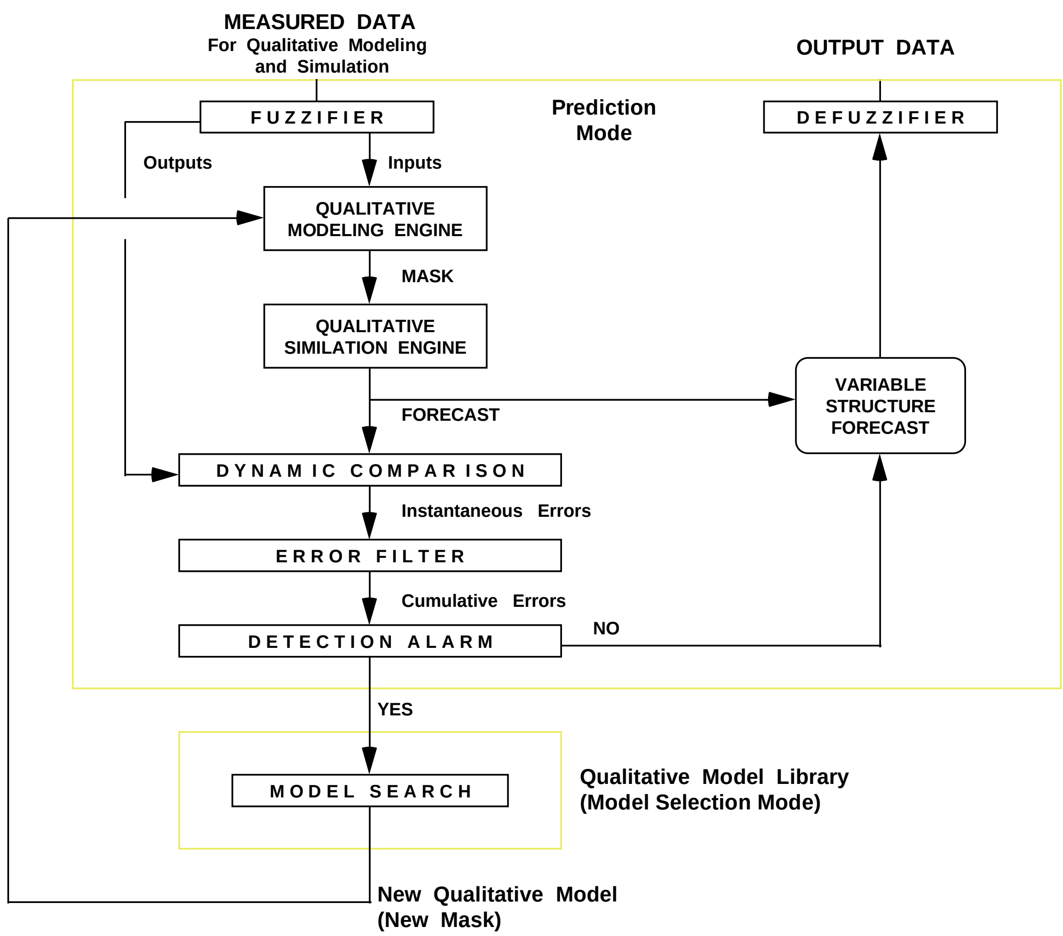


Figure 4.4: "Qualitative Model Library" scenario.

This approach avoids the necessity for the FMS of going back to a training mode during a large period of time, i.e., it avoids the disconnection of the prediction mode during a large period of time. However, a (usually much shorter) disconnection during the library search cannot be avoided. Thus, this option suffers from the same, though less severe, problems as the previous *Back to Training Mode* option. Once a structural change has been detected, isolated, and characterized, the *Qualitative Model Library* must be searched for a new model that is capable of explaining the observed system behavior. While this search is proceeding, behavior forecasting for fault monitoring purposes is disabled. The FMS continues with the prediction of system behavior, but the purpose of this prediction is that of comparing the new reality with each one of the qualitative models in the library, in order to select the one that best fits the new situation.

Although this approach works amazingly well for dynamic systems in which chained faults do not occur, it also fails to predict the system behavior precisely when the fault is taking place, i.e., when such a prediction would be most useful. This approach can be used to build any of the three types of FMSs described in the previous *Back to Training Mode* assumption, or a continuous FMS[4] with functions for:

- Early warning.
- Quick detection of an evolving anomaly with isolation, characterization, diagnosis, and analysis capabilities.
- Correct selection of a new qualitative control model that fits the new situation every time an anomaly occurs.
- Off–line post–accident analysis.

This *Qualitative Model Library* approach was successfully applied to a quantitative model of a Boeing 747 aircraft at cruise flight. The control elements of the aircraft's autopilot were the engine thrust and the elevator deflection angle, whereas the related output variables predicted by the inductive reasoner were the lift, the drag, and the flight path angle. Four different malfunctions or accidents were introduced into the quantitative model in order to alter its normal behavior. The FIR–based FMS was allowed to learn the behavior of the aircraft in its five structural modes, and to build a qualitative model library with the five qualitative models. In a subsequent simulation experiment, malfunctions were triggered to occur randomly in such a way that the qualitative model was kept in the dark with respect to when the accident would take place, and which of the accidents had been selected.

[4]Except for the time needed to find the new qualitative model in the library.

The FMS was capable to detect, characterize, and diagnose the four accidents unambiguously. The purpose of the supervisory control was to provide a *watchdog monitor* for the aircraft's autopilot. The methodology worked fine for the task at hand.

These results were first presented at the *QUARDET'93, Qualitative Reasoning and Decision Technologies Workshop* [de Albornoz and Cellier, 1993a]. An improved version of this article in which the advantages of a Fuzzy Inductive Reasoner over a Crisp Inductive Reasoner were demonstrated was later published in *Simulation* [de Albornoz and Cellier, 1994]. The results of this experiment, and the experiment itself, will be presented at the end of this chapter.

A second application example of this approach was a large–scale quantitative model of a Boiling Water Nuclear Reactor at 100% power. The numerical model used to obtain the measurement data had been built in a previous project [Ramos and de Albornoz, 1988; Ramos, 1991], and had been intended for transient analysis during all phases of plant operation (start–up, steady state, and shut–down). Since the numerical model is quite large, it contains approximately 500 variables and more than 100 differential equations, the purpose here was limited to the selection of a minimum set of meaningful variables to provide the inductive reasoner with, and to build a qualitative model hierarchy of inductive reasoners, i.e., one inductive reasoner for each identified subsystem, rather than building qualitative models of all possible transients. For this reason, only two structural modes were simulated and characterized in the qualitative library.

The selection and causal grouping of variables was made using Optimal Mask Analysis rather than Reconstruction Analysis, because the latter methodology had not yet been implemented at the time when the research was made. This means that the evaluation of the temporal hierarchical array of subsystems was made using some expert knowledge from the modeler rather than the Optimal Structure Analysis of the Reconstruction Analysis methodology. The two structural modes were properly identified and discriminated. In this case, the purpose was limited to providing a human plant operator with additional information that could prove useful when dealing with these two types of developing emergencies. This experiment was also presented at the *QUARDET'93* Workshop [de Albornoz and Cellier, 1993b].

This example will be used once more in Chapter 7 to demonstrate the validity of the combination of Inductive Reasoning and Reconstruction Analysis for building a qualitative Fault Monitoring System to be applied to quantitative large–scale systems.

#### 4.2.5.3 Forecasting all Possible Structures

With this strategy, the mode transition is both detected and discriminated almost immediately. Like in the previously discussed *Qualitative Model Library* option, all possible structural modes must have been previously modeled and characterized. However, instead of placing them in a model library, all qualitative models are used in parallel to constantly predict different qualitative behaviors of the system, i.e., all models corresponding to all structural modes are used in parallel to predict the future behavior of the system. Obviously, only one of the models can represent the true behavior of the system at any one time. This model is identified by continuously comparing all predictions against the real measurement data just like in the aforementioned scenarios [de Albornoz *et al.*, 1994].

Figure 4.5 shows a schematic diagram of the whole process. The measured data are fed in parallel to the different hierarchies of masks representing the different structural modes. Each of them produces a stream of qualitative forecasts. A dynamic comparison is made for all structures simultaneously, leading to several instantaneous error vectors, one for each structural mode. The error filter works as described before generating a moving average of errors accumulated over the most recent $k$ steps. The *Alarm Module* is substituted by a *Mode Selector Module* that bases its decision on the smallest among the filtered errors.

In this way, the FIR–based FMS can switch from one qualitative model to another almost immediately after a structural change has taken place. The determination of a qualitative model capable of representing the behavior of the system at all times is guaranteed, irrespective of the structural mode the real system is in.[5] Thus the diagnosis is done almost immediately which lets the FMS perform an analysis that can influence the control of the current developing anomaly by avoiding the transition shock and steering the system smoothly into its new situation.

This approach avoids the shortcomings of the two previously explained approaches. A continuous FMS using this approach has all their aforementioned capabilities, and the following advantages:

- The prediction of system behavior is always enabled.
- The time used to diagnose the anomaly is reduced to a minimum.

[5] Assuming that all possible structural modes the system can be in are precisely those that the FMS is using in parallel.

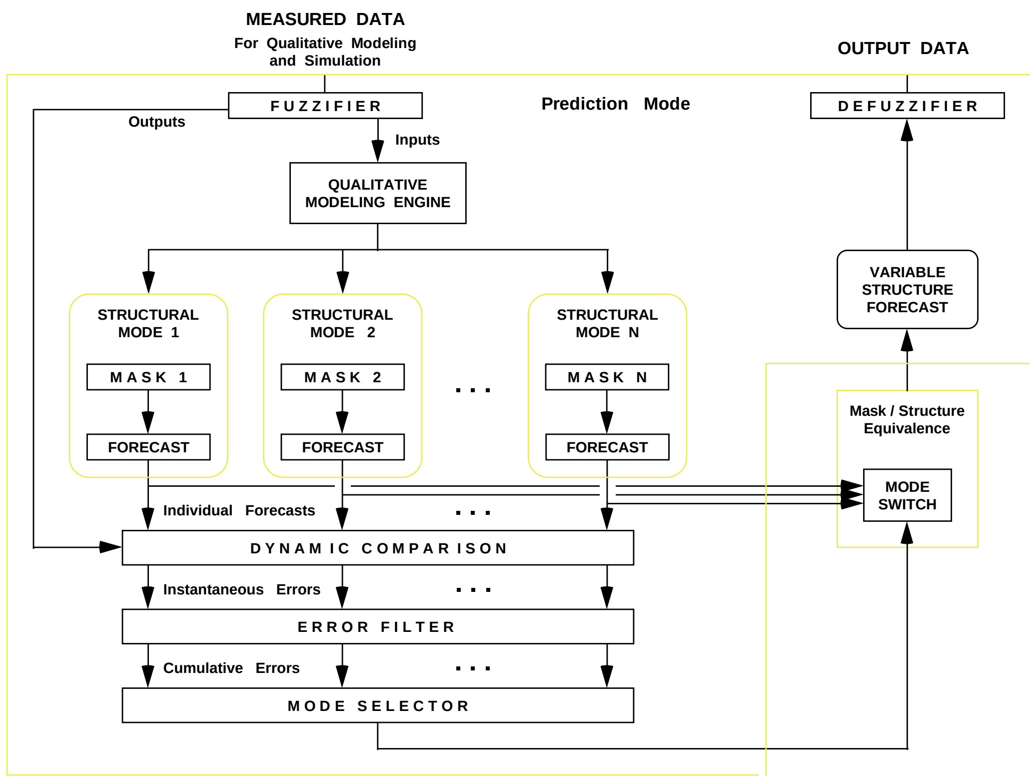


Figure 4.5: "Forecasting all Possible Structures" scenario.

- The FMS captures the behavior of the developing anomaly.
- The analysis performed by the FMS can influence the control of the current developing anomaly.
- This fault monitoring strategy can be applied to dynamic systems with fault chains, and to variable structure systems.

The *Forecasting All Possible Structures* approach was successfully applied to two different variable structure system examples. In the first one, a quantitative model of a two–interconnected–water–tank system was used to obtain the measurement data. The FMS included quantitative models of the four possible structural modes the system can be in, and predicted the behavior of the water level in each of the two tanks using these four quantitative models in parallel. In another simulation, the quantitative model was randomly driven through all structural modes in such a way that the FMS was kept in the dark with respect to when the structural change would take place, and which of the structural changes had been selected. The FMS proved capable of detecting, isolating, characterizing, identifying, diagnosing, and analyzing the four structural changes.

The second application example of this approach was much more involved. It consisted of an electrical circuit with three binary switches and eight possible different structural modes that exhibit behavioral patterns similar enough to make their correct identification a difficult problem. The simulation experiments were carried out in the same way as in the two–interconnected–water–tank example. In this case, the FMS was capable of identifying most of the structural modes the system was in, but not all of them.

These experiments and their results were presented at the *ESM'94 Conference on Qualitative Information, Fuzzy Systems, and Neural Networks in Simulation* [de Albornoz *et al.*, 1994]. An extended version of this paper has been accepted for publication in the *Mathematical Modeling of Systems* journal. Both experiments will be explained in full in the implementation section of this chapter.

## 4.3 Implementation of the Qualitative Model Library Assumption

In this section, a qualitative FMS based on the *Qualitative Model Library* assumption is applied to a quantitative model of a Boeing 747 aircraft at high–altitude horizontal cruise flight [Vesanterä, 1988].

Following the steps explained in the past sections of this chapter, it is now clear that various qualitative models of the airplane must be built, one for each structural mode, in order to be used by the FMS. These qualitative models must be capable of mimicking part of the human situation assessment process by learning how the system behaves, and the FMS must be capable of identifying specific events that can be treated as faults taking place. In other words, the qualitative FMS can be trained to determine when a malfunction occurs in the quantitative model, it can be made to hypothesize about the nature of this malfunction, and may eventually be brought to suggest a global control strategy that would allow to safely operate the quantitative aircraft model under the modified flying conditions.

Such an algorithm (qualitative aircraft model and FMS) could be implemented as a "watchdog autopilot," i.e., as an addition to a conventional autopilot that would allow the autopilot to remain operational after a malfunction has taken place. On a shorter–term basis, such a system could be used by a human pilot as an on–line automated advisor, i.e., a diagnostic aid and a consultation system, that alerts the pilot to perceived problems, and offers advise as to how to deal with them.

This application has two different objectives. The first one is to show that a FIR–based Fault Monitoring System is able to recognize, within a few seconds after a simulated malfunction has taken place, that the aircraft has qualitatively changed its behavior, triggering then a diagnostic engine based on the aforementioned *qualitative model library* approach, all this in order for the FMS to distinguish unambiguously between 5 different types of malfunctions.

The second objective is to demonstrate that, by incorporating fuzzy measures into the inductive reasoning process and by modifying the algorithm for the evaluation of the quality factor of the qualitative (structural) relationships, the discriminatory power of the inductive reasoner is enhanced, i.e., to demonstrate that a FIR–based FMS has an enhanced predictive and discriminatory power over a crisp inductive reasoner–based FMS [Vesanterä, 1988; Vesanterä and Cellier, 1989], by comparing two fault monitoring systems, one using the crisp approach and the other using the fuzzy approach, applied to the same quantitative aircraft model.

In the following sections, a description of the quantitative and qualitative models will be given, emphasizing the differences between the crisp and the fuzzy inductive reasoning approaches, as well as the importance of the chosen application from the point of view of the autopilots.

### 4.3.1 Aircraft Control and Autopilots

Small aircrafts can be manually controlled as shown in figure 4.6. The pilot observes the instruments and the scenery and controls the aircraft as to follow the desired flight path.

Larger aircrafts, such as commercial airliners, cannot be controlled in this fashion. The reason is that the desired managerial flight characteristics (high speed, low fuel consumption) are in conflict with the technical flight characteristics (stability, disturbance suppression). An economic airliner cannot be built with sufficiently benign technical characteristics for it be flown by a human pilot. Such an airliner is by itself an almost unstable system.

Therefore, all airliners include automatic controllers that assist the human pilot in his or her task. The control configuration of a commercial airliner is shown in Figure 4.7. The figure is highly simplified, because the automatic

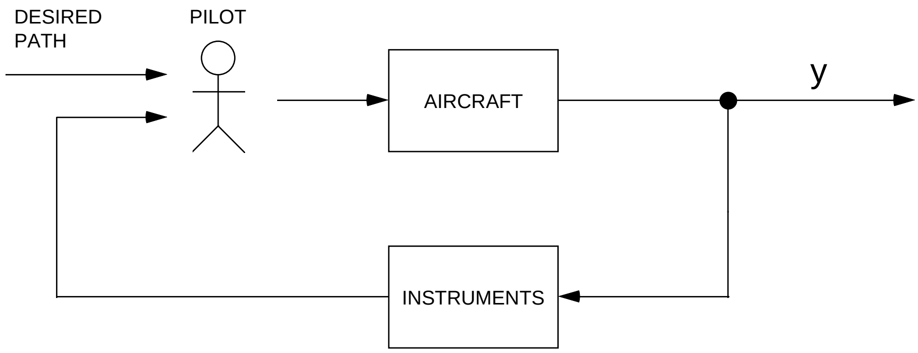


Figure 4.6: Aircraft with human control.

control system is in itself a highly complex system consisting of several controllers for different tasks.

The *Stability Augmentation System (SAS)* artificially enhances the stability of the aircraft. It consists of three separate controllers, namely a *roll damper*, a *pitch damper*, and a *yaw damper*.

A commercial airliner has also considerably more parts to be controlled than a small aircraft. For example, it has flaps on the wings to increase the lift at low speed. Human pilots have difficulties to control all these parts separately. They need automated support in reducing the control functions to a manageable number.

The *Control Augmentation System (CAS)* supplies this functionality. A small number of logical human control functions will be expanded to a considerably larger number of physical automated control functions. The human pilot thus perceives the aircraft as a quite different device from what it really is. However, with the SAS and CAS in place, the enhanced aircraft can now be flown in basically the same manner as a small aircraft would.

Autopilots are designed to replace human pilots during routine operation of the aircraft. Figure 4.8 shows a conventional autopilot. The autopilot simply assumes the place formerly occupied by the human pilot, and the role of the human pilot is reduced to providing the flight plan. Autopilots can be classified in accordance with their functionality. There are two basic types of autopilots:

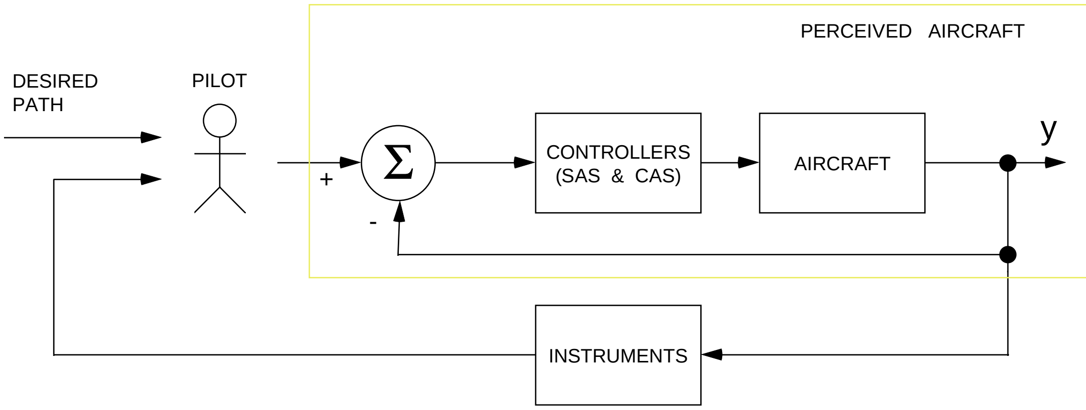


Figure 4.7: Aircraft with SAS and CAS control systems.

a) pilot–relief autopilots, and b) navigation–coupled autopilots [Stevens and Lewis, 1992].

a) The *pilot–relief autopilots* are designed to meet specifications on steady–state error and disturbance rejection, with less emphasis on dynamic response. Their main function is that of relieving human crew from routine tasks such as holding the main flying variables steady, including:

   - pitch angle,
   - altitude, and
   - velocity.

b) The *navigation–coupled autopilots* are designed to assume control of more advanced (but normal) operations such as:

   - navigation and guidance, and
   - automated landing.

Autopilots are rarely equipped to handle abnormal situations. Recently, some navigation–coupled autopilots have been designed to support small foreseen structural changes, as for example, the autopilots with automatic landing capabilities. Modern automatic landing autopilots not only control

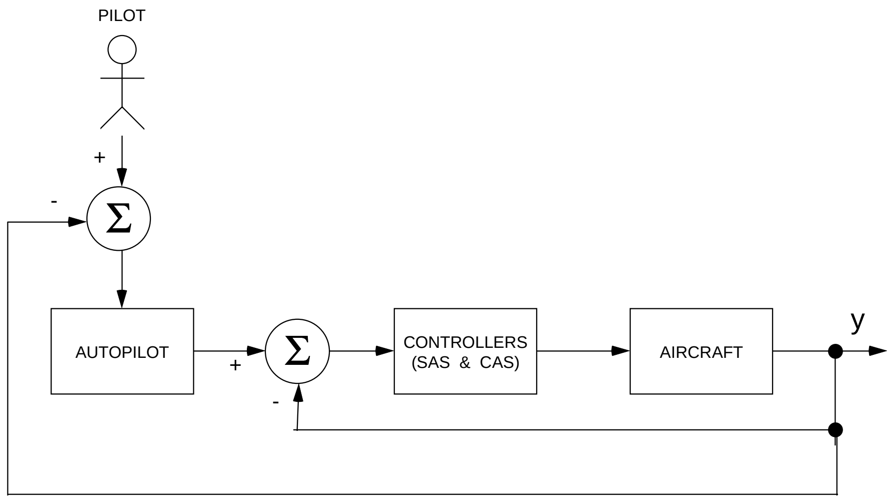


Figure 4.8: Conventional autopilot scheme.

the descending rate and the flight path as they used to, but instead, they are responsible for the entire landing maneuver including control of the structural change of the wings. During takeoff and landing, the aircraft wings are usually "reconfigured" (structurally modified) by deploying flaps and wing leading–edge devices so that the wings effectively have more camber, providing the airplane with more lift at low speed [Air France, 1989].

However, if structural changes are more serious, i.e., the aerodynamic parameters change drastically, as for example when a thin ice plate has formed on the wings, when one or more engines have been shutoff, when the rudder or the elevators are stuck, to give just a few examples, the autopilot is automatically being disengaged, because it is not able to understand what is happening, and consequently, it would try to maintain the previous flight specifications that may no longer be valid under the modified circumstances. This, in turn, requires the human pilot to take full control of the airplane.

By using a "watchdog autopilot," situations such as the ones described above can be aided. A watchdog autopilot is an addition to a conventional autopilot with two possible functions. On the one hand, it can detect that flying conditions are changing in such a way that they could become dangerous, and report its findings immediately to the human crew or to the destination airport, depending on the perceived seriousness of the anomaly. This function is depicted in Figure 4.9. Alternatively, it could try to prevent the modified flying conditions from ever becoming an emergency by informing the autopilot of the problem, such that the autopilot can modify its behavior and remain operational even under the modified flying conditions. The latter kind of watchdog autopilot is shown in Figure 4.10.

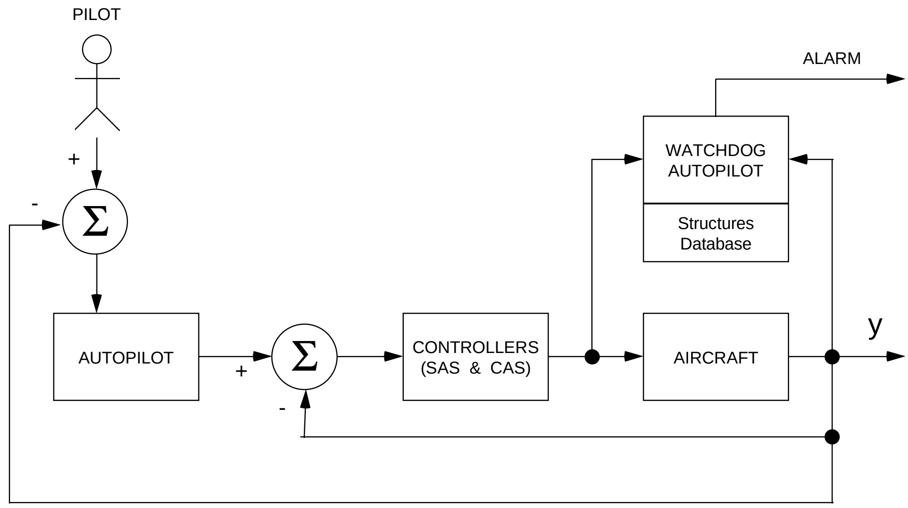


Figure 4.9: Watchdog autopilot with alarm function.

An aircraft watchdog autopilot can be designed by means of a differential equation model of the dynamics that is run in real time driven by the same control actions that are applied to the real aircraft. The differences between the measured aircraft performance and the expected (simulated) aircraft performance can then be used to disable the autopilot and trigger the alarm. However, such a solution offers relatively little in terms of an explanation of the nature of the perceived anomaly. If the mathematical model is not just intended for normal flight simulation, but for encompassing anomalous behavior as well (for the purpose of providing a better explanation of what is going on), the quantitative simulation will become sluggish, and very fast and expensive computers will be needed to perform the required operations in real time. However, if the purpose of the simulation is only that of distinguishing between a number of structurally different anomalous behaviors, all these detailed computations may in fact not be needed. To this end, we propose an alternative approach based on our qualitative Fuzzy Inductive Reasoning methodology to be applied to this problem. The purpose here is the construction of a qualitative watchdog autopilot capable of detecting and identifying sudden changes in the flying conditions by means of the previously explained *Qualitative Model Library* assumption.

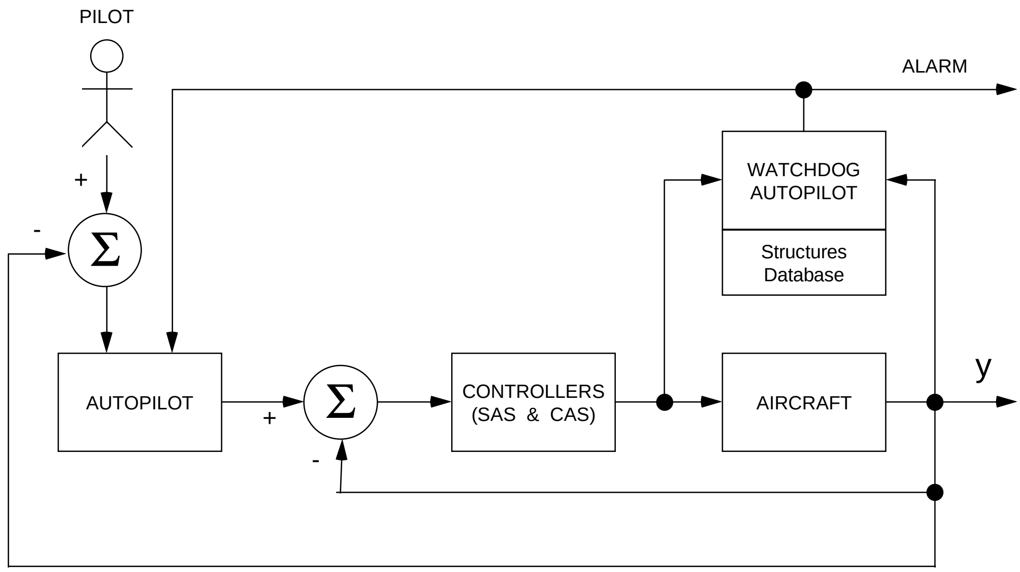


Figure 4.10: Watchdog autopilot with control function.

### 4.3.2 The Quantitative Model

A brief explanation of the mathematical aircraft model is needed to introduce the variables that will be used in the qualitative model. The mathematical aircraft model used in this study reflects an essentially longitudinal flight restricted to longitudinal deviations from a trimmed reference flight condition, which is characterized by the requirement that the resultant forces and moments acting on the aircraft center of mass are zero.

By using the coordinate system shown in Figure 4.11[6], where the origin is placed at the center of gravity of the airplane, the $x$–axis points in the direction of the motion, the $z$–axis points downward, and the $y$–axis runs spanwise and points to the right, the three angles that are needed to describe the relative position of the center of gravity of the airplane, and of the velocity vector with respect to an earth–fixed reference frame and a fuselage–fixed reference frame, can be defined. The first one, $\alpha$, is the angle of attack (or incidence) which is defined using the velocity $x$–axis component, $u$, and the velocity $z$–axis component, $w$. The angle of attack denotes the inclination of the fuselage reference line with respect to the velocity vector, i.e., the tangent to the real flight path line. Hence:

[6] This figure has been taken from [Vesanterä, 1988].

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \tag{4.1}$$

The second is $\gamma$, the flight path angle of the aircraft, representing the inclination of the velocity vector to the horizontal, and the last is $\theta$, the pitch angle, representing the relative position between the two reference frames, that is, between the fuselage reference line and the horizontal line. The pitch angle is defined in terms of the previous angles as:

$$\theta = \gamma + \alpha \tag{4.2}$$

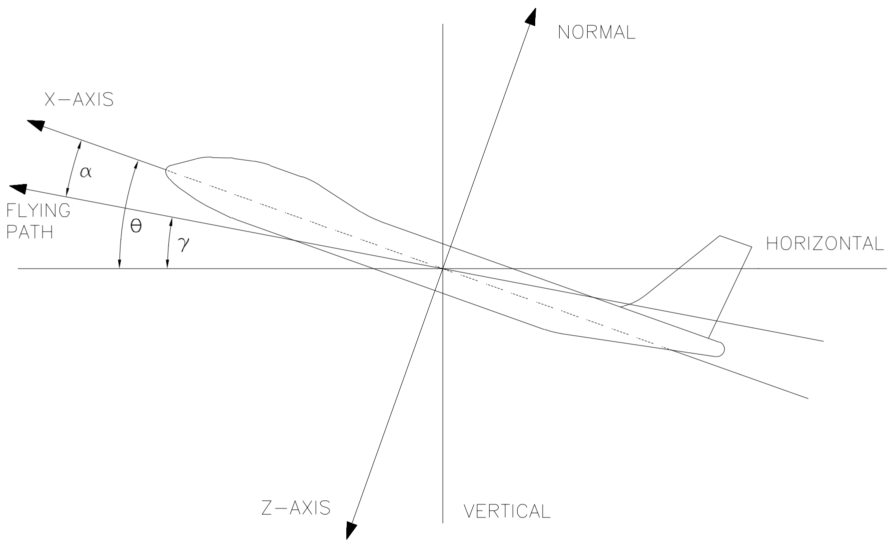


Figure 4.11: Reference angles of the aircraft.

When considering a rigid body on an essentially longitudinal flight, the resultant force can be decomposed into its tangential and normal components, and can be written in terms of the reference angles as:

$$F_t = m\frac{d\mathbf{v}}{dt} \quad ; \quad F_n = m\mathbf{v}\frac{d\gamma}{dt} \tag{4.3}$$

The only active moment about the center of gravity of the airplane considered as a rigid body and under the specified flying conditions will be the pitching moment, i.e., the one about the axis that is perpendicular to the longitudinal symmetry plane called $y$–axis:

$$M_y = I_y \frac{d^2\theta}{dt^2} \tag{4.4}$$

The quantities affecting the airplane in flight, shown in Figure 4.12[7], are its weight $W$, the thrust $T$ developed by the engines, the aerodynamic forces Lift $L$ and Drag $D$, and the aerodynamic pitching moment $M$. The weight of the aircraft is considered constant. The thrust is considered as being a function of the flight velocity and of its own control variable $\delta_T$, the throttle opening. For reasons of simplicity, the thrust line will be assumed to coincide with the $x$–axis. The aerodynamic forces, Lift $L$ and Drag $D$, compose the force response of the aircraft to the motion. The Lift is assumed as being the normal component of the aerodynamic force with respect to the flight path, and the Drag is its tangential component. The aerodynamic pitching moment $M$ about the center of gravity is defined to be positive for a nose–up effect, and negative for a nose–down effect.

The standard way of expressing the aerodynamic forces $L$ and $D$, and the aerodynamic momentum $M$ is through their non–dimensional aerodynamic coefficients $C_D$, $C_L$, and $C_M$.

$$L = \frac{1}{2}\rho v^2 S C_L \tag{4.5}$$

$$D = \frac{1}{2}\rho v^2 S C_D \tag{4.6}$$

$$M = \frac{1}{2}\rho v^2 S \frac{\bar{c}}{2} C_M \tag{4.7}$$

where

[7]This figure has been taken from [Vesanterä, 1988].

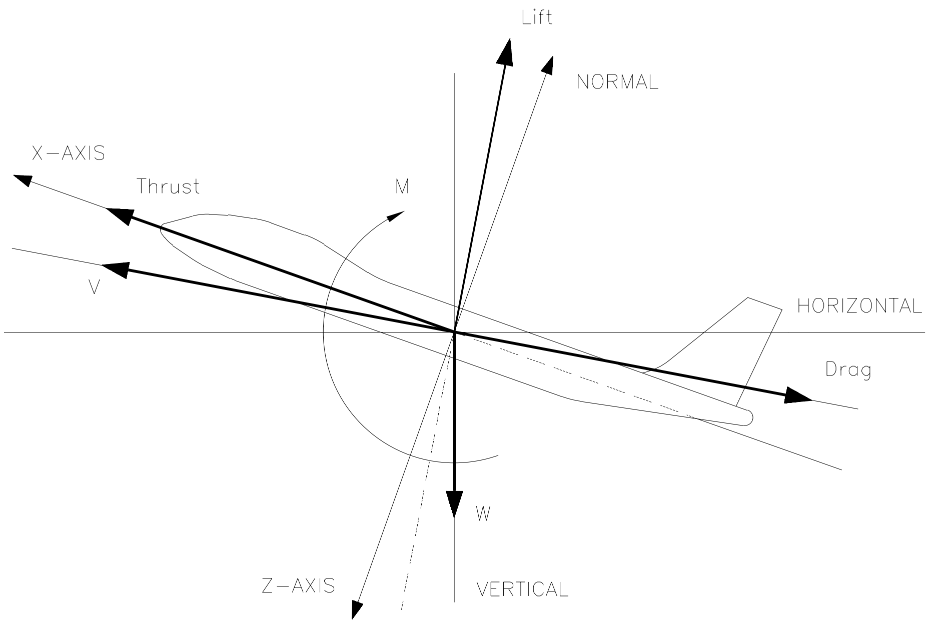


Figure 4.12: Forces and moment acting on the aircraft.

| | | |
|---|---|---|
| $\rho$ | = | is the local air density; |
| $v$ | = | is the cruising velocity; |
| $S$ | = | is the size of the aerodynamic surface of the plane; |
| $\bar{c}$ | = | is the mean aerodynamic chord of the wing. |

These three non–dimensional coefficients express the aerodynamic response of the airplane to variations in the angle of attack $\alpha$, the elevator deflection $\delta_e$, the angle of attack rate $\dot{\alpha}$, and the pitch rate $q$. By expanding these non–dimensional coefficients into a Taylor series around an initial value, they can be expressed in terms of $\alpha$, $\delta_e$, $\dot{\alpha}$, and $q$ as:

$$C_L = C_{L_0} + \frac{\partial C_L}{\partial \alpha}\alpha + \frac{\partial C_L}{\partial \delta_e}\delta_e + \frac{\partial C_L}{\partial \dot{\alpha}}\dot{\alpha} + \frac{\partial C_L}{\partial q}q \tag{4.8}$$

$$C_D = C_{D_0} + \frac{\partial C_D}{\partial \alpha}\alpha \tag{4.9}$$

$$C_M = C_{M_0} + \frac{\partial C_M}{\partial \alpha}\alpha + \frac{\partial C_M}{\partial \delta_e}\delta_e + \frac{\partial C_M}{\partial \dot{\alpha}}\dot{\alpha} + \frac{\partial C_M}{\partial q}q \tag{4.10}$$

The aerodynamic reactions of the airplane can be represented approximately by means of stability derivatives, i.e., the coefficients of the Taylor series expansion. The relations between $\alpha$, $\delta_e$, $\dot{\alpha}$, and $q$ are described by these stability derivatives in the following way:

**$\alpha$ coefficients $C_{L_\alpha}$, $C_{D_\alpha}$, and $C_{M_\alpha}$.** They describe how changes in the angle of attack $\alpha$ affect the aerodynamic forces and moments. An increase in the angle of attack generally induces an increase in the Lift, an increase in the Drag and a negative pitching moment.

**$\delta_e$ coefficients $C_{L_{\delta_e}}$ and $C_{M_{\delta_e}}$.** They describe the effect that a deflection of the elevator has on the Lift and on the Pitching Moment. A positive elevator deflection is defined as being *elevator down*, which causes an increase in the Lift and a negative pitching moment increment.

**$\dot{\alpha}$ coefficients $C_{L_{\dot{\alpha}}}$ and $C_{M_{\dot{\alpha}}}$.** Basically, they represent the adjustment of the pressure distribution on the aerodynamic surfaces to sudden changes in the angle of attack.

**$q$ coefficients $C_{L_q}$ and $C_{M_q}$.** They represent the aerodynamic effects induced by a rotation of the airplane about its spanwise axis when the angle of attack is kept constant, e.g., keeping the fuselage tangential to an arbitrarily varying flight path.

The equations in which the non–dimensional aerodynamic coefficients are expressed in terms of the Taylor series expansion stability derivatives are, for the Lift coefficient:

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_{\delta_e}}\delta_e + \frac{\bar{c}/2}{v}\left[C_{L_{\dot{\alpha}}}\dot{\alpha} + C_{L_q}q\right] \tag{4.11}$$

for the Drag coefficient:

$$C_D = C_{D_0} + C_{D_\alpha}\alpha \tag{4.12}$$

and for the aerodynamic momentum coefficient:

$$C_M = C_{M_0} + C_{M_\alpha}\alpha + C_{M_{\delta_e}}\delta_e + \frac{\bar{c}/2}{v}\left[C_{M_{\dot{\alpha}}}\dot{\alpha} + C_{M_q}q\right] \tag{4.13}$$

In order to simplify the analysis, the equations of motion will not be stated in the stability axes but in the tangential and normal axes with respect to the flight path, and they are [Stevens and Lewis, 1992]:

$$m\dot{v} = T\ \cos\alpha - D - W\ \sin\gamma \tag{4.14}$$

$$mv\dot{\gamma} = T\ \sin\alpha + L - W\ \cos\gamma \tag{4.15}$$

The pitching moment is given by:

$$I_y\dot{q} = M \tag{4.16}$$

The pitch angle rate is:

$$\dot{\theta} = q \tag{4.17}$$

The position of the airplane with respect to the ground is given by:

$$\dot{h} = v\ \sin\gamma \tag{4.18}$$

$$\dot{x} = v\ \cos\gamma \tag{4.19}$$

Two control laws are implemented in the model for stability and flight control, one for the elevator deflection with feedback on the pitch angle, the other for the thrust with feedback on the velocity:

$$\delta_e = \delta_{e_{\mathrm{trim}}} + K_\theta(\theta - \theta_{\mathrm{trim}}) \tag{4.20}$$

$$T = T_{\text{trim}} + K_u(u - u_{\text{trim}}) \tag{4.21}$$

The subscript *trim* refers to the trimmed value of the variables, $K_\theta$ and $K_u$ are the feedback gains, and $u$ refers to the $x$–component of the velocity vector:

$$u = v \ \cos\alpha \tag{4.22}$$

The parameters used in the model are those of a Boeing 747 aircraft in a horizontal cruise flight.

The numerical aircraft model has been coded in the continuous system simulation language ACSL. Some transients/accidents were built into the aircraft model in order to alter its normal structural behavior. These structural changes in the longitudinal flight are simulated by modifying the original airplane parameters in a discrete event section of the program that is executed at a scheduled time instant. Once the event has been activated, another random number is drawn to determine which of the accidents is supposed to occur. The simulated emergencies are not necessarily realistic in terms of what might happen to the real aircraft, but this is not essential to our goal. It is, however, important to realize that the qualitative model is kept in the dark with respect to when the accident takes place, and which of the accidents has been selected.

### 4.3.3 The Qualitative Model

The practicality of combining qualitative and quantitative simulation models of continuous–time processes using fuzzy inductive reasoning techniques has already been demonstrated in Chapter 3. Also, the construction of the qualitative model itself has been described in detail in the same chapter. In this section, only a brief description will be given in order to show how the qualitative model was built.

Two qualitative models of the aircraft will be created using crisp inductive reasoning and fuzzy inductive reasoning. Both are intended for constructing a fault monitoring system (one with each model). The goal is to make a comparison of the following items:

- The number of errors in the qualitatively predicted states of both approaches.
- The length of the error chains produced by those errors.
- The simulated time span needed to identify the same fault.
- The difficulties to discriminate between different types of malfunctions that make the aircraft react in similar ways.

Thus, in accordance with what has been said in Chapter 3, the process of building the qualitative model should include:

i) Excitation of the system by means of random binary noise and/or harmonic functions.

ii) Selection of input and output variables.

iii) Fuzzy recoding of the qualitative variables.

iv) Selection of a minimum set of meaningful variables that represent the system by means of Optimal Mask Analysis.

#### 4.3.3.1 Excitation of the Aircraft

Since the aircraft model is intended for longitudinal flight, the variables that should be used to excite the model are the elevator deflection angle $e$ and the engines thrust $T$. Thus, the model input variables are perturbations affecting the reference values of the two control variables of the model, Equations (4.20 and 4.21), $e_{\text{trim}}$ and $T_{\text{trim}}$. The trimmed values of these variables are preset such that the simulation always starts out in a perfectly stable horizontal flight. A change in any of these two variables will perturb the model, forcing it into a new steady state or into a transient condition.

The ACSL simulation language offers the user the possibility of computing the eigenfrequencies of a non–linear system by means of its linearized model. In particular, the smallest and largest eigenfrequencies $w_{\text{low}}$ and $w_{\text{high}}$ respectively. In this way, the largest time constant, $t_{\text{settling}}$, and the shortest time constant $t_{\text{fast}}$ of the system can be computed in accordance with Equation (3.12). The results are:

$$t_{\text{settling}} = 12\text{sec.} \quad ; \quad t_{\text{fast}} = 2\text{sec.} \tag{4.23}$$

Thus, the optimal masks should cover 12 sec., and due to Equation (3.13), the sampling interval should be set to 1 sec. In accordance with Equation (3.14), the mask depth should hence be chosen as 13. However, with these values, FIR proved incapable of finding a qualitative model to represent the aircraft. Simulation experience[8] tells us that the fastest time constant created instabilities in the model, so it was decided that only the slow behavior of the aircraft should be captured by the qualitative model, and consequently, the sampling interval should be set to 6, and the mask depth should be set to 3.

The numerical model is excited with random ternary noise. In this way, the recorded data are rich in information about the reaction of the system to input stimulation at all frequencies of interest. This mode of operating the aircraft is called the "shaken flight." The values of the two control variables may vary once every 6 sec. They assume at random one of three values including the trimmed values themselves, i.e., $\delta_{e_{\text{trim}}} = 33005.5$ lb. and $T_{\text{trim}} = 0.027886$ rad. $\approx 1.6$ degrees, as well as positive and negative deviations from these trimmed values with $\Delta\delta_{e_{\text{trim}}} = 0.001$ rad. $\approx 0.5$ degrees and $\Delta T_{\text{trim}} = 3000$ lb. Using so small deviations in the elevator and so large variations in the thrust may not be realistic. Yet, these values were chosen since the model is very sensitive to changes in the elevator deflection and much less sensitive to changes in the thrust. The results of this simulation are stored in the *shaken flight raw data matrix.*

The numerical model is excited once more, this time with harmonic functions of fairly long periods. This way of operating the aircraft is called "normal flight with harmonic perturbations." In this way, a more realistic but still dynamic flight simulation results. This step is used to determine the limits that the variables can realistically assume. The results of this simulation are stored in the *normal flight raw data matrix.*

The data extracted from the numerical ACSL simulations constitute the measurement data (the raw data matrices) of the qualitative model. The execution of the quantitative model and the extraction of the raw data matrices were done under the control of Matlab. As has been explained before, the raw data matrices are real–valued matrices in which each column represents one recorded variable, and each row represents one complete data record collected

[8]The FIR methodology still has some aspects that are heuristically solved by means of experience. Chapter 8 will provide a list of these open problems and their possible solutions.

at one time instant. These data are then recoded to enable the qualitative reasoning process using both the crisp and the fuzzy recoding processes.[9] The crisp recoding process has not been explained before. As in the fuzzy case, it maps continuous behavior into discrete qualitative states, but without adding any measure of confidence of that state. Remember that, in the fuzzy case, a fuzzy membership value is appended to the recoded (class) value as shown in Chapter 3.

#### 4.3.3.2 Variable Selection

To build the qualitative model of the aircraft, not all of the variables that were obtained by the numerical simulation were used, but only a subset of five variables that capture the main characteristics of the aircraft during high altitude longitudinal cruise flight. This subset of variables could be obtained by previous expertise of the modeler, or by means of Optimal Mask Analysis.[10]

In many applications, the true system inputs are predetermined. In the given example, these are the elevator deflection and the thrust, i.e., the two control variables that can be influenced by the pilot. Often, also the system outputs are known. They are those variables that the qualitative model is supposed to predict. However, it can happen that the "distance" between the true inputs and the true outputs of the system is too large, i.e., there doesn't exist a sufficiently strong correlation between the input and output variables to support the generation of a reliable and accurate qualitative model. In this case, the optimal mask analysis will lead to a model with poor prediction capability.

In such a situation, some additional variables must be chosen that are located logically somewhere between the true inputs and the true outputs. An optimal mask is then constructed that relates current and past values of the true system inputs and the auxiliary variables, as well as past values of the true outputs to the current values of the true outputs. Then additional optimal masks are found that predict the auxiliary variables from the inputs. In this way, a chain of optimal masks can be obtained if necessary. More about this process shall be presented in Chapter 6 of this thesis.

[9]Crisp and fuzzy recoding processes are used because one of the objectives of this chapter is the comparison between Crisp Inductive Reasoning and Fuzzy Inductive Reasoning.

[10]Remember that the Optimal Mask Analysis is used with two different purposes: on the one hand, for the selection of a minimum set of meaningful variables, and on the other hand, for the determination of the qualitative model that represents the causal relationships between the selected variables.

In the given example, The Optimal Mask Analysis did not need cascaded qualitative models. The distance between the system inputs and the possible system outputs was not too large to be bridged by a single qualitative model. However, it was important to select output variables that were sufficiently different from each other, covering the dynamics of the aircraft sufficiently well, in order to allow a clear and unambiguous identification of the different "accidents" that were considered in this study.

Since it was determined, by previous experience with aircraft models, that the lift, drag, and flight path angle together capture the dynamics of an airplane sufficiently well,[11] the Optimal Mask Analysis methodology was used to obtain a qualitative model of the aircraft, but not to perform an exhaustive search of the most significant variables.

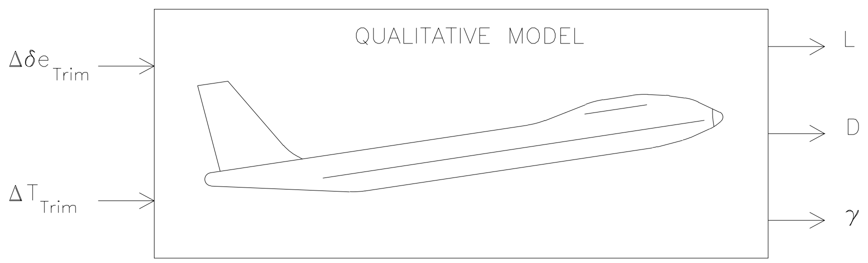


Figure 4.13: Qualitative input and output variables.

Figure 4.13 depicts the two inputs and three outputs determined in an *ad hoc* manner using Optimal Mask Analysis. They are:

| | | |
|---|---|---|
| $\delta_e$ | = | the differential elevator deflection; |
| $\delta_T$ | = | the differential thrust; |
| $L$ | = | the lift; |
| $D$ | = | the drag; |
| $\gamma$ | = | and the flight path angle. |

[11] This choice is by no means unique, and depends on the objectives of the model.

#### 4.3.3.3 Optimal Masks

The inductive reasoner operates exclusively on the qualitative class values and reasons about qualitative spatial and temporal relationships among the aircraft variables without proposing a single quantitative relationship and, in fact, without even knowing that they exist. Following the steps described in Chapter 3 to find the optimal masks, it should be decided into how many qualitative levels the variables should be recoded. It has already been mentioned than an odd number of levels is preferred because a central level is desired where one state of the variables can be considered "normal." In this case, the minimum possible number is 3. Finally, expertise has demonstrated that a complexity between 4 or 5 works fine with five variables. Thus, FIR needs, in accordance with Equation (3.2) at least 1215 data points to identify optimal masks with the following characteristics:

$$
\begin{array}{lcl}
\text{number of variables} & = & 5; \\
\text{number of qualitative levels} & = & 3; \\
\text{maximum mask complexity} & = & 5; \\
\text{depth of the masks} & = & 3;
\end{array}
$$

that have been obtained following Equations (3.12), (3.13), and (3.14). 2500 data points from the *shaken flight raw data matrix*, known as B4 model, and 2500 data points from the *normal flight raw data matrix* will be recoded in order to have enough data to feed into the FMS. The mask candidate matrices were built following Equation (3.15). The resulting optimal masks for the three output variables in the crisp case, using the *shaken flight raw data matrix*, are for the $lift$:

$$
B4_{L\ crisp} \quad = \quad \begin{array}{c} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array}
\begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\
\begin{pmatrix} -1 & 0 & 0 & 0 & -2 \\ -3 & -4 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 \end{pmatrix} \end{array}
$$

for the $Drag$:

$$
B4_{D\ crisp} \quad = \quad \begin{array}{c} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array}
\begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\
\begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ -2 & -3 & 0 & -4 & 0 \\ 0 & 0 & 0 & +1 & 0 \end{pmatrix} \end{array}
$$

and for the flight path angle:

$$B4_{\gamma\ crisp} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & +1 \end{pmatrix} \end{array}$$

Notice that the optimal mask found for the flight path angle $\gamma$ is a mask of complexity 3.[12] The resulting optimal masks for the three output variables in the fuzzy case, using the *shaken flight raw data matrix*, are for the $lift$:

$$B4_{L\ fuzzy} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & -2 \\ -3 & -4 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 \end{pmatrix} \end{array}$$

for the $Drag$:

$$B4_{D\ fuzzy} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ -2 & -3 & 0 & -4 & 0 \\ 0 & 0 & 0 & +1 & 0 \end{pmatrix} \end{array}$$

and for flight path angle:

$$B4_{\gamma\ fuzzy} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & -3 & -4 \\ 0 & 0 & 0 & 0 & +1 \end{pmatrix} \end{array}$$

Notice that the optimal masks for the *Lift* and the *Drag* are the same in the crisp and in the fuzzy cases. However, in the fuzzy case, the optimal mask found for the flight path angle $\gamma$ is one with maximum complexity.

[12] The crisp optimal mask for model B4 and for all subsequent models have been successfully compared with those provided in [Vesanterä, 1988].

The optimal masks obtained from the *shaken flight raw data matrix* and the landmarks obtained from the *normal flight raw data matrix* are used to forecast the future *normal flight* behavior of the airplane in the crisp and fuzzy cases.[13] The resulting qualitative variables will then be *regenerated* in order to make a comparison between them and the real values obtained from the quantitative ACSL aircraft model.

There exist three main differences between crisp inductive reasoning and fuzzy inductive reasoning. The first lies in the utilization of the available fuzzy measures in the reasoning process, the second lies in the computation of the quality measure of the masks, and the third (and possibly most important) one relates to quantitative predictions made.

First, crisp inductive reasoning operates on crisp landmarks in the recoding of the measurement data, i.e., does not employ any measure of confidence. These rigid landmarks are responsible for a loss of valuable information about the system that can no longer be exploited by the qualitative model, and this, in turn, leads to a reduction in its forecasting capabilities, which diminishes the discriminatory power in comparison with the FIR–based FMS.

The second difference lies in the computation of the quality measure of the masks. Crisp inductive reasoning uses the frequency of previous observations of an input/output state to estimate the likelihood of future observations of the same state, whereas FIR uses the information contained in the fuzzy membership functions as a measure of confidence in a particular input/output state.

Third, the fuzzy membership information available to FIR can be used to interpolate between qualitative (class value) predictions, enabling FIR to make *quantitative predictions* by use of its *regenerate* module. The crisp inductive reasoner does not have this information available, and therefore, can only make *qualitative predictions*.

Figure 4.14 compares the qualitative simulation results (class values only) obtained using crisp and fuzzy forecasting. Notice in the error matrix that the crisp methodology produces a reiterative error on the Drag $D$ variable that the fuzzy methodology avoids. Also, the chained errors produced with the crisp approach are avoided with the fuzzy approach. Both simulations correspond to the same aircraft model excited with harmonic functions of long periods.

These qualitative signals can then, in the fuzzy case only, be regenerated using FIRs *regenerate* module. Figures 4.15, 4.16, and 4.17 show a

[13] The justification for combining optimal masks and landmarks steming from different excitations of the aircraft model was given in Section 1.3.3.1 "Excitation of the Aircraft."

Crisp Inductive Reasoning

| | Real Values | | | Forecast Values | | | Error Matrix | | |
|---|---|---|---|---|---|---|---|---|---|
| $_{step}\backslash^{var}$ | $L$ | $D$ | $\gamma$ | $L$ | $D$ | $\gamma$ | $error(L)$ | $error(D)$ | $error(\gamma)$ |
| 2501 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| 2502 | 2 | 3 | 3 | 2 | 3 | 3 | 0 | 0 | 0 |
| 2503 | 2 | 2 | 3 | 2 | 3 | 3 | 0 | 1 | 0 |
| 2504 | 1 | 1 | 3 | 1 | 1 | 3 | 0 | 0 | 0 |
| 2505 | 1 | 1 | 3 | 1 | 1 | 3 | 0 | 0 | 0 |
| 2506 | 1 | 1 | 2 | 1 | 1 | 2 | 0 | 0 | 0 |
| 2507 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 2508 | 2 | 1 | 1 | 2 | 1 | 1 | 0 | 0 | 0 |
| 2509 | 2 | 2 | 1 | 2 | 1 | 1 | 0 | $-1$ | 0 |
| 2510 | 1 | 2 | 1 | 1 | 3 | 1 | 0 | 1 | 0 |
| 2511 | 1 | 1 | 1 | 1 | 3 | 1 | 0 | 2 | 0 |
| 2512 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 2513 | 1 | 2 | 1 | 2 | 2 | 1 | $-1$ | 0 | 0 |
| 2514 | 1 | 2 | 2 | 2 | 2 | 2 | $-1$ | 0 | 0 |
| 2515 | 1 | 2 | 2 | 1 | 2 | 2 | 0 | 0 | 0 |

Fuzzy Inductive Reasoning

| | Real Values | | | Forecast Values | | | Error Matrix | | |
|---|---|---|---|---|---|---|---|---|---|
| $_{step}\backslash^{var}$ | $L$ | $D$ | $\gamma$ | $L$ | $D$ | $\gamma$ | $error(L)$ | $error(D)$ | $error(\gamma)$ |
| 2501 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| 2502 | 2 | 3 | 3 | 2 | 3 | 3 | 0 | 0 | 0 |
| 2503 | 2 | 2 | 3 | 2 | 2 | 3 | 0 | 0 | 0 |
| 2504 | 1 | 1 | 3 | 1 | 1 | 3 | 0 | 0 | 0 |
| 2505 | 1 | 1 | 3 | 1 | 1 | 3 | 0 | 0 | 0 |
| 2506 | 1 | 1 | 2 | 1 | 1 | 2 | 0 | 0 | 0 |
| 2507 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 2508 | 2 | 1 | 1 | 2 | 1 | 1 | 0 | 0 | 0 |
| 2509 | 2 | 2 | 1 | 2 | 2 | 1 | 0 | 0 | 0 |
| 2510 | 1 | 2 | 1 | 1 | 2 | 1 | 0 | 0 | 0 |
| 2511 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 2512 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 2513 | 1 | 2 | 1 | 2 | 2 | 1 | 1 | 0 | 0 |
| 2514 | 1 | 2 | 2 | 1 | 2 | 2 | 0 | 0 | 0 |
| 2515 | 1 | 2 | 2 | 1 | 2 | 2 | 0 | 0 | 0 |

Figure 4.14: Differences in the error matrices between the crisp and the fuzzy FMS.

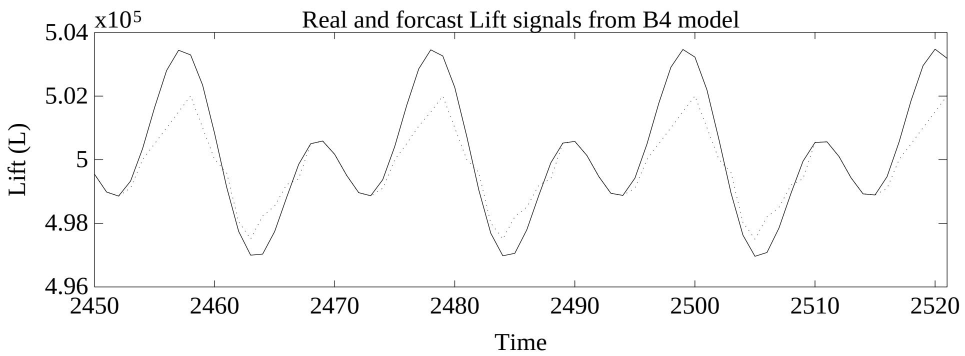


Figure 4.15: Real *vs.* forecast (then regenerated) *Lift* signals.

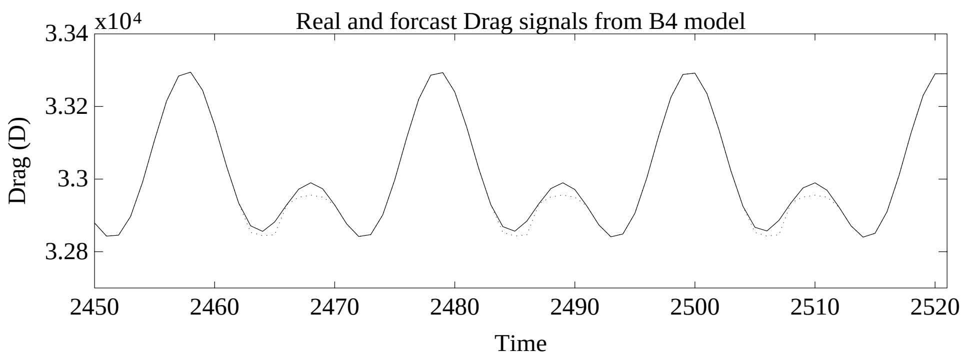


Figure 4.16: Real *vs.* forecast (then regenerated) *Drag* signals.

comparison between the quantitative model (continuous line) and the qualitative (regenerated) FIR model (dashed line) of the output variables *Lift*, *Drag*, and $\gamma$, respectively.

These graphics represent the quantitative and qualitative behavior of the three output variables, corresponding to the same time points of a "normal flight with harmonic perturbations."[14] Notice the extraordinary performance of the qualitative model that accurately follows its quantitative counterpart, especially in the cases of the Drag and the flight path angle depicted in Figures 4.17 and 4.16, where the differences between the two curves (real and forecast signals) are almost indistinguishable. This accuracy of the qualitative models will prove essential for the detection and identification of the aircraft structural changes.

[14]cf. Section 4.3.3.1 for the description of a normal flight with harmonic perturbations.

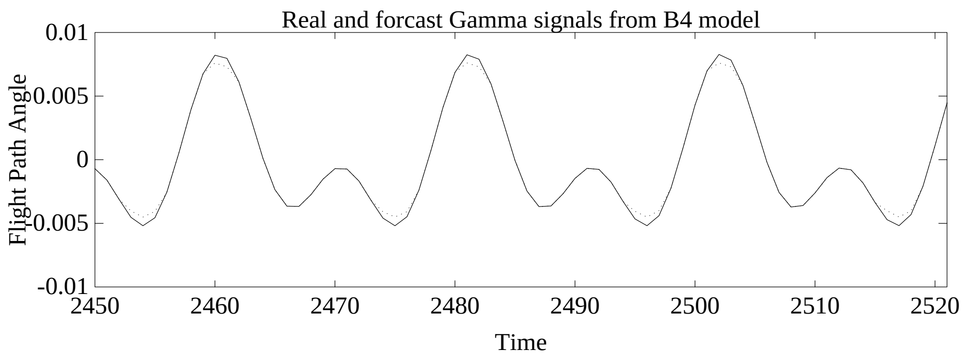


Figure 4.17: Real *vs.* forecast (then regenerated) $\gamma$ signals.

#### 4.3.3.4 Qualitative Structural Changes

The original aerodynamic parameters of the Boeing 747 airplane at cruise flight (B4 model) were modified to obtain four different models with which a library was constructed. These models represent structural changes of the original plane, and were thought to be sufficiently representative to be considered as "accidents," in spite of them not being realistic emergencies. The four models were obtained in the same way as the original one, following the sequence of steps presented at the beginning of this chapter in Section 4.2. The main characteristics of these models are:

**Model B4** is the original model that represents a Boeing 747 in cruise flight at high altitude. Its aerodynamic parameters are considered as reference values for the other models. This is the model that has been built along this section up to this point.

**Model B747** represents a Boeing 747–300 in cruise flight. Basically, this is a bigger aircraft than the one represented by model B4. Thus, the values for the weight and wing span are considerable larger, and the aerodynamic parameters Lift, $L$, Drag, $D$, aerodynamic momentum, $M$, and pitch angle, $\theta$, have also been changed in comparison with the B4 model. The reference value of the elevator deflection has also suffered a drastic change (including a change of sign), while the reference value of the thrust has remained unchanged. However, the thrust itself has dropped to about two thirds of its original value due to a step decrease in the velocity (characteristic of the B747 model).

It clearly never happens that a B747 airplane miraculously changes to become a B747–300 aircraft during cruise flight. In this sense, it is quite clear that the experiment lacks realism. However, icing of the wings, or a sudden deployment of the landing flaps, or a sudden opening of the hatch of the cargo compartment can have equally drastic effects on the flight characteristics of the aircraft. In lack of decent data for realistic accidents, a set of artificial structural changes of the aircraft had to be concocted.

The quantitative data is obtained in an analogous manner as in the case of the B4 model, with the difference that the flight is started with the original model B4 and changed to the B747 model during the tenth communication interval. Once the model is in stable condition (steady state), the excitation phase can be started in order to obtain rich data for building up the qualitative model. This procedure is necessary because every flight has to be started trimmed to avoid the initial iterative trimming phase of the model, which would change the initial conditions that were carefully chosen for the B4 model.

The qualitative B747 model is constructed in the same way as the qualitative B4 model. The sampling interval for the B747 model was evaluated based on its two time constants: $t_{\text{fast}} = 2$ sec, and $t_{\text{slow}} = 22$ sec. Following Equations (3.12) and (3.13), the sampling interval is $\delta t = 11$ sec. The number of qualitative classes, and the depth and complexity of the masks are the same as in the case of the B4 model. The optimal masks obtained for the Lift $L$, Drag $D$, and flight path angle $\gamma$ using crisp inductive reasoning are:

$$B747_{L\ crisp} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -2 & -3 \\ 0 & 0 & +1 & 0 & 0 \end{pmatrix} \end{array}$$

$$B747_{D\ crisp} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & 0 & -3 & -4 \\ 0 & 0 & 0 & +1 & 0 \end{pmatrix} \end{array}$$

$$B747_{\gamma\ crisp} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ -2 & -3 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 \end{pmatrix} \end{array}$$

The optimal masks for the same qualitative variables using fuzzy inductive reasoning are:

$$B747_{L\ fuzzy} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -2 & -3 \\ -4 & 0 & +1 & 0 & 0 \end{pmatrix} \end{array}$$

$$B747_{D\ fuzzy} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & 0 & -3 & -4 \\ 0 & 0 & 0 & +1 & 0 \end{pmatrix} \end{array}$$

$$B747_{\gamma\ fuzzy} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ -2 & -3 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 \end{pmatrix} \end{array}$$

It can be noticed that the optimal masks in the two cases are almost the same. Only the masks for the *Lift* variable are slightly different. However, this distinction is not essential. Both masks would work with either methodology. The additional discriminatory power of FIR stems from the *experience data base* used by the FIR methodology, and not from a refined selection of a more appropriate optimal mask.

**Model B5** represents a change of the original B4 model, in which the aerodynamic parameters $L$ and $D$ are increased, whereas $M$ and $\theta$ are decreased, and the remainder of the B4 model parameters are kept unchanged. This completely modifies the influence that the angle of attack has on the aerodynamic response of the aircraft, particularly on the *Drag* and pitching moment. The sampling interval for the B5 model is $\delta t = 19$ sec. The number of qualitative classes, and the depth and complexity of the masks are the same as in the previous models. The optimal masks obtained for the lift, $L$, the drag, $D$, and the flight path angle, $\gamma$, using fuzzy inductive reasoning are:[15]

[15]for this and the next structural changes, only the fuzzy case will be treated because the comparison between crisp and fuzzy inductive reasoning can be made with the previously described models B4 and B747.

$$B5_{L\ fuzzy} = \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & -2 & 0 & -3 & -4 \\ 0 & 0 & +1 & 0 & 0 \end{pmatrix} \end{array}$$

$$B5_{D\ fuzzy} = \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & -2 \\ -3 & -4 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 \end{pmatrix} \end{array}$$

$$B5_{\gamma\ fuzzy} = \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & 0 & -3 & 0 \\ 0 & 0 & 0 & 0 & +1 \end{pmatrix} \end{array}$$

**Model B13** represents another change of the B4 model. Here, $L$, $D$, and $\theta$ are increased, whereas $M$ is decreased. The most significant changes were made in the effect of the angle of attack on the *Drag*, which has now almost tripled from its original value, and in the effect of the elevator deflection on the aerodynamic moment, which makes this model more sensitive to this control variable. The sampling interval for the B13 model is $\delta t = 15$ sec. The number of qualitative classes, and the depth and complexity of the masks are the same as in the previous models. The optimal masks obtained for the lift, $L$, the drag, $D$, and the flight path angle, $\gamma$, using fuzzy inductive reasoning are:

$$B13_{L\ fuzzy} = \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -2 & 0 & -3 \\ 0 & -4 & +1 & 0 & 0 \end{pmatrix} \end{array}$$

$$B13_{D\ fuzzy} = \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & 0 & -3 & -4 \\ 0 & 0 & 0 & +1 & 0 \end{pmatrix} \end{array}$$

$$B13_{\gamma\ fuzzy} = \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & -1 & 0 \\ -2 & 0 & -3 & -4 & 0 \\ 0 & 0 & 0 & 0 & +1 \end{pmatrix} \end{array}$$

**<u>Model B14</u>** is very similar to the B4 Model. The only difference is that $M$ and $\theta$ are a little increased. The intention of this model is that of reducing as much as possible the influence of the elevator deflection on the aerodynamic moment. The sampling interval for the B14 model is $\delta t = 12$ sec. The optimal masks obtained for the qualitative variables $L$, $D$, and $\gamma$ using fuzzy inductive reasoning are:

$$B14_{L\ fuzzy} = \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & -2 \\ -3 & -4 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 \end{pmatrix} \end{array}$$

$$B14_{D\ fuzzy} = \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ -2 & -3 & 0 & -4 & 0 \\ 0 & 0 & 0 & +1 & 0 \end{pmatrix} \end{array}$$

$$B14_{\gamma\ fuzzy} = \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} \delta_e & \delta_T & L & D & \gamma \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & -2 \\ -3 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 & +1 \end{pmatrix} \end{array}$$

Notice that these optimal masks are almost the same optimal masks obtained for the B4 model in the fuzzy case. This is due to the very similar trajectory behavior presented by both models. For this reason, the transition from the B4 to the B14 model was the one that presented the most difficulties for being detected.

#### 4.3.3.5 A FIR–Based Continuous Fault Monitoring System

The inductive reasoning functions were used to qualitatively and inductively reason about the measurement data taken from five ACSL simulation runs. The quantitative data were used to build five qualitative models that represent the behavior of the airplane in the vicinity of five different steady–state trajectories. Then, as explained in previous sections, a qualitative model library was constructed with those five qualitative models representing a variety of structural changes of the aircraft. A threshold error alarm, based on a moving average error filter, was incorporated to decide when a structural change has taken place. The complete fault monitoring scheme of this assumption is shown in Figure 4.4.

If a sudden structural change occurs, the qualitative model in use will receive inputs that have never been seen before, which means that it will no longer be able to predict the future behavior of the system. If this condition holds, an alarm will be triggered indicating that an accident has happened. The failure detector works through a moving average alarm that counts the incorrect forecasts within a given period of time, and trips the alarm whenever the accumulated number of incorrectly predicted future states surpasses a threshold that is built into the detector. Once the original model is no longer able to predict the system behavior, the models in the library are consulted in an orderly fashion, trying to identify the one that best predicts the new behavior of the aircraft.

The recognition of the type of accident starts when the qualitative model initiates the search for another set of optimal masks that represent the behavior of the new system. The process is very similar to that of the failure detection. For each qualitative model (optimal mask) in the data base, the forecast values of this new qualitative model are compared with the true “measured” values of the quantitative model of the structurally modified aircraft. If the error surpasses the threshold, the currently investigated qualitative model does not represent the behavior of the system, and another qualitative model must be chosen from the library.

If the accident is included in the library, exactly one qualitative model should pass the test, irrespective of when the error occurs during the simulation. If no qualitative model is able to pass the test, the threshold value has been chosen too stringently, whereas multiple models passing the test indicate poor failure discrimination. It is hoped that the range of acceptable threshold values is wide, i.e., the correct accident can be identified using a fairly small threshold value, whereas all other sets of optimal masks will call

Crisp Model B–4

$$
\begin{array}{c}
\\ {}_{step}\backslash^{var} \\ 2495 \\ 2496 \\ 2497 \\ 2498 \\ 2499 \\ 2500 \\ 2501 \\ 2502 \\ 2503 \\ 2504 \\ 2505 \\ 2506 \\ 2507 \\ 2508 \\ 2509 \\ 2510
\end{array}
\begin{array}{c}
\text{Error Matrix} \\
\begin{array}{ccc} error(L) & error(D) & error(\gamma) \end{array} \\
\left(\begin{array}{ccc}
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & -1 \\
0 & 0 & -1 \\
0 & 1 & 0 \\
-2 & 2 & -1 \\
-2 & 1 & -1 \\
1 & 1 & 0 \\
0 & 0 & 0 \\
0 & 1 & 2 \\
-2 & 2 & 2
\end{array}\right)
\end{array}
\qquad
\begin{array}{c}
\text{Alarm Vector} \\
alarm \\
\left(\begin{array}{c}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1
\end{array}\right)
\end{array}
$$

Fuzzy Model B–4

$$
\begin{array}{c}
\\ {}_{step}\backslash^{var} \\ 2495 \\ 2496 \\ 2497 \\ 2498 \\ 2499 \\ 2500 \\ 2501 \\ 2502 \\ 2503 \\ 2504 \\ 2505 \\ 2506 \\ 2507 \\ 2508 \\ 2509 \\ 2510
\end{array}
\begin{array}{c}
\text{Error Matrix} \\
\begin{array}{ccc} error(L) & error(D) & error(\gamma) \end{array} \\
\left(\begin{array}{ccc}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
1 & 0 & -2 \\
-2 & -1 & 1 \\
0 & -2 & 1 \\
-2 & 1 & -1 \\
-1 & 2 & 0 \\
2 & -1 & 1 \\
-1 & 1 & 0 \\
-2 & 0 & -2 \\
-2 & 2 & 0 \\
-2 & 2 & 0
\end{array}\right)
\end{array}
\qquad
\begin{array}{c}
\text{Alarm Vector} \\
alarm \\
\left(\begin{array}{c}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1
\end{array}\right)
\end{array}
$$

Figure 4.18: Error and alarm matrices of the qualitative detection of a B4 to B747 structural change.

for a considerably higher threshold value before they pass the test, if they do it at all. Since each of the qualitative models in the library represents a particular type of accident, in the very moment when a library model has become capable of correctly predicting the behavior of the structurally modified aircraft, the FMS is able to conclude what accident has happened, i.e., to discriminate between different types of accidents; and with this information, it can decide upon an appropriate corrective action to be taken. Once the new model is identified, the monitoring system acts exactly as before the accident.

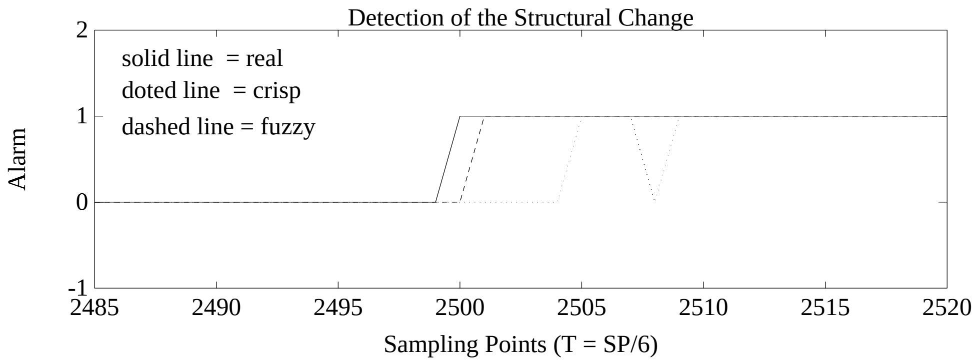


Figure 4.19: Detection alarm *vs.* time.

Four experiments were carried out, one for each possible structural change, or accident (B4 to B747, B4 to B5, B4 to B13, and B4 to B14). Each one started with a completely trimmed flight using the B4 model excited with harmonic functions of fairly long periods, and in each one, a sudden structural change was scheduled to occur after time $t = 2500$ sec. As a matter of fact, five rather than four experiments were performed, because the first of the four possible structural changes was simulated twice, once using crisp inductive reasoning and once using fuzzy inductive reasoning. The ACSL simulation's communication interval had to be set to 1 sec. in the five experiments, because that is the greatest common divisor of all sampling intervals of the models in the qualitative library (6, 11, 19, 15, and 12 sec), and the data generated by these simulation runs must fit every model.

The B4 to B747 transition will be analyzed in more detail than the others, since it will be used to compare the crisp and fuzzy methodologies. Figure 4.18 depicts the differences between the detection process corresponding to the crisp and fuzzy methodologies applied to the B4 to B747 structural change. Notice that the detection of the accident is made faster with the fuzzy approach. This is due to the reduction in the number of forecast errors, which permits reducing

Crisp Model B–747

| $step \backslash^{var}$ | $error(L)$ | $error(D)$ | $error(\gamma)$ | $alarm$ |
|---|---|---|---|---|
| | **Error Matrix** | | | **Alarm Vector** |
| 2497 | 0 | −1 | −2 | 1 |
| 2498 | 1 | −1 | −1 | 1 |
| 2499 | 1 | 0 | −1 | 1 |
| 2500 | 0 | −1 | −1 | 1 |
| 2501 | 0 | −1 | −2 | 1 |
| 2502 | 0 | −2 | −1 | 1 |
| 2503 | −1 | −2 | −1 | 1 |
| 2504 | −1 | −2 | −1 | 1 |
| 2505 | −1 | −1 | 0 | 1 |
| 2506 | −1 | 0 | −1 | 1 |
| 2507 | 0 | −1 | −1 | 1 |
| 2508 | 0 | −1 | −1 | 1 |
| 2509 | 0 | 0 | 0 | 0 |
| 2510 | 0 | 0 | −1 | 0 |
| 2511 | 0 | 0 | 0 | 0 |
| 2512 | 0 | 0 | −1 | 0 |
| 2513 | 0 | 0 | 0 | 0 |

Fuzzy Model B–747

| $step \backslash^{var}$ | $error(L)$ | $error(D)$ | $error(\gamma)$ | $alarm$ |
|---|---|---|---|---|
| | **Error Matrix** | | | **Alarm Vector** |
| 2497 | 0 | −1 | −2 | 1 |
| 2498 | 1 | −1 | −2 | 1 |
| 2499 | 1 | −1 | −1 | 1 |
| 2500 | 0 | −1 | −1 | 1 |
| 2501 | 1 | −1 | −2 | 1 |
| 2502 | 1 | 0 | −1 | 1 |
| 2503 | 0 | 0 | −1 | 0 |
| 2504 | 0 | 0 | 0 | 0 |
| 2505 | 0 | 0 | 0 | 0 |
| 2506 | 0 | 0 | 0 | 0 |
| 2507 | 0 | 0 | 0 | 0 |
| 2508 | 0 | 0 | −1 | 0 |
| 2509 | 0 | 0 | 0 | 0 |
| 2510 | 0 | 0 | 0 | 0 |
| 2511 | 0 | 0 | 0 | 0 |
| 2512 | 0 | 0 | −1 | 0 |
| 2513 | 0 | 0 | 0 | 0 |

Figure 4.20: Recognition of the B747 model after the accident.

the threshold value of the moving average error filter in the alarm modules. Figure 4.19 represents the detection alarm against time.

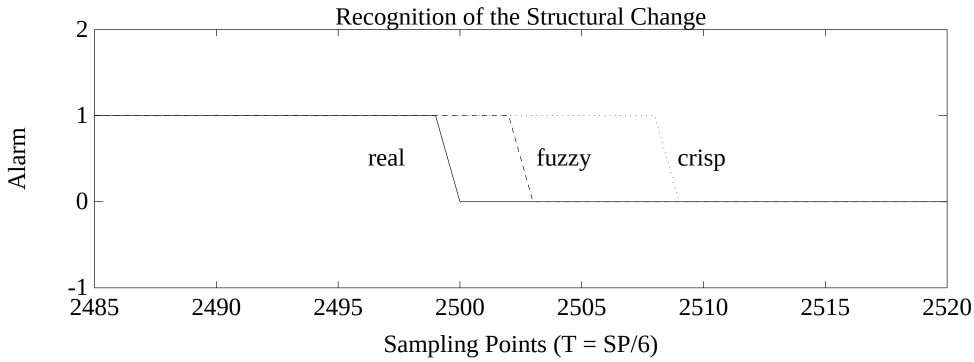


Figure 4.21: Recognition alarm *vs.* time.

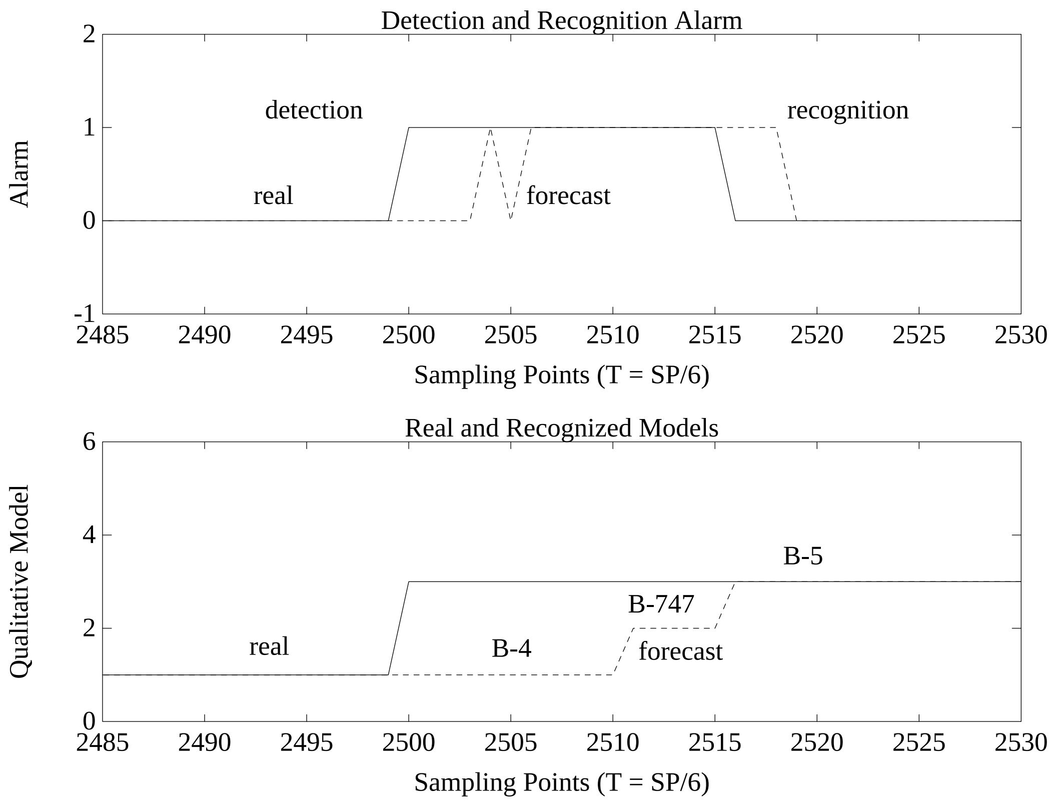


Figure 4.22: Detection and recognition of the B4 to B5 structural change.

Figure 4.20 shows the differences between the crisp and the fuzzy methodologies when recognizing the B747 model after the accident. Figure 4.21 represents the recognition alamrm against time. Notice that the crisp approach has more difficulties than the fuzzy approach in recognizing the right qualitative model from the library.

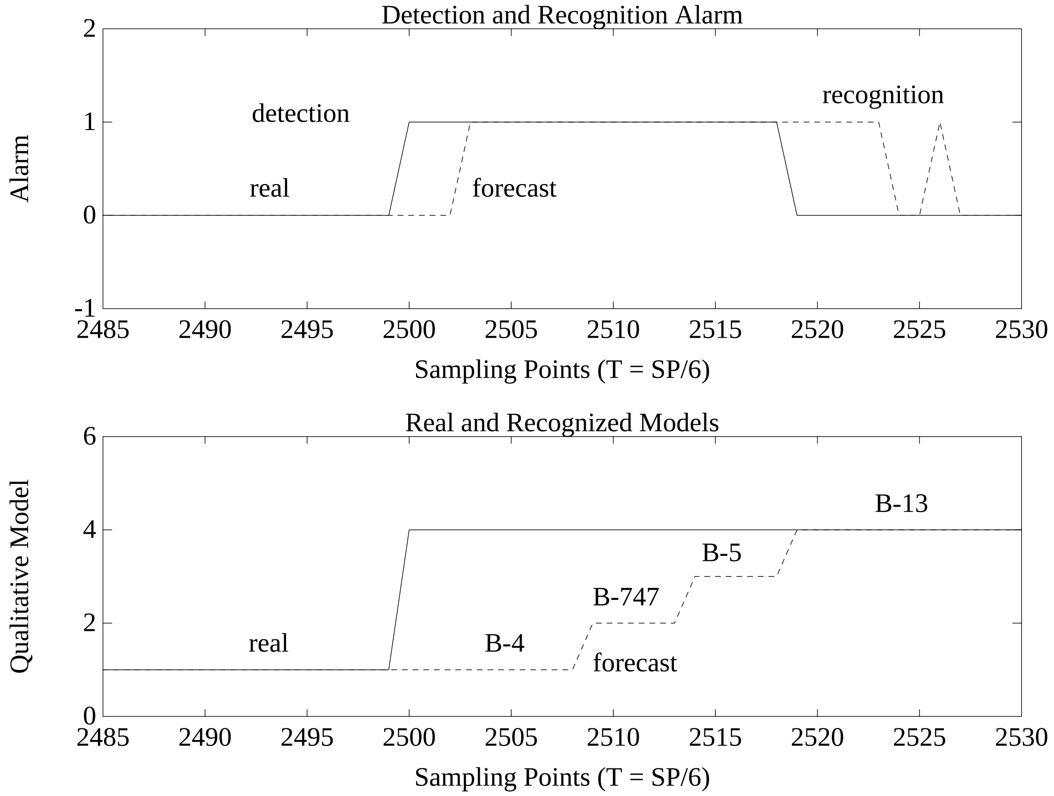


Figure 4.23: Detection and recognition of the B4 to B13 structural change.

The main advantage of the fuzzy methodology over the crisp methodology is a significant reduction of the number of forecasting errors. The crisp approach has between 25% and 33% errors in the forecast points, while in the fuzzy approach the percentage of errors varies consistently between 3% and 12% when the correct qualitative model is being used. This permits a reduction in the threshold value used by the FMS. Furthermore, with the crisp approach, an incorrect forecast often leads to an entire chain of consequence errors, which would immediately trip the alarm if the accumulation window was selected too narrowly. The fuzzy approach doesn't exhibit this problem any longer. False alarms are no longer caused by error chains when using FIR, and therefore, the accumulation window of the threshold error alarm can be made much shorter, which allows the structural change to be detected several points earlier than using the crisp approach.

With respect to the detection and recognition of the accident itself, the B4 to B747 transition is detected four sampling intervals earlier, and recognized six sampling intervals earlier with the fuzzy approach than with the crisp approach. This unambiguous identification of the post–accident steady state, i.e., the new set of optimal masks that correctly describe the post–accident behavior, is the other main advantage of FIR over crisp inductive reasoning.

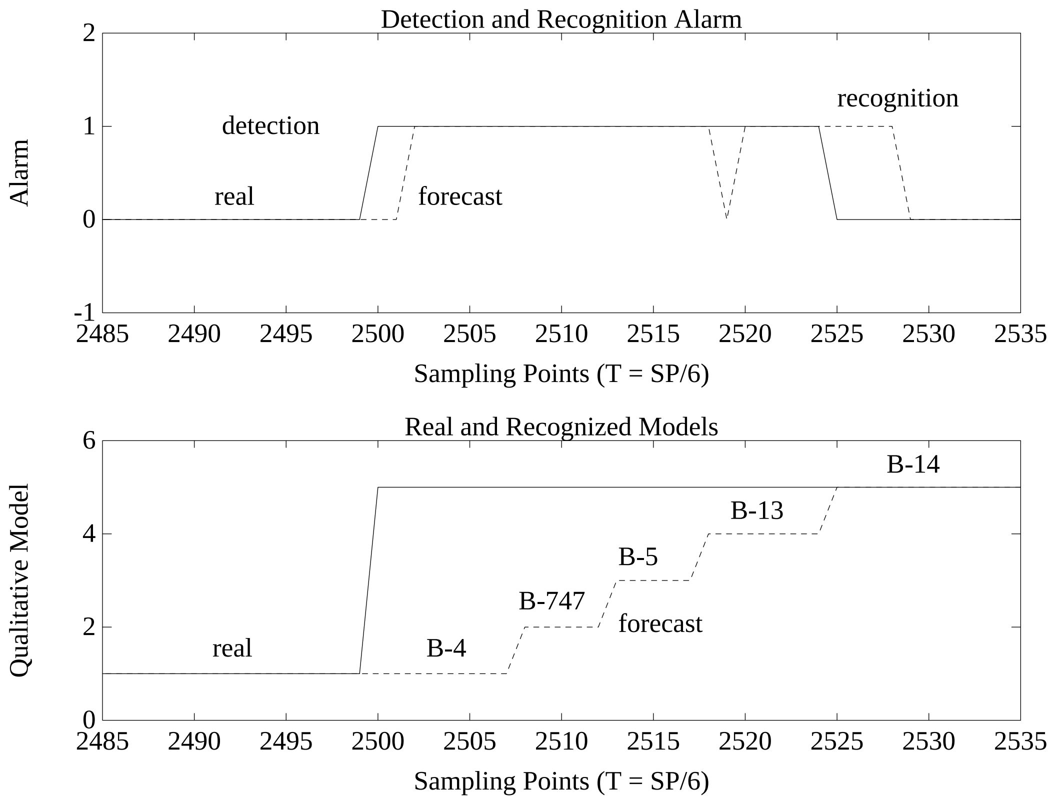


Figure 4.24: Detection and recognition of the B4 to B14 structural change.

Let us now play with the other three models, i.e., models B5, B13, and B14. In this experiment, a structural change occurs at the time step 2500. Once it has been detected (the alarm is triggered) by the FMS, the system goes into the library of qualitative models and tries to forecast the behavior of the structuraly modified system with each of the available models, until the right model is found (the alarm is shut–off). The switching between models in the library occurs if the alarm condition is maintained during five or more time steps, that is, a model is not cosidered the appropriate one if it cannot predict the behavior of the new system during five time steps.

The detection and recognition alarms of the B4 to B5 transition are depicted in Figure 4.22, and those of the B4 to B13 transition are depicted in Figure 4.23.

The B5 and B13 post–accident states exhibit very similar behavioral patterns. However, the fuzzy FMS was capable of unambiguously recognizing them, whereas the crisp FMS had difficulties discriminating between them. This is another important advantage of fuzzy inductive reasoning as compared to crisp inductive reasoning. The available historical knowledge maintained in the experience data base helps to sharpen the discriminatory power of the fuzzy FMS.

Finally, the detection and recognition alarms of the B4 to B14 transition are depicted in Figure 4.24.

The first thing that can be concluded from this Qualitative Model Library implementation is that a FIR–based FMS proved better than a crisp–based FMS. Important shortcomings such as number of errors, error chains, and identification delays are improved, or completely avoided by using the former instead of the latter approach.

With respect to the other objective of this example, i.e., demonstrating that a FIR–based FMS using the Qualitative Model Library fault monitoring approach could be used as a watchdog autopilot to determine when a structural malfunction occurs, and to hypothesize about the nature of this malfunction, it can be said that it has been fully satisfied. It does not matter that the proposed structural changes were not realistic emergencies of a real aircraft. The fact that the qualitative watchdog autopilot proposed here proved capable of detecting, characterizing, and identifying sudden changes in the simulated flying conditions is a good indication that the proposed methodology would be able to deal with a real emergency in a real aircraft just as well. The main shortcomings of this approach come from the number of possible structural changes, and the time needed to recognize the model in the library.

## 4.4 Implementation of Forecasting All Possible Structures Assumption

As had already been stated in the first chapter, in a variable structure system (VSS) a structural change cannot be viewed as a fault or an emergency taking place, but is a perfectly normal event that will happen regularly during system operation. Since these changes can occur frequently, it does not seem practical to search a qualitative model library for another model that fits the new situation. By the time the new model has been identified, several more structural changes might have taken place. Consequently, a fault monitoring approach is needed that is capable of detecting that a structural change has occurred and is capable of identifying the new structural mode of the system almost instantaneously.

Thus, the main role of the FIR–based FMS will be that of identifying the right structural mode the VSS is in at any point in time, instead of detecting and recognizing developing emergencies with diagnosis and analysis purposes. For this reason, the *Forecasting All Possible Structures* fault monitoring approach is proposed instead of the previously used *Qualitative Model Library* approach.

In order to demonstrate the application of this approach to a VSS, two examples will be presented. The first is a fairly simple example commonly used in the VSS literature [Strömberg *et al.*, 1993], and is intended to show how the FMS works when applied to a VSS. The second is a very tough example, intended to fully demonstrate the capabilities of the FIR methodology for correctly discriminating between different structural modes that are characterized by very similar behavioral patterns.

### 4.4.1 The Interconnected Tank Model

This example describes a system that consists of two interconnected water tanks. The system description is borrowed from [Strömberg *et al.*, 1993], and is shown in Figure 4.25.[16] The quantitative system has one inflow, the incoming water flow, $Q$, and two outflows, the outgoing water flows, $Q_1$ and $Q_2$. The qualitative model used by the inductive reasoner has one input, the incoming water flow, $Q$, and two outputs, the water levels in the two tanks, $Y_1$ and $Y_2$. The system can operate in three different structural modes:

[16] This figure has been taken from [Strömberg *et al.*, 1993].

i) **Mode I.** The water level in the first tank does not reach up to the height of the separating wall, $Y_1 < H_1$. In this mode, the first tank can be either filled or emptied, whereas the second tank can only be emptied.

ii) **Mode II.** The first tank is full up to the separating wall between the two tanks, $Y_1 = H_1$, whereas the level in the second tank does not extend all the way up to the separating wall, $Y_2 < H_1$. In this mode, the first tank can only be emptied, whereas the second tank can be either filled or emptied.

iii) **Mode III.** Both tanks are full to or beyond the level of the separating wall, $Y_1 = Y_2 \geq H_1$. In this mode, both tanks are being filled or emptied together.

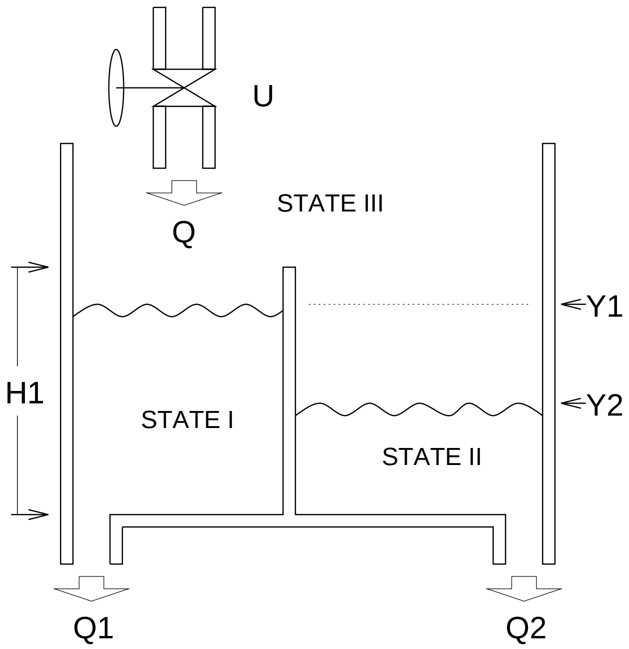


Figure 4.25: Interconnected water tanks system.

Due to the simplicity of this system, it was modeled directly in Matlab rather than using a general–purpose simulation language. Three quantitative simulations, each with the system in one of the three possible structural modes, were conducted across 1000 sec, with the inputs being excited with random binary noise. The slowest time constant was $t_{settling} = 1.2sec.$, the communication interval was set to $0.3sec.$, the sampling interval was set to $\delta t = 0.6sec.$, and the mask depth was set to 3. To obtain the three qualitative models, the Optimal Mask Analysis was performed with the following characteristics:

$$\begin{array}{lcl} \text{number of variables} & = & 3; \\ \text{number of qualitative levels} & = & 3; \\ \text{maximum mask complexity} & = & 4; \\ \text{depth of the masks} & = & 3. \end{array}$$

For the first operating mode, i.e., mode I, the optimal masks for the two output variables $y_1$ and $y_2$ are:

$$\text{mode I } (y_1) \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} Q & Y_1 & Y_2 \end{array} \\ \begin{pmatrix} -1 & -2 & 0 \\ -3 & 0 & 0 \\ 0 & +1 & 0 \end{pmatrix} \end{array}$$

$$\text{mode I } (y_2) \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} Q & Y_1 & Y_2 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & +1 \end{pmatrix} \end{array}$$

for the operating mode II, they are:

$$\text{mode II } (y_1) \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} Q & Y_1 & Y_2 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & +1 & 0 \end{pmatrix} \end{array}$$

$$\text{mode II } (y_2) \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} Q & Y_1 & Y_2 \end{array} \\ \begin{pmatrix} -1 & 0 & -2 \\ -3 & 0 & 0 \\ 0 & 0 & +1 \end{pmatrix} \end{array}$$

and for the operating mode III, they are:

$$\text{mode III } (y_1) \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} Q & Y_1 & Y_2 \end{array} \\ \begin{pmatrix} -1 & -2 & 0 \\ -3 & 0 & 0 \\ 0 & +1 & 0 \end{pmatrix} \end{array}$$

$$\text{mode III } (y_2) \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} Q & Y_1 & Y_2 \end{array} \\ \begin{pmatrix} -1 & -2 & 0 \\ -3 & 0 & 0 \\ 0 & 0 & +1 \end{pmatrix} \end{array}$$

With these modes already characterized, an additional quantitative simulation was performed, once more across 1000 sec of simulated time, but during this simulation, structural changes took place once every 200 sec, driving the system through the three possible modes. This information constitutes the real data that are continuously compared against the values obtained from the three qualitative simulations.

Figure 4.26 shows the entire simulation of this experiment. The first graph depicts the behavior of the input variable, $Q$. The oscillatory behavior is due to the random binary noise used as input signal. The second and third graphs of the same figure correspond to the output signals, $Y_1$ and $Y_2$, i.e., the water levels in the two tanks. Notice that when the first tank is full to the top of the separating wall, and while the second tank is being filled, the horizontal line corresponding to the constant water level of the first tank exhibits numerical noise, an anomaly of the MATLAB simulation.

Figure 4.27 shows the identification results by comparing the real and predicted modes as functions of time. Notice that there usually is a short delay after the true mode change, before the mode switch is activated. This delay is caused by the moving average error filter. If the number of past points used in the filter is reduced, the delay time will be shortened, but this goes at the expense of occasional errors in determining the correct operating mode of the system as a consequence of sporadic errors in the qualitative simulation.

This is a fairly straightforward example since the different modes can easily be discerned by the naked eye. It does not take much discriminatory power for the inductive reasoner to distinguish between the three operational modes of the system. The example was chosen only to show in a clear manner how

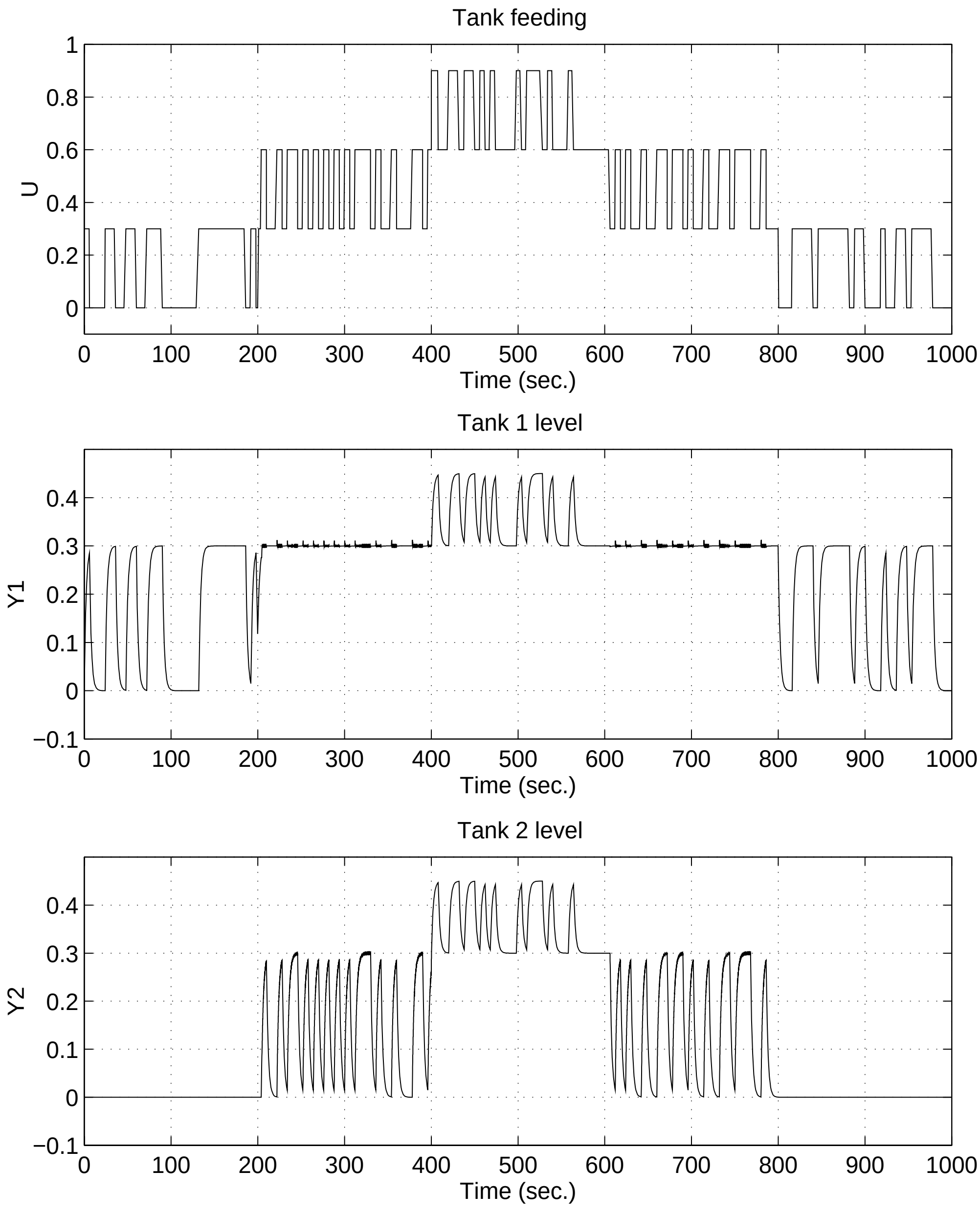


Figure 4.26: Simulation of the interconnected water tank example.

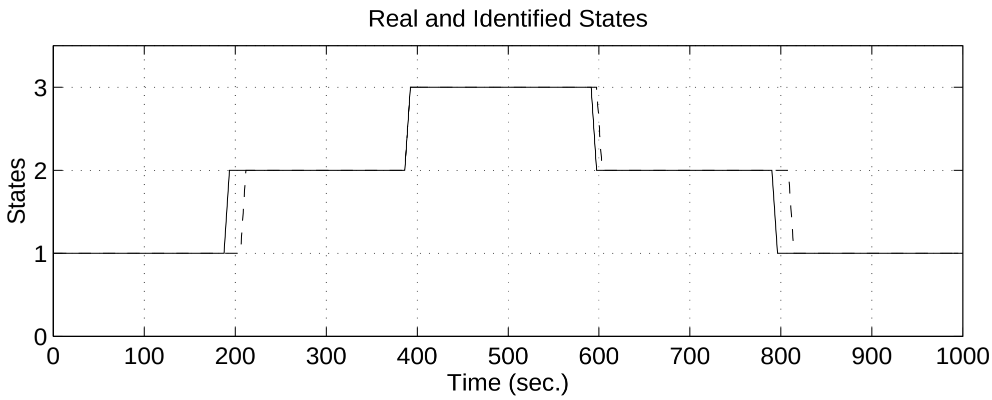


Figure 4.27: Real and identified structural modes.

the methodology works, and since it had been previously discussed in the VSS literature [Strömberg *et al.*, 1993].

### 4.4.2 Electrical Circuit

The second example is much more involved. It is much more difficult for the inductive reasoner to correctly identify the various structural modes, and to know when a transition from one mode to another takes place. It is therefore proposed as a benchmark for structure identification.

The system consists of an electrical circuit containing three binary switches as shown in Figure 4.28. Consequently, the system can be in any one of eight different structural modes depending on the switch positions. The resulting eight structural modes exhibit behavioral patterns that are different enough to be characterized, yet similar enough to make their correct identification a difficult problem.

#### 4.4.2.1 The Quantitative Model

The system has two inputs, $U_1$ and $I_1$, one output, $U_{R4}$, and several electrical components. The quantitative circuit model has been built using Dymola [Elmqvist, 1993; Elmqvist *et al.*, 1993] that in turn generates an ACSL

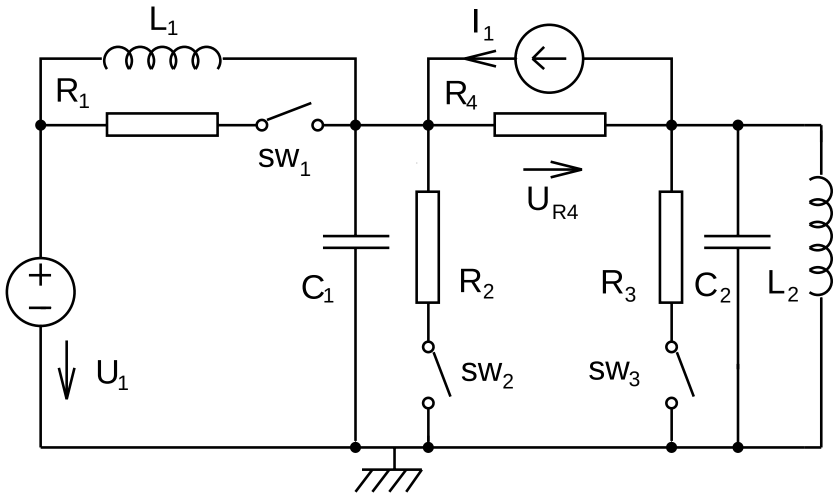


Figure 4.28: Electrical circuit variable structure system.

simulation model. The use of Dymola, an object–oriented modeling language, allows the treatment of each model component as a particular subclass of a specialized library. In this example, the library includes electrical components. Two of the major advantages of this language, beside from its object orientation, are on the one hand, that the modeling of complex systems can be reduced to a very few code lines if the specialized library is already available, and on the other hand, that this very simple code can be automatically translated to ACSL code, even if it contains implicit descriptions of state and time events. The Dymola model for the electrical circuit of Figure 4.28 is presented below.

```
model Circuit
  submodel (CSource)   I0
  submodel (VSource)   V0
  submodel (Resistor)  R1(R = 50.0), R2(R = 10.0), ->
                       R3(R = 100.0), R4(R = 50.0)
  submodel (Capacitor) C1(C = 10.0E-6), C2(C = 10.0E-6)
  submodel (Inductor)  L1(L = 6.0E-3), L2(L = 3.0E-3)
  submodel (Switch)    Sw1, Sw2, Sw3
  submodel Common

  input  u1, u2, o1, o2, o3
  output y
  node   n0, n1, n2, n3, n4, n5, n6

    connect ->
      Common  at    n0, ->
      V0      from  n1  to  n0, ->
      R1      from  n1  to  n2, ->
```

```
        Sw1    from n2 to n3, ->
        L1     from n1 to n3, ->
        C1     from n3 to n0, ->
        R2     from n3 to n4, ->
        Sw2    from n4 to n0, ->
        I0     from n5 to n3, ->
        R4     from n5 to n3, ->
        R3     from n5 to n6, ->
        Sw3    from n6 to n0, ->
        C2     from n5 to n0, ->
        L2     from n5 to n0

    V0.V0 = u1
    I0.I0 = u2

    Sw1.OpenSwitch = o1
    Sw2.OpenSwitch = o2
    Sw3.OpenSwitch = o3

    y = R4.u
end
```

The code should be self–explanatory. It starts out with the invocation of the objects (their instantiation from object classes), and then provides a topological connection of the circuit similar to the netlist of a Spice program.

In a first quantitative experiment, the switch positions were frozen in one of the eight possible positions, and the two inputs were excited with binary random noise to collect data (1000 data points) for characterizing the behavior of the system in that particular mode. The experiment was repeated for all eight switch combinations, characterizing the behavior of the system in all eight possible structural modes.

#### 4.4.2.2 The Qualitative Model

The numerical simulations described above were sampled once every 0.0003 sec. This means that each structural mode was simulated during 0.3 sec of real time.[17] Using the FIR methodology, eight different qualitative models were obtained, each one with its own optimal masks and its own set of *landmarks* (the borderlines between neighboring qualitative classes). The numerical information was recoded into five qualitative classes. The mask candidate matrices were designed to cover the slowest time constant of the system, for that reason they are of $depth = 5$, but with only three rows of data separated

[17] The eight structural modes were considered to have the same stability constants for reasons of simplicity.

by rows filled with zeros. The maximum complexity of the masks was set to 4. The optimal masks found for each of the structural modes were:

$$\text{mode } 000 \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} U_1 & I_1 & U_{R_4} \end{array} \\ \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -3 & +1 \end{pmatrix} \end{array}$$

$$\text{mode } 100 \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} U_1 & I_1 & U_{R_4} \end{array} \\ \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \\ 0 & -3 & +1 \end{pmatrix} \end{array}$$

$$\text{mode } 010 \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} U_1 & I_1 & U_{R_4} \end{array} \\ \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \\ -3 & 0 & +1 \end{pmatrix} \end{array}$$

$$\text{mode } 001 \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} U_1 & I_1 & U_{R_4} \end{array} \\ \begin{pmatrix} 0 & 0 & -1 \\ 0 & -2 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \\ -3 & 0 & +1 \end{pmatrix} \end{array}$$

$$\text{mode } 110 \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} U_1 & I_1 & U_{R_4} \end{array} \\ \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \\ -3 & 0 & +1 \end{pmatrix} \end{array}$$

$$
\text{mode } 101 \quad = \quad \begin{array}{ll} {}_{t}\backslash^{x} & \begin{array}{ccc} U_1 & I_1 & U_{R_4} \end{array} \\ \begin{array}{l} t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} & \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -3 & +1 \end{pmatrix} \end{array}
$$

$$
\text{mode } 011 \quad = \quad \begin{array}{ll} {}_{t}\backslash^{x} & \begin{array}{ccc} U_1 & I_1 & U_{R_4} \end{array} \\ \begin{array}{l} t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} & \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \\ -3 & 0 & +1 \end{pmatrix} \end{array}
$$

$$
\text{mode } 111 \quad = \quad \begin{array}{ll} {}_{t}\backslash^{x} & \begin{array}{ccc} U_1 & I_1 & U_{R_4} \end{array} \\ \begin{array}{l} t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} & \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \\ 0 & -3 & +1 \end{pmatrix} \end{array}
$$

It has to be said that that some of these masks are not the optimal ones found by FIR, but suboptimal that, due to the characteristics of the example, gave better forecasting results than the optimal.[18]

The mode number represents a decimal encoding of the binary (three–bit) mode patterns. For example, mode 000 refers to the particular mode in which all three switches are in their closed positions, whereas mode 101 refers to the particular mode in which switches 1 and 3 are open whereas switch 2 is closed, etc.

[18]It has happened sometimes that the optimal mask obtained by the Optimal Mask Analysis, i.e., the mask with highest quality, is not the mask with the best forecasting capabilities. This obviously has to do with the implemented quality measure, and for this reason, alternate quality measures have been proposed but they are not yet implemented.

#### 4.4.2.3 A FIR–Based Structure Identification System

This problem is much tougher than the two–water–tank problem. Several of the eight modes lead to input/output behavioral patterns that are almost indistinguishable. At least by simply looking at the data, it is not obvious that they were generated from different structural modes.

Before trying to identify the eight different structural modes, let us try to demonstrate that the FIR methodology is capable of discriminating between two very similar structural modes. To this end, two very similar input/output behavior modes were chosen. They were mode 001 (switches 1 and 2 are in the closed position, whereas switch 3 is in the open position), and mode 000 (switches 1, 2, and 3 are all in the closed position). A numerical simulation was carried out across 100 data points with only one structural change in the middle of the simulation. The obtained data streams were used as the "real data" for identifying the structural mode the system is in, but first they were recoded (fuzzified) twice, once for each structural mode involved, using the specific landmarks obtained for the respective qualitative models.[19]

The first graph of Figure 4.29 depicts the results of the mode 001 to mode 000 transition. The continuous line represents the true trajectory behavior of the system that corresponds, from sampling point 1 to sampling point 50, to the structural mode 001; and from sampling point 51 to sampling point 100, to the structural mode 000. The dashed line represents the FIR forecast of the system using the qualitative 001–mode model, and the dotted line represents the FIR forecast of the system using the qualitative 000–mode model. Notice that the prediction made with the qualitative model 001 is very accurate during the first 50 sampling points, whereas it is not accurate at all during the second 50 sampling points, i.e., after the structural change. The reverse can be stated for the qualitative model 000. Its prediction is not accurate at all in the left part of the graph, whereas it is very accurate in the right part of the graph. This means that each qualitative model is capable of recognizing its own characteristic behavior, but incapable of recognizing the behavior of the other.

The second graph of Figure 4.29 shows the detection and identification results. The structural mode change occurs at sampling point 50. Due to the transient taking place immediately following the switching event, and due

[19]In the previous examples with several operating modes or structural changes (aircraft and two–water–tank), the landmarks obtained for one of these modes were used for recoding the data of the others, however, in this example the circuit is so sensitive, and the structural modes so similar, that each of them had to be recoded with its particular landmarks and consequently, the quantitative data filled into the FMS must be recoded eight different times.

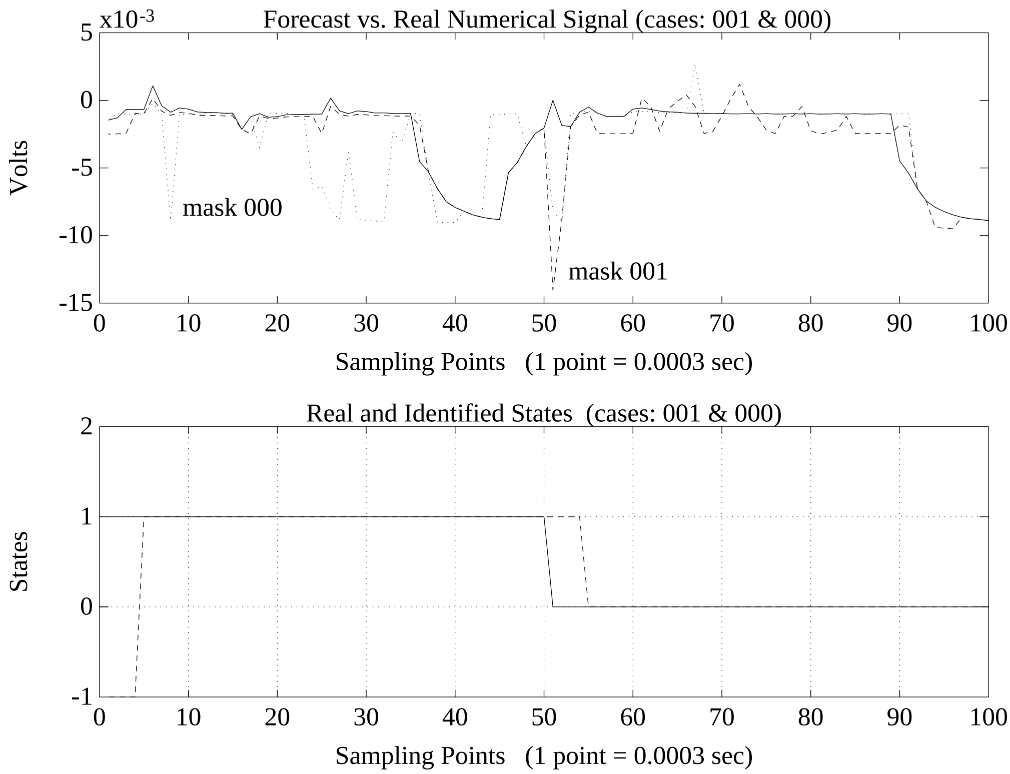


Figure 4.29: Recognition of two very similar structural modes.

to the lag behavior of the error filter (the width of the moving average error filter was chosen to be five sampling intervals), it takes five sampling points, before the mode selector switches from mode 001 to mode 000, i.e., before the FMS detects the structural change and recognizes the new structural mode. Notice that the FIR mode detector was capable of unambiguously identifying the correct modes.

Once the FIR–based FMS has demonstrated its discriminatory power between two very similar structural modes, let us try to find out if it is capable of discriminating between the eight possible structural modes of the circuit. To this end, the numerical model was rerun in a new experiment, this time including a mechanism to change the switch positions once every 50 sampling intervals. By going through eight randomly chosen structural modes, 400 data points of variable structure circuit simulation were obtained to be used as the "real data" for identifying the structural mode that the circuit is in at any one point in time.

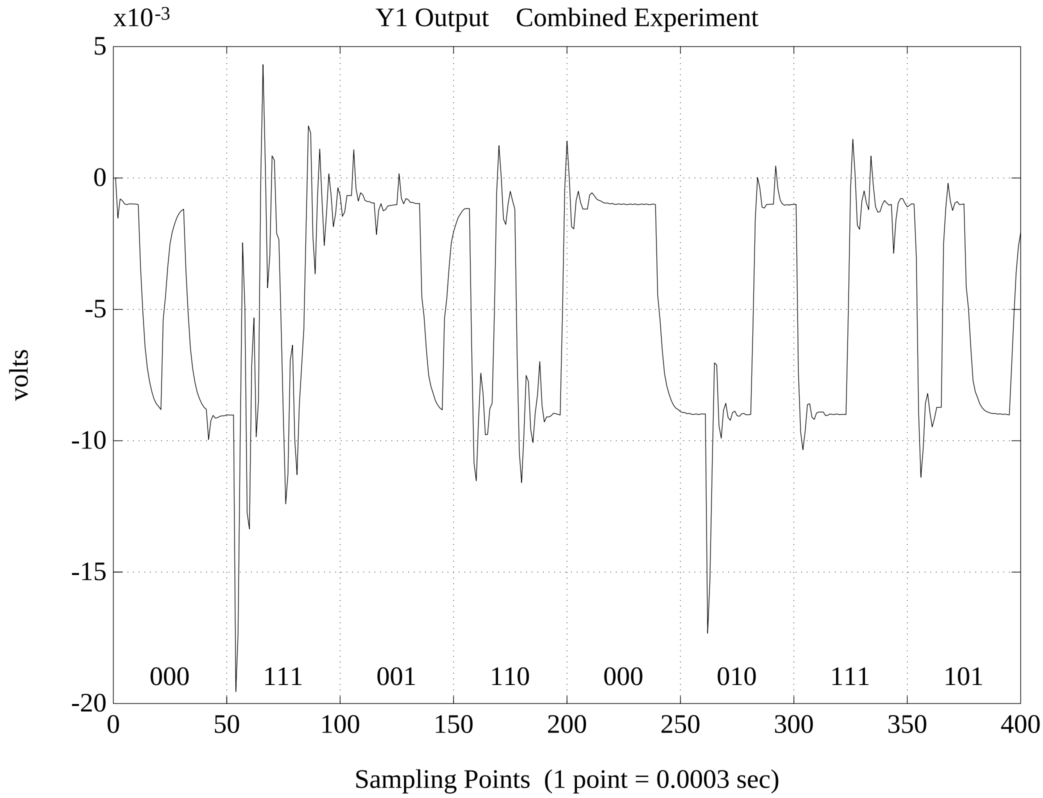


Figure 4.30: Real output of the electrical circuit VSS.

The 400 data points collected from the numerical variable structure simulation were recoded eight times using the specific landmarks obtained for the eight qualitative models, i.e., the same number of qualitative classes for all models, but specific landmarks for each one, in order for each model to be capable of identifying its own behavior. It was now possible to forecast the system behavior eight times over the entire period using the eight different sets of fuzzy optimal masks. It was hoped that each of the eight qualitative models would produce a decent prediction during the time span when the real model was operating in its corresponding structural mode, and a poor prediction during all other times, as it happened in the previous mode transition experiment shown in Figure 4.29.

Figure 4.30 shows the numerical 400 points of VSS simulation. Notice that some structural modes present very similar output behavior.

Figure 4.31 shows the identification results obtained by the inductive reasoning methodology of the eight different structural modes shown in Figure 4.30, with a moving average error filter width of 10 sampling intervals.

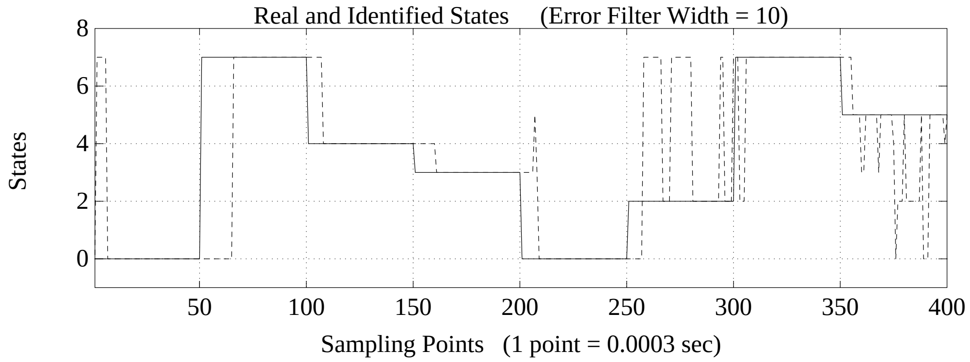


Figure 4.31: Structural modes identification with a 10 sample moving average error filter.

The solid line represents the true mode the system is in, whereas the dashed line denotes the mode identified by the FIR method.

Every 50 data points the system changed from one mode to another. As can be seen, the FIR–based FMS was able to identify the correct mode in which the system is operating in six out of the eight cases. From points 251 to 300, corresponding to mode 010 (switches 1 and 3 closed, and switch 2 open), the FMS toggled between two plausible modes, the correct mode 010 (switches 1 and 3 closed, and switch 2 open) and the very similar mode 111 (switches 1, 2, and 3 open). In this case, the similarities between the two modes were so strong that the FMS was not able to discriminate between them.

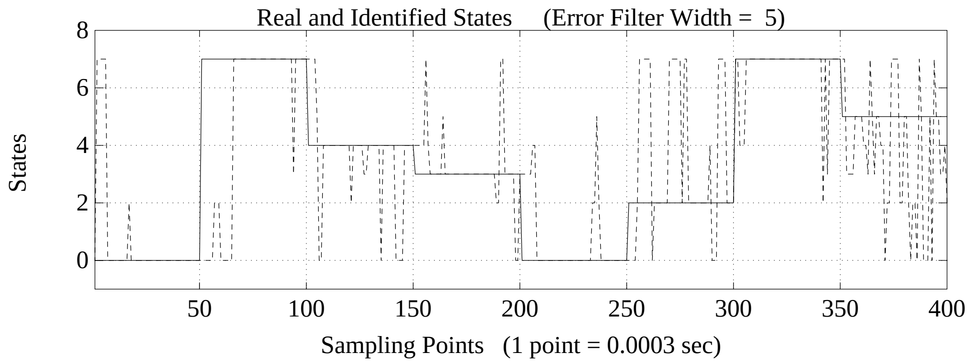


Figure 4.32: Structural modes identification with a 5 sample moving average error filter.

From points 351 to 400, corresponding to mode 101 (switches 1 and 3 open, and switch 2 closed), the FMS was not able to identified anything. In this case, the wrong identification has a different cause, namely an incorrect characterization of the 101 mode. Some structural modes presented more difficulties than others for their characterization due to the effects of the excited input signals on the single output signal of the system.

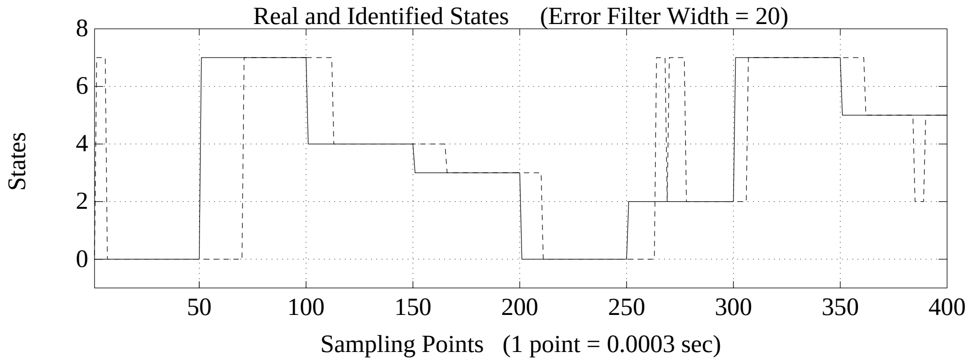


Figure 4.33: Structural modes identification with a 20 sample moving average error filter.

It is possible to overcome some of the identification problems by changing the width of the moving average error filter. Figures 4.32, 4.33, 4.34, and 4.35 show the same experiment as depicted in Figures 4.30 and 4.31, but with different widths of the moving average error filter. As can be seen in Figure 4.32, corresponding to a moving average error filter width of 5 sampling intervals, if the width is reduced, so will be the identification delay, but this goes at the expense of a great amount of local aberrations and occasional errors that, in turn, can lead to a wrong structure identification.

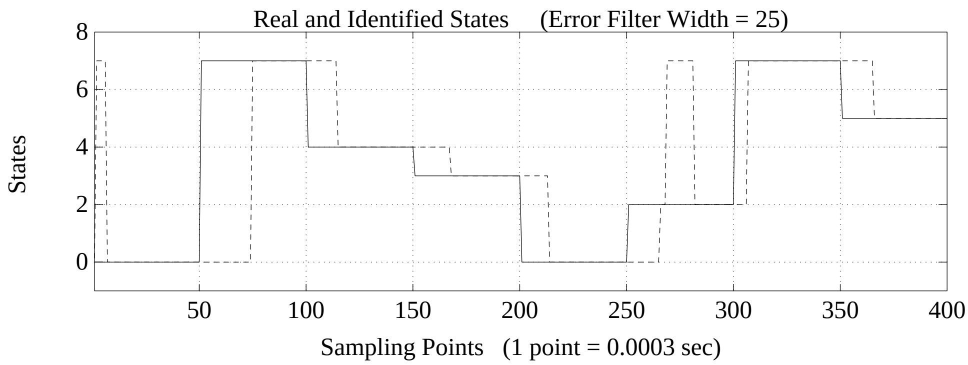


Figure 4.34: Structural modes identification with a 25 sample moving average error filter.

On the contrary, if the width is augmented, as can be seen in Figures 4.33 and 4.34, corresponding to moving average error filters with widths of 20 and 25 sampling intervals respectively, the local aberrations and accidental hits almost vanish, but this goes at the expense of an increased identification delay. The extreme case is shown in figure 4.35 that corresponds to a moving average error filter width of 50 sampling intervals.

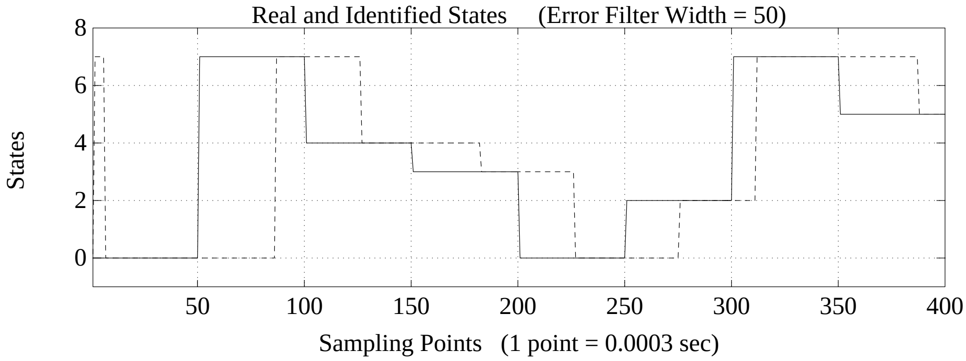


Figure 4.35: Structural modes identification with a 50 sample moving average error filter.

Of course, the structure identification problem can be made less difficult by adding additional output signals to the circuit, so that the FIR method obtains redundancy in determining the right mode. However, it was decided to keep the problem as tough as it is, and present the results as they were obtained.

There are still some problems that must be tackled in order to improve the overall performance of the FIR–based FMS methodology as applied to VSS structure identification problems. They are:

i) The structural mode characterization, i.e., the determination of the most appropriate landmarks and the subsequent identification of the optimal masks.

ii) The selection of the moving average error filter width.

In the two shown examples, the characterization of each structural mode was originally carried out in exactly the same way, i.e., using the same excitation signals and amplitudes, using the same operational limits (landmarks), and assuming the same number of qualitative classes. In the electrical circuit example, this proved to be an inadequate approach, since none of the structural modes could be properly characterized. Then it was decided to compute for each one of the eight circuit structural modes its characteristic landmarks. This proved to be enough to characterize six of the eight modes, but still one mode could not be properly characterized, and another could not be properly identified.

It might be possible to overcome this problem by trying different numbers of qualitative classes for each structural mode, and by varying the type of excitation during the mode characterization phase in one of two ways: exciting all structural modes at the same time but letting the excitation signal go through all structure characteristic signals, or exciting each structural mode separately with its own characteristic signal.[20]

With respect to the other stated problem, i.e., the selection of the moving average filter width, there must be a compromise reached between the affordable delay in the identification of the new structural mode, and the tolerated number of local aberrations or accidental hits that may temporarily lead to a wrong identification.

One more problem, that has to do with the number of possible structural modes the system can be in, should be stated. Theoretically, there are no limitations to the number of structural modes that the FIR–based FMS can manage with the Forecasting All Possible Structures Assumption, but in practice, limitations due to computational capacity will always arise.

## 4.5 Conclusions

In this chapter, the basis for the design of a FIR–based Fault Monitoring System was presented. The resulting FMS proved capable of detecting, isolating, characterizing, identifying, diagnosing, and analyzing developing anomalies that can be considered faults or structural changes, depending on the dynamic system monitored. Some practical examples of applications of a FIR–based FMS were also presented to demonstrate the practicality of the *Qualitative Model Library* and the *Forecasting All Possible Structures* fault monitoring approaches when dealing with dynamic systems and variable structure systems, respectively.

[20]In FIR terms, the characteristic excitation signal of a system is the one that makes the system exhibit the highest possible number of behavioral patterns, i.e., all operating frequencies are richly represented. The higher the number of behavioral patterns observed, the better the characterization of the system.

The aircraft example had two objectives. On the one hand, the intention was that of demonstrating that a FIR–based FMS using the Qualitative Model Library fault monitoring approach can be used as a watchdog autopilot to determine when a structural malfunction occurs, and to hypothesize about the nature of this malfunction. On the other hand, the intention was that of demonstrating the enhanced discriminatory power and improved forecasting capabilities of a fuzzy inductive reasoning FMS in comparison with a crisp inductive reasoning FMS when applied to this problem.

The main advantages of a FMS based on fuzzy inductive reasoning over a FMS based on crisp inductive reasoning are:

1. The number of errors in the qualitatively predicted states has been reduced from one third (using the crisp approach) to less than one tenth (using the fuzzy approach), and the error chains produced by those errors have almost vanished.

2. Whereas the crisp FMS has sometimes difficulties to discriminate between different types of malfunctions, the fuzzy FMS is able to discriminate clearly and unambiguously between different types of malfunctions that make the aircraft react in similar ways.

3. The fuzzy FMS is able to identify malfunctions in a shorter span of simulated time of the quantitative model than its crisp counterpart.

4. In addition, the FIR–based FMS allows to predict a quasi–continuous response spectrum, whereas its crisp counterpart is able to predict discrete (class) values only.

5. The FIR–based FMS presents improved predictive and discriminatory power when compared with its crisp counterpart.

The implementation of the *Forecasting All Possible Structures* approach demonstrated that the FIR–based FMS methodology is a powerful tool for structure characterization and identification in variable structure systems. The parallel forecasting of all possible structures enhances the previously used *Qualitative Model Library* scenario in several ways:

- The disabling of the qualitative simulator during the library search is avoided. In this way, the structural change itself can be simulated and characterized.

- Once a structural change has taken place, its detection and the identification of the new structural mode are almost instantaneous. In this way, the FIR can switch from one structural mode to another almost immediately depending only on the width of the moving average error filter.

- The determination of a qualitative model capable of representing the behavior of the system *at all times* is guaranteed, irrespective of the structural mode the real system is in.[21]

Two examples have been presented, a (fairly simple) two–water–tank problem, and an electrical circuit model containing three switches. The latter example is a very tough problem, and it is therefore proposed as a benchmark problem for structure identification by means of qualitative algorithms and codes.

The results shown in this chapter confirm that the combination of a FIR–based supervision and control system on–line with a quantitative dynamic system or VSS is a powerful tool that should be considered when qualitative fault monitoring of quantitative dynamic processes is to be attempted.

[21]If and only if that particular structural mode has been previously observed, characterized and modeled.

# Chapter 5

# Reconstruction Analysis

## 5.1 Introduction

When the second implementation of the Systems Approach Problem Solver (SAPS) was first presented in 1987 [Cellier and Yandell, 1987], the epistemological hierarchy of the General Systems Problem Solver (GSPS) methodology[1] had already been implemented up to level 4, i.e., up to the *structural level.* However, the tool had only been used for any practical purposes up to level 3, i.e., up to the *behavioral level.*

To be more precise, the tool that had been used until now is the *Optimal Mask Analysis.* As described in Chapter 3, the Optimal Mask Analysis is a method for reasoning about the behavior of a system while deciding something about its structure. The optimal mask denotes a selection of variables chosen from a larger set that suffice to qualitatively characterize the behavior of a system. In this sense, the Optimal Mask Analysis bridges the gap between the epistemological levels 3 and 4, since it draws its inputs from level 3, but delivers outputs at level 4. When fuzzy measures were added to the methodology, it was hoped that the *Fuzzy Optimal Mask Analysis* would provide the user with an improved resolution that would enable him or her to distinguish better between similar behavioral patterns. This hope was indeed fully justified [de Albornoz

[1]A detailed description of the GSPS epistemological levels has been given in Chapter 3, and SAPS–II has also been presented in Chapter 3

and Cellier, 1994]. However, it was recognized soon that fuzzy reasoning had another beneficial side effect. It enables the user to regenerate estimates of quantitative (real–valued) signals from their qualitative counterparts [Cellier, 1991b].

This initially unintended side effect became soon the cornerstone of the entire methodology, since it enabled us to perform mixed quantitative and qualitative simulations of complex technical processes [Cellier *et al.*, 1996a] and opened up an entire array of interesting new applications of this technology. It turned out that Fuzzy Optimal Mask Analysis is a very powerful tool indeed that enables the user to conclude a lot about the structure of a system under investigation beside from its behavior. For example, in Chapter 4 it was shown how this methodology can be used to successfully discern between different structures in a variable structure model, although the behavioral patterns exhibited by the various different structures were quite similar. This really constituted a breakthrough in the analysis of variable structure systems, and consequently, the Fuzzy Optimal Mask Analysis reached even into the realm of the epistemological level 5, i.e., that of the *meta–structures*.

Yet, we feel that we now have come to a limit of what can be reasonably expected of this tool. Although it is possible to use a level 3 tool to reason about level 5 problems, it may not be efficient to do so. It may be useful to involve a level 4 tool like *Reconstruction Analysis* to enhance the overall efficiency of the analysis.

### 5.1.1 The Need for Reconstruction Analysis

It should be remembered that the main objectives of this dissertation are: a) to help bridge the gap between the two worlds of quantitative computation and qualitative reasoning by developing a combined quantitative/qualitative methodology based on tools steming from General Systems Problem Solving, and b) the application of this combined methodology for qualitative fault monitoring and troubleshooting of dynamic large-scale systems, i.e., for addressing the information overload problems inherent in such systems.

As has been remarked earlier in this dissertation, the comportment of intelligent Fault Monitoring Systems (FMSs) is not fundamentally different from that of human plant operators. They also suffer, in some fashion, from the same overload problem that humans are faced with. The FMS based on Fuzzy Inductive Reasoning (FIR) proposed in Chapter 4, is capable of avoiding this problem if the number of variables and subsystems is kept below reasonable limits, that is, the FIR–based FMS alone suffices to deal with problems of fault monitoring and troubleshooting in small– to medium–sized systems. However, when dealing with large–scale systems, in which the number of variables and subsystems may vary with time, depending on either the scheduled phase of the operation or characteristics of an observed anomaly, also this FIR–based FMS will need tools that help it focus its attention on essential system properties while ignoring features and facets of lesser importance. In terms of the FIR methodology, a tool is needed that continuously provides FIR with the right set of variables to reason with.

In Chapters 2 and 4, the task of fault monitoring was subdivided into various phases[2]. In fault detection, FIR will need to be provided with a carefully selected set of variables to reason with. This set should be the *minimum set of meaningful variables*, minimum, in order to provide FIR with the smallest possible number of variables, and meaningful, in order not to overlook any misbehavior as it occurs. This set may depend on the currently scheduled phase of operation, and several FIR models, reasoning about different subsets of variables, may be used in parallel that monitor different facets of the operation or subsystems of the overall plant.

After detection, the anomaly must be isolated to a particular subsystem. To this end, another set of variables may be selected, depending on the previously observed variables that have exhibited aberrant behavior, a set of variables that may allow the reasoner to learn more about the nature of the observed anomaly.

Once the defective subsystem has been isolated, the anomaly needs to be characterized within the subsystem. Each subsystem is responsible for one or several facets of plant operation. Fault characterization refers to deciding, which facet (or facets) of operation is (are) affected by the fault in order to discriminate between different faults, which means that a comparison between sets of variables must be carried out.

Finally, fault diagnosis relates to determining, which among a foreseen set of faults has actually occurred. For this purpose, it may be necessary to drive the system into a particular operating point first or excite the system in a specific fashion to be able to distinguish (discriminate) between the effects of similar hypothesized faults, while continuously comparing different subsets of variables.

[2]Chapter 2 provides a general description of the fault monitoring phases, whereas Chapter 4 presents a particular description of how these phases are carried out by a FIR–based FMS

Thus, our FIR–based FMS will need to reason about different subsets of variables during each of the aforementioned stages, and consequently, a tool is needed that determines which variables to use in each case. We stipulate that GSPS level 4 tools provide such capabilities. The tool that has been selected and implemented to carry out these tasks is the *Reconstruction Analysis.*

## 5.2 Reconstruction Analysis Methodology

Reconstruction Analysis (RA) originated with the *Reconstructability Analysis* proposed in [Cavallo and Klir, 1981; Cavallo and Klir, 1982]. It was revisited in [Cellier, 1991a; Klir, 1991], and was taken up again in [Cellier *et al.*, 1996b; de Albornoz and Cellier, 1996; de Albornoz *et al.*, 1996b]. It is a tool that is quite closely related to the FIR methodology. It allows to inductively recognize temporal causal structures (potential subsystems) that are characterized by a large number of behavior trajectories, i.e., it can be used to determine the correlations between subsets of these trajectories. In contrast to statistical techniques used for the same purpose, Reconstruction Analysis deals with the behavior information in qualitative (crisp or fuzzy) terms. The input data used by this methodology are the same *recoded* data that are also used by the inductive reasoner. Contrary to the statistical correlation analysis that computes a linear measure of dependence between variables, Reconstruction Analysis operates on non–linear dependence, and is therefore better suited to characterize the behavior of non–linear systems than classical techniques like correlation analysis.

The purpose of the RA methodology is then the determination of the minimum sets of meaningful variables capable of representing the original system, in such a way that each set of variables can be considered as a subsystem. These sets of variables, or subsystems, will be fed into the FIR methodology as the behavior trajectories to reason with.

Since many of the concepts that will be used in this chapter have already been introduced in Chapter 3 (Fuzzy Inductive Reasoning Methodology), this gives us the opportunity to explain the RA methodology by means of an example that will be carried on along the entire chapter.

### 5.2.1 Reconstructability: The Basic Idea

A *mathematical model* of a system consists of a set of variables and a set of rules mapping them into each other. *Input variables* are variables whose behavior is exogenous to the model, i.e., for which no generative rules or maps exist inside the model. All other variables are generated (computed) by the model using some form of mapping mechanism.

The *behavior* of a model can be defined as a description of the states of the endogenous variables in function of the states of the exogenous variables, and the states of endogenous variables at earlier time points. If all feasible behavioral patterns of a model have been recorded, it is possible to predict correctly the states of the endogenous variables from the states of the exogenous variables. This is basically how level 3 tools within the GSPS epistemological hierarchy work. Although a level 3 tool can, in principle, explain the functioning of a system, such a description is evidently never compact, and reasoning along these lines can be quite inefficient when sufficiently many variables are involved (as it usually happens in large–scale systems).

The *structure* of a model can be defined as a description of the rules that are used inside the model to map variables into each other. In the most abstract of senses, determining the structure in a model can be defined as searching for submodels that cooperate in producing the overall model behavior. In doing so, we may, in a first step, try to define the interfaces between the submodels rather than the submodels themselves, i.e., we may be interested in knowing which variables are used at the interfaces of each of the submodels rather than describing the rules themselves that connect these variables to each other. This is what the Reconstruction Analysis tries to accomplish.

For example, Figure 5.1 shows a model, $M$, that is composed of two submodels, $M_1$ and $M_2$. The model contains five variables, the input variable $v_1$, the output variable $v_5$, and the internal variables $v_2$, $v_3$, and $v_4$. $v_1$ is also an input variable of submodel $M_1$, whereas $v_2$, $v_3$, and $v_4$ are output variables of $M_1$. It is possible that the submodels are decomposed further, but their internal structure is not shown, thus, it is not known whether these submodels contain any internal variables themselves.

In RA, *composite structures* are denoted by row vectors. The variables are simply enumerated. Substructures are separated by zero elements in the vector. Thus, the composite structure of model $M$ would be encoded as:

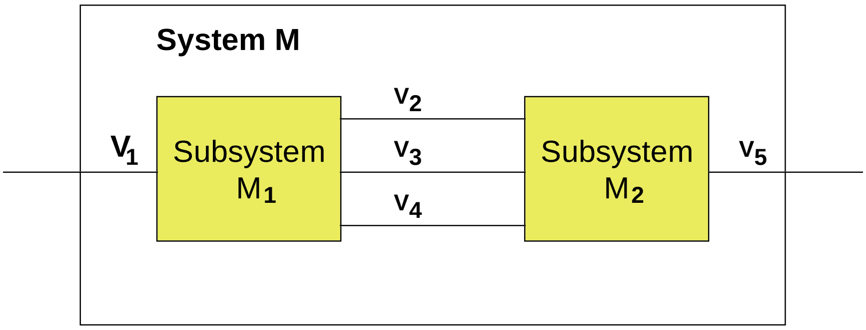


Figure 5.1: System M with subsystems $M_1$ and $M_2$.

$$cs1 \;=\; [\ 1\ 2\ 3\ 4\ 0\ 2\ 3\ 4\ 5\ ] \tag{5.1}$$

denoting that the model $M$ contains two submodels, the first of which relates the variables $v_1$, $v_2$, $v_3$, and $v_4$ to each other, whereas the second describes relationships between the variables $v_2$, $v_3$, $v_4$, and $v_5$. At the chosen abstraction level, there is no distinction made between inputs and outputs, i.e., the model structures are temporally acausal. This may be just as well, since "causality" in physical system modeling is a dubious concept anyway [Cellier and López, 1995].

Structures may be represented in a second fashion using RA, namely in the form of a so–called *binary structure*. The binary structure of a model consists of an enumeration of all binary connections of variables within its submodels. The binary structure of the model $M$ would be:

$$bs1 = \left[ \begin{array}{cc} 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 2 & 3 \\ 2 & 4 \\ 2 & 5 \\ 3 & 4 \\ 3 & 5 \\ 4 & 5 \end{array} \right] \tag{5.2}$$

The only binary connection that is missing is the connection $< 1, 5 >$, since there is no direct connection between the variables $v_1$ and $v_5$. A totally connected model, i.e., a model without any internal structure, contains a complete list of binary connections in its binary structure, whereas a totally unconnected model, i.e., a model containing a set of completely unrelated variables, does not contain any binary connections, i.e., the binary structure is empty. Variables that are not connected to anything else are represented in the binary structure as unitary variables. For example, the composite structure:

$$cs2 = [\ 1\ 2\ 0\ 3\ ] \tag{5.3}$$

would be mapped into the binary structure:

$$bs2 = \left[ \begin{array}{cc} 1 & 2 \\ 3 & 0 \end{array} \right] \tag{5.4}$$

In this case, the zero element denotes that $v_3$ has no relationship with any of the other variables. The totally unconnected composite structure:

$$cs3 = [\ 1\ 0\ 2\ 0\ 3\ ] \tag{5.5}$$

would be mapped into:

$$bs3 = \left[ \begin{array}{cc} 1 & 0 \\ 2 & 0 \\ 3 & 0 \end{array} \right] \tag{5.6}$$

The two types of structure representations (composite and binary) are in some ways equivalent, and one can be mapped into the other by means of the *binary* and *compose* functions:

$$bs = \text{binary}\,(cs)$$

computes the binary structure $bs$ from the composite structure $cs$, and:

$$cs = \text{compose}\,(bs)$$

computes the composite structure $cs$ from the binary structure $bs$.

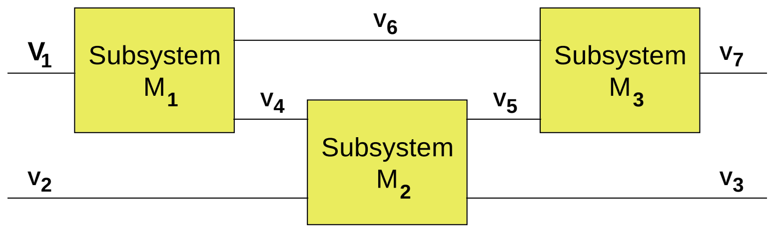


Figure 5.2: Representation of $cs4a$ composite structure.

The transition from the composite structure to its binary counterpart is obviously unique. The reverse is not true. For example, let us consider the following two new composite structures:

$$cs4a = [\ 1\ 4\ 6\ 0\ 2\ 3\ 4\ 5\ 0\ 5\ 6\ 7\ ] \tag{5.7}$$

and:

$$cs4b = [\ 1\ 4\ 6\ 0\ 2\ 3\ 4\ 5\ 0\ 5\ 6\ 7\ 0\ 4\ 5\ 6\ ] \tag{5.8}$$

that may represent the causal structures shown in Figures 5.2 and 5.3. They map into the same binary structure $bs4$:

$$bs4 = \begin{bmatrix} 1 & 4 \\ 1 & 6 \\ 2 & 3 \\ 2 & 4 \\ 2 & 5 \\ 3 & 4 \\ 3 & 5 \\ 4 & 5 \\ 4 & 6 \\ 5 & 6 \\ 5 & 7 \\ 6 & 7 \end{bmatrix} \tag{5.9}$$

RA uses thus the additional rule that the *compose* operator always computes the *minimal structure*, i.e., redundant structures are never shown. This makes sense since, if the variables of Figure 5.3 are algebraic variables, i.e., there is no time difference between any of the variables, then the model of Figure 5.3 contains an algebraic loop involving variables $v_4$, $v_5$, and $v_6$, and when solving this algebraic loop, another structure will be obtained, namely that of Figure 5.2.

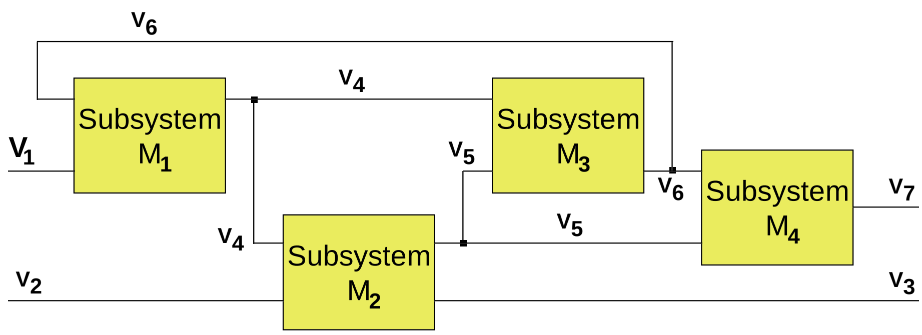


Figure 5.3: Representation of $cs4b$ composite structure.

Once the way in which structures are represented in RA has been explained, the time has come to talk about the main steps of the RA methodology. It has been mentioned that the same qualitative behavior used to feed FIR can also be used to feed RA. The main objective is to obtain a reconstruction of the structure represented by the behavior. To this end, the RA methodology proposes reconstruction hypotheses about what this structure may look like by means of behavior projections and reconstructions. Each one of the reconstructed behavioral patterns may differ from those observed in the original physical system. If this difference, i.e., a distance function between the original and the reconstructed behavior, is sufficiently small, then the proposed structure represents a decent reconstruction of the original system. Finally, the determination of the most suitable reconstruction hypothesis, i.e., the most suitable decomposition into subsystems, is carried out by means of suboptimal search strategies known as: *Optimal Structure Analysis.*

Thus, the RA methodology rests on the following main processes:

i) Generation of Reconstruction Hypotheses.

ii) Reconstruction from Subsystems.

iii) Computation of the Information Distance.

iv) Optimal Structure Analysis.

Each of those will be fully described in the next sections.

The entire reconstruction process is shown in Figure 5.4. An original system with a behavior $[b,p]$ is decomposed (its behavior is projected) into $n$ subsystems with behaviors $[b1,p1]$, $[b2,p2]$ to $[bn,pn]$, that form a reconstruction hypothesis. In a second step, these subsystems are reconstructed two by two (their behavior is recombined) forming a reconstructed system with a behavior $[br,pr]$. The comparison between $[b,p]$ and $[br,pr]$ is the error measure called the *information distance.*

The final step of the methodology, i.e., *Optimal Structure Analysis,* that will be discussed a few sections ahead, proposes different algorithms for the suboptimal determination of the most suitable reconstruction hypothesis, that is, the most suitable decompositions into sybsystems.

### 5.2.2 Generation of Reconstruction Hypotheses

In RA, the *behavior* of a system is defined in exactly the same fashion that had been introduced earlier in the context of FIR, namely as a set of variables, a

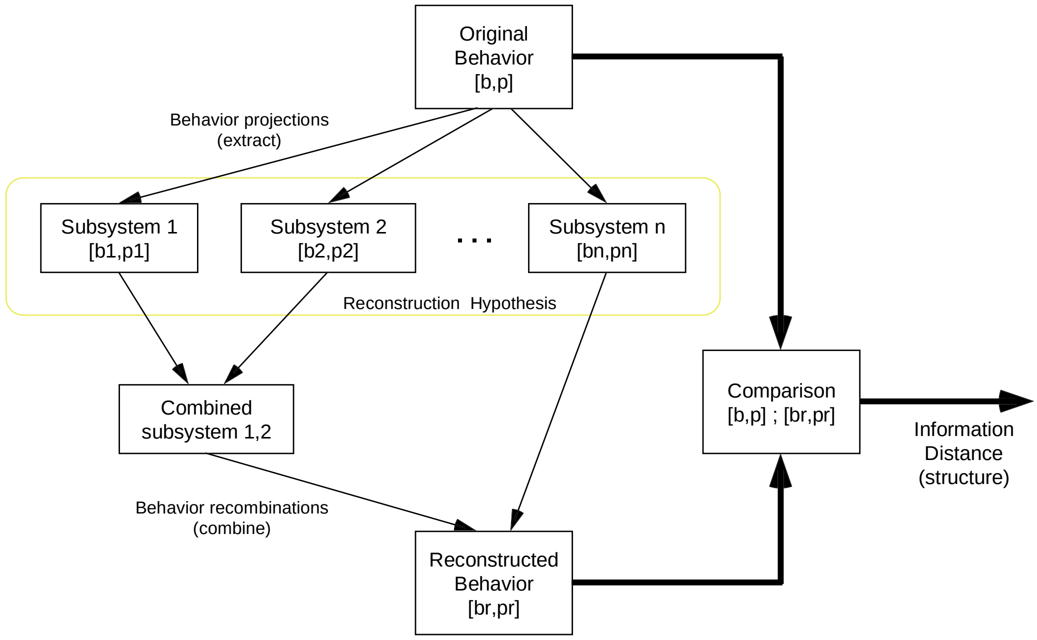


Figure 5.4: The reconstruction process.

behavior matrix that lists all qualitative states that the system can be in, and a probability (or confidence) vector that specifies the likelihood of occurrence of each state. A reconstruction hypothesis is a set of subsets of variables that are hypothesized to be used in each of the subsystems, i.e., a set of *possible causal structures* (subsystems).

The behavior of a subsystem is obtained by projecting the behavior of the overall system onto the subspace spanned by the selected variables of that subsystem. As a consequence, variables that are not included in the subsystem are eliminated from the behavior of the system, and states that can no longer be distinguished from each other are amalgamated into one, accumulating their individual probability (or confidence) values.

#### 5.2.2.1 Behavior Projections

How are the reconstruction hypotheses obtained, and how meaningful is a particular hypothesized model structure? To answer these questions, the quality of the proposed structure must be assessed relative to the observed behavior. This can be accomplished in many different ways. The Optimal

Mask Analysis starts out with a given output variable (it is normally known which among a set of variables is the desired output), and determines a set of "input" variables, i.e., a submodel, such that the input/output behavior of that submodel is as deterministic as possible. This makes a lot of sense, but the search for the optimal mask is expensive and can therefore not be used in the presence of too many variables. Reconstruction analysis goes a different way.

RA uses the same measurement data used by FIR, and the same recoding process. Let us remember how this process works. Let an observation, *obs*, of the behavior of a system be given in the form of a set of variables sampled equidistantly in time. The observation can be stored in a matrix where each column represents one variable, and each row represents one sampling instant. In Chapter 3 we called this matrix the *raw data matrix*. The variables in this matrix need to be *recoded* into discrete classes. The *basic behavior* of this observation can then be obtained using the command:

$$[b, p] \;=\; \text{behavior}\,(obs) \tag{5.10}$$

| | | |
|---|---|---|
| $b$ | = | denotes a numerically ordered list of individual records (rows) of the observation; |
| $p$ | = | denotes a numerically ordered list of the observation frequencies relative to the records of $b$. |

Vector $p$ can be reinterpreted as a probability vector. Let us suppose the following $b$ matrix and $p$ vector:

$$b = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad p = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.15 \\ 0.1 \\ 0.22 \\ 0.03 \\ 0.05 \\ 0.15 \end{bmatrix} \tag{5.11}$$

denote such a basic behavior. This is a five–variable system. Each variable is binary, thus, there are 32 possible different states (observations) in the system.

However, only eight of them have ever been observed. Their observation frequencies (probabilities of occurrence) are shown in vector $p$.

Let us assume that this behavior corresponds to the one of model $M$ described earlier in Figure 5.1. We can extract the behavior of the submodel $M_1$ by projecting the five–dimensional behavior space of the variables $v_1$, $v_2$, $v_3$, $v_4$, and $v_5$ (model $M$) onto the four–dimensional behavior space of the variables $v_1$, $v_2$, $v_3$, and $v_4$ (submodel $M_1$). The composite structure of model $M$ has been defined as $cs1$ in Equation 5.1, the composite structure of submodel $M_1$ is:

$$cs5 \;=\; [\;1\;2\;3\;4\;] \tag{5.12}$$

The projection to extract the submodel $M1$ behavior is accomplished using the function:

$$[b5, p5] \;=\; \text{extract}\,(b, p, cs5) \tag{5.13}$$

where the input variables stand for:

| | | |
|---|---|---|
| $b$ | = | is the behavior of model $M$; |
| $p$ | = | is the probability vector of model $M$; |
| $cs5$ | = | is the composite structure of submodel $M1$. |

And the output parameters stand for:

| | | |
|---|---|---|
| $b5$ | = | is the behavior of submodel $M1$; |
| $p5$ | = | is the probability vector of submodel $M1$. |

The result of this operation will be the four variable behavior matrix $b5$, and its correspondent probability vector $p5$:

$$
b5 = \left[ \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{array} \right] \qquad p5 = \left[ \begin{array}{c} 0.1 \\ 0.2 \\ 0.15 \\ 0.1 \\ 0.22 \\ 0.03 \\ 0.05 \\ 0.15 \end{array} \right] \tag{5.14}
$$

The projection onto the subspace of submodel $M2$ is made similarly by defining first its composite structure as:

$$
cs6 \;=\; [\ 2\ 3\ 4\ 5\ ] \tag{5.15}
$$

and then by extracting the behavior of submodel $M_2$:

$$
[b6, p6] \;=\; \text{extract}\,(b, p, cs6) \tag{5.16}
$$

The result of this operation is the four variable behavior matrix $b6$ and its correspondent probability vector $p6$:

$$
b6 = \left[ \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{array} \right] \qquad p6 = \left[ \begin{array}{c} 0.03 \\ 0.1 \\ 0.2 \\ 0.2 \\ 0.25 \\ 0.22 \end{array} \right] \tag{5.17}
$$

This result is a little more interesting since the number of states and the probability values are different from those of Behaviors (5.1) and (5.14), that correspond to model $M$ and submodel $M1$ respectively. The behavioral states of submodel $M2$ were extracted from the set of overall states, they were then rearranged in numerical order, and because of multiple occurrences of the same state, their observation frequencies (probabilities) were added.

### 5.2.3 Reconstruction from Subsystems

The behavior of the proposed subsystems in a reconstruction hypothesis can be recombined in such a way that the original overall system can be "reconstructed." The idea behind this procedure is the following: If the subsystem structure, i.e., the combination between all subsets of variables, that is proposed in the reconstruction hypothesis, corresponds to the real structure of the physical system under study, then the projections represent true behaviors of physical subsystems. In this case, it should be possible to obtain the observed overall system behavior as a reconstruction from the behaviors exhibited by its physical subsystems without much of an error.

#### 5.2.3.1 Behavior Recombinations

Let us look at the two projections obtained at the preceding section for submodels $M_1$ and $M_2$, and *reconstruct* a new five–dimensional space out of the knowledge that we still possess through these two projections in such a way that the projections of this new five–dimensional space onto the two four–dimensional spaces are the same as the projections from the original five–dimensional space.

This is accomplished by defining the overall composite structure of model $M$, i.e., Structure (5.1) once more as the wanted structure of the reconstructed system:

$$cs7 = [\ 1\ 2\ 3\ 4\ 0\ 2\ 3\ 4\ 5\ ] \tag{5.18}$$

and by combining the projections obtained for submodels $M1$ and $M2$ using the function:

$$[b7, p7] = \text{combine}(b5, p5, b6, p6, cs7) \tag{5.19}$$

where the output variables stand for:

$b7$ = is the behavior of the reconstructed system $cs7$;
$p7$ = is the probability vector of the reconstructed system $cs7$.

The result of this operation will be:

$$b7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad p7 = \begin{bmatrix} 0.0231 \\ 0.0769 \\ 0.2 \\ 0.15 \\ 0.1 \\ 0.22 \\ 0.0069 \\ 0.0231 \\ 0.05 \\ 0.15 \end{bmatrix} \tag{5.20}$$

The reconstructed behavior is calculated by means of the previously defined function *combine*, which combines the behavioral states of the two original behaviors $b5$ and $b6$, and calculates the probability of each reconstructed state as the conditional probability of that state in the first behavior $p5$ multiplied by the probability of the same state in the second behavior $p6$. For example, in the structure $< v_2, v_3, v_4 >$ that is common to both behaviors, the state

$$< v_2, v_3, v_4 > = < 1, 0, 0 > \tag{5.21}$$

occurs twice within the behavior $b5$, once with probability 0.1 and $v_1 = 0$, and once with probability 0.15 and $v_1 = 1$; and once within the behavior $b6$ with probability 0.25 and $v_5 = 0$. This means that in the combined behavior $b7$ there will be two states with $< v_2, v_3, v_4 >=< 1, 0, 0 >$, one with $v_1 = 0$, the other with $v_1 = 1$, and both with $v_5 = 0$. The conditional probability for the state

$$< v_1, v_2, v_3, v_4 > = < 0, 1, 0, 0 > \tag{5.22}$$

is determined by the fraction

$$\frac{0.1}{(0.1 + 0.15)} \tag{5.23}$$

and since the probability of the state

$$< v_2, v_3, v_4, v_5 > = < 1, 0, 0, 0 > \tag{5.24}$$

from behavior $b6$ is 0.25, the probability of the reconstructed state

$$< v_1, v_2, v_3, v_4, v_5 > = < 0, 1, 0, 0, 0 > \tag{5.25}$$

is

$$[\frac{0.1}{(0.1 + 0.15)}] \cdot 0.25 = 0.1 \tag{5.26}$$

In the same way, the probability of the second state derived from this combination

$$< v_1, v_2, v_3, v_4, v_5 > = < 1, 1, 0, 0, 0 > \tag{5.27}$$

is

$$[\frac{0.15}{(0.1 + 0.15)}] \cdot 0.25 = 0.15 \tag{5.28}$$

In the reconstruction, all the original states will reappear, but some additional states may appear as well. In the above reconstruction, two additional states are present. This can be seen by comparing Behavior (5.11) with Behavior (5.20).

The reconstruction carried out here is considered free of distortion since the two projections of the reconstructed five–dimensional space onto the two four–dimensional spaces spanned by the two submodels are indeed the same as for the original system, i.e., the same projections $[b5, p5]$ and $[b6, p6]$ are obtained both from $[b, p]$, and from $[b7, p7]$.

Unfortunately, this result does not extend to general reconstructions. If more than two submodels are present in the model structure, the reconstruction will have to be done sequentially. For example, let us consider the new structure $cs8$ with the same original behavior $[b, p]$ represented in Behavior (5.11):

$$cs8 \;=\; [\; 2\; 4\; 0\; 3\; 5\; 0\; 4\; 5\; 0\; 1\; 2\; 3\;] \tag{5.29}$$

that can be reconstructed following the next sequence of projections and recombinations. The projection for the first substructure in $cs8$, that can be defined as $cs8a$, can be obtained by specifying:

$$\begin{aligned} cs8a &= [\; 2\; 4\;] \\ [b8a, p8a] &= \text{extract}\,(b, p, cs8a) \end{aligned} \tag{5.30}$$

For the second substructure defined as $cs8b$:

$$\begin{aligned} cs8b &= [\; 3\; 5\;] \\ [b8b, p8b] &= \text{extract}\,(b, p, cs8b) \end{aligned} \tag{5.31}$$

for the third substructure defined as $cs8c$:

$$\begin{aligned} cs8c &= [\; 4\; 5\;] \\ [b8c, p8c] &= \text{extract}\,(b, p, cs8c) \end{aligned} \tag{5.32}$$

and for the fourth substructure defined as $cs8d$:

$$\begin{aligned} cs8d &= [\; 1\; 2\; 3\;] \\ [b8d, p8d] &= \text{extract}\,(b, p, cs8d) \end{aligned} \tag{5.33}$$

Then, the behavior recombinations must be done two by two, i.e., combining behavior $[b8a, p8a]$ with $[b8b, p8b]$ in a subreconstruction called $cs8e$:

$$\begin{aligned} cs8e &= [\ 3\ 5\ 0\ 2\ 4\ ] \\ [b8e, p8e] &= \text{combine}\,(b8b, p8b, b8a, p8a, cs8e) \end{aligned} \tag{5.34}$$

behavior $[b8c, p8c]$ with $[b8e, p8e]$ in a subreconstruction called $cs8f$:

$$\begin{aligned} cs8f &= [\ 4\ 5\ 7\ 0\ 2\ 3\ 4\ 5\ ] \\ [b8f, p8f] &= \text{combine}\,(b8c, p8c, b8e, p8e, cs8f) \end{aligned} \tag{5.35}$$

and behavior $[b8d, p8d]$ with $[b8f, p8f]$ in the final reconstruction called $cs8g$:

$$\begin{aligned} cs8g &= [\ 1\ 2\ 3\ 0\ 2\ 3\ 4\ 5\ ] \\ [b8g, p8g] &= \text{combine}\,(b8d, p8d, b8f, p8f, cs8g) \end{aligned} \tag{5.36}$$

The result of the whole operation, i.e., the reconstructed system, is:

$$
b8g = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad p8g = \begin{bmatrix} 0.1202 \\ 0.0930 \\ 0.0867 \\ 0.0776 \\ 0.0724 \\ 0.0564 \\ 0.0436 \\ 0.2200 \\ 0.0120 \\ 0.0093 \\ 0.0087 \\ 0.0259 \\ 0.0241 \\ 0.0846 \\ 0.0654 \end{bmatrix} \tag{5.37}
$$

Unfortunately, although the projections of the intermediate reconstruction are free of distortion, those of the reconstruction itself are no longer the same as those of the original space, and therefore, the final reconstruction contains a small amount of distortion.

### 5.2.4 Information Distance

Why are we at all interested in reconstructions? We want to identify true or at least plausible subsystems. Their reconstructions should exhibit behavioral patterns that are as similar as possible to those of the original system.

A distance function can be defined, namely the *Information Distance*, between the original and the reconstructed behavior, and if that distance function assumes a sufficiently small value, then the proposed structure represents a decent hypothesis for what the true structure of the physical system might look like. Thus, the information distance can be seen as a *loss of information measure.*

This distance function is computed in the following way. First, the original behavior is augmented by the additional reconstructed states. However, a probability value of 0.0 is assigned to these states in the original probability vector, since they have never been observed. Then, the distance function is computed as the $\mathcal{L}_2$–norm (the Euclidean norm) of the difference between the original and the reconstructed probability vectors.

The *structure* function:

$$err = \text{structure}\,(b, p, cs)$$

performs all the necessary projections and recombinations for a proposed composite structure, $cs$, and then computes the information distance function, $err$. Since normally reconstructions are not entirely free of distortions, the distance will also depend slightly on the sequence of the reconstructions, i.e., the same behavior applied to the same composite structure where the substructures are specified in a different sequence may lead to slightly (but not drastically) different values of the distance function.

For example, the *structure* function applied to the original behavior $[b, p]$ provided in Behavior (5.11) and to the structure $cs8$ defined in Structure (5.29) in the following way:

$$err8 = \text{structure}\,(b, p, cs8) \tag{5.38}$$

leads to a distance of $err9 = 0.3438$. However, if the substructures in $cs8$ are rearranged as shown in $cs9$:

$$cs9 = [1\ 2\ 3\ 0\ 3\ 5\ 0\ 2\ 4\ 0\ 4\ 5\,] \tag{5.39}$$

and the *structure* function is applied to the same behavior $[b, p]$ and to this new $cs9$ structure

$$err9 = \text{structure}\,(b, p, cs9) \tag{5.40}$$

a distance of $err9 = 0.3557$ is obtained. We had to search for quite some time to find an example with a difference in distance as large as in this example, due just to the sequence in which the different substructures are specified. However, it is claimed that this is quite alright. After all, we are performing a *qualitative* analysis only, and small variations in the obtained numerical values should not influence our decisions in any significant way.

#### 5.2.4.1 Correction of Distortions

Let us obtain the reconstructed behavior of the previous two examples $cs8$ and $cs9$ supposing that both structures have the original behavior $[b, p]$ defined in Behavior (5.11). The results will be $[b8, p8]$ and $[b9, p9]$. Since $b8 = b9$, they can be expressed by a single matrix:

$$b8 = b9 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad p8 = \begin{bmatrix} 0.1202 \\ 0.093 \\ 0.0867 \\ 0.0776 \\ 0.0724 \\ 0.0564 \\ 0.0436 \\ 0.22 \\ 0.012 \\ 0.0093 \\ 0.0087 \\ 0.0259 \\ 0.0241 \\ 0.0846 \\ 0.0654 \end{bmatrix} \quad p9 = \begin{bmatrix} 0.1148 \\ 0.0866 \\ 0.0837 \\ 0.0837 \\ 0.0809 \\ 0.0615 \\ 0.0464 \\ 0.1972 \\ 0.0115 \\ 0.0087 \\ 0.0084 \\ 0.0279 \\ 0.027 \\ 0.0922 \\ 0.0695 \end{bmatrix} \tag{5.41}$$

Seven additional states have been added in comparison with the original behavior $[b, p]$ shown in Behavior (5.11). The reconstructed states do not depend on the sequence of the reconstruction, only on the structure itself. However, the probability vectors may vary slightly depending on the sequence of the substructures.

It may sometimes be useful to guarantee a distortion–free reconstruction. Neither of the above reconstructions is distortion–free. In the above example, we know the reconstructed states. What is unknown are the correct values of the 15 probabilities associated with the reconstructed states. Let us call these probabilities $q_1$ to $q_{15}$.

What it is known in addition, for a distortion–free reconstruction, is that the projections of the reconstructed states are supposed to have the same probabilities as the projections of the original states. For example, the first substructure in $cs8$, namely $< v_2, v_4 >$ is equal to the state $< 0, 0 >$ ($< v_2, v_4 > = < 0, 0 >$) in the first, second, fourth, ninth, tenth, and twelfth reconstructed states of $b8$. It is also know that, in the original behavior $[b, p]$, this is true for the first, third, sixth, and seventh states. Adding up the corresponding probabilities from the original behavior, we obtain a value of 0.3332. Thus, we know that the sum should be:

$$q_1 + q_2 + q_4 + q_9 + q_{10} + q_{12} \;=\; 0.3332 \tag{5.42}$$

We can proceed similarly for all the other states of the same substructure, and for all the other substructures. In this way, 16 equations in 15 unknowns are obtained that can be written in a matrix form as:

$$M \cdot q \;=\; y \tag{5.43}$$

where $M$ and $y$ are known, and $q$ is unknown:

$$M = \left[\begin{array}{ccccccccccccccc}
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}\right] \qquad y = \left[\begin{array}{c}
0.33 \\ 0.2 \\ 0.47 \\ 0.28 \\ 0.3 \\ 0.42 \\ 0.28 \\ 0.52 \\ 0.2 \\ 0.3 \\ 0.15 \\ 0.1 \\ 0.22 \\ 0.03 \\ 0.05 \\ 0.15
\end{array}\right] \tag{5.44}$$

Since the rank of $M$ is nine, there are six degrees of freedom. Consequently, there are many ways for getting a distortion–free reconstruction. What can be done is to find a distortion–free reconstruction that has its probabilities as close as possible to the ones found by the recombination algorithm. In Matlab, this is done by augmenting the set of equations in the following way:

$$MM = [\ M\ ;\ \mathrm{eye}(15)\ ] \tag{5.45}$$
$$yy = [\ y\ ;\ p8\ ] \tag{5.46}$$

where *eye* is the Matlab command used to obtain the identity matrix, and $p8$ is the vector of probabilities found using the recombination algorithm, and solve the overdetermined set of 31 equations in a least square sense:

$$q = MM \backslash yy \tag{5.47}$$

The remaining distortion can now be computed as:

$$err = MM * q - yy \tag{5.48}$$
$$err = \mathrm{norm}(err(1:16), '\mathrm{inf}') \tag{5.49}$$

Clearly, the error is still not zero, i.e., the reconstruction is still not distortion–free. However, we can use the obtained $q$ vector as the new improved $p_1$ vector and keep iterating:

```
while  err > 1.0e-6,
   p8 = q
   yy = [ y ; p8 ];
   q = MM\yy;
   err = MM * q - yy;
   err = norm(err(1 : 16),'inf'),
end
```

until the error has decreased below the desired threshold. In the above example, nine iterations are required for convergence. The same procedure was repeated with $p9$ instead of $p8$. The resulting probability vectors were:

$$p8corr = [\ \begin{matrix} 0.1215 \\ 0.0901 \\ 0.0884 \\ 0.0753 \\ 0.0747 \\ 0.0585 \\ 0.0415 \\ 0.22 \\ 0.0133 \\ 0.0063 \\ 0.0104 \\ 0.0236 \\ 0.0264 \\ 0.0867 \\ 0.0633 \end{matrix}\ ] \qquad p9corr = [\ \begin{matrix} 0.1206 \\ 0.0898 \\ 0.0895 \\ 0.0751 \\ 0.0749 \\ 0.0589 \\ 0.0411 \\ 0.22 \\ 0.0128 \\ 0.0074 \\ 0.0097 \\ 0.0242 \\ 0.0258 \\ 0.0877 \\ 0.0623 \end{matrix}\ ] \tag{5.50}$$

The two reconstructions are still not identical. However, they are now both distortion–free.

A reconstruction of any composite structure *cs*, with correction for the distortion can be obtained using the *reconstruct*[3] function in the following way:

$$[brec, prec] \ = \ \text{reconstruct}\,(b, p, cs) \tag{5.51}$$

where *brec* is the behavior of the reconstructed system and *prec* its probability vector. However, since the correction is not normally needed and since it reduces the overall efficiency of the algorithm, the previously introduced *structure* command does not correct for distortions.

### 5.2.5 Fuzzy Reconstruction

Since the Fault Monitoring System proposed in Chapter 4 is based on the Fuzzy Inductive Reasoning methodology developed in Chapter 3, it will be normally the case that the *recoding* of an observed behavior will be *fuzzy* rather than *crisp*. The *fbehavior* function is the fuzzy equivalent of the crisp *behavior*

[3] Not yet implemented.

function presented in Equation (5.10), and is used to compute the possibilistic behavior of a fuzzy observation:

$$[b, c] \;=\; \text{fbehavior}\,(obs, conf)$$

where the input variables stand for:

| | | |
|---|---|---|
| $obs$ | $=$ | is the matrix of class values of the fuzzy observation; |
| $conf$ | $=$ | is the confidence vector associated with these observations. |

and the output variables stand for:

| | | |
|---|---|---|
| $b$ | $=$ | denotes a numerically ordered list of individual records (rows) of the observation; |
| $c$ | $=$ | accumulated confidences of the corresponding records of $b$. |

Usually, the confidence of a fuzzy observation is defined as the smallest of the fuzzy membership values associated with any of the variables within this observation.

Whereas $b$ is the same behavior matrix as in the crisp case, $c$ no longer denotes observation frequencies. Instead, this vector expresses the *accumulated confidence* in a particular observed state. If a state has been observed many times, the confidence in that state grows. Thus, the accumulated confidence in a given state is simply the sum of the confidences assigned to each individual previous observation of that state.

If a confidence vector already exists, it would be a pity to throw this valuable information away. Hence, it makes sense to look for a tool that can reason about structures related to possibilistic behaviors.

Let us use the same Model $M$ of Figure 5.1 with the same behavior $[b, p]$, but this time with the probability vector $p$ reinterpreted as a confidence vector $c$, i.e., Model $M$ has this time a possibilistic behavior instead of a probabilistic one:

$$b = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad c = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.15 \\ 0.1 \\ 0.22 \\ 0.03 \\ 0.05 \\ 0.15 \end{bmatrix} \tag{5.52}$$

Fuzzy capabilities were added to the RA methodology in order to deal with fuzzy behavior. The functions *fextract*, *fcombine*, and *fstructure*, are the fuzzy equivalent of the crisp functions *extract*, *combine*, and *structure* previously explained. Thus, the extraction of the possibilistic behavior of submodel $M_1$ is made exactly in the same way as in the crisp case, except for the use of *fextract* instead of *extract*. From Structure (5.15), we know that $cs6$ is:

$$cs6 = [\ 2\ 3\ 4\ 5\ ] \tag{5.53}$$

Thus the fuzzy projection onto the space spanned by the variables $v_2$, $v_3$, $v_4$, and $v_5$, is obtained with:

$$[b6, c6] = \text{fextract}\,(b, c, cs6) \tag{5.54}$$

And the result is:

$$b6 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \qquad c6 = \begin{bmatrix} 0.03 \\ 0.1 \\ 0.2 \\ 0.15 \\ 0.15 \\ 0.22 \end{bmatrix} \tag{5.55}$$

We chose the same example as in the crisp case so that the results of crisp and fuzzy reconstruction analysis can be compared more easily with each other. Clearly, it is always possible to reinterpret a probabilistic behavior as possibilistic, although the reverse is not true. The difference becomes evident after the projection, since the confidences of the projection no longer add up to one.

Similarly, a projection onto the subspace $< v_1, v_2, v_3, v_4 >$ is accomplished using the $cs5$ structure given in Structure (5.12):

$$cs5 \;=\; [\;1\;2\;3\;4\;] \tag{5.56}$$

and the function:

$$[b5, c5] \;=\; \mathrm{fextract}\,(b, c, cs5) \tag{5.57}$$

The result of this operation is:

$$b5 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \qquad c5 = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.15 \\ 0.1 \\ 0.22 \\ 0.03 \\ 0.05 \\ 0.15 \end{bmatrix} \tag{5.58}$$

It would have made perfect sense to compute the confidences of the projection by accumulating the confidences of individual multiple observations. However, in accordance with the theoretical work of [Cavallo and Klir, 1982], this was not done. Instead, the confidence of the projection of a given state is simply the largest among the confidences of the previous observations of that state.

A recombination (or join) of two projections is accomplished by combining the states in the same manner as was done in the crisp case. However, the confidence in the join is computed as the smallest among the individual confidences. Let us take the original composite structure $cs7$ of Model $M$ proposed in Structure (5.18):

$$cs7 \;=\; [\;1\;2\;3\;4\;0\;2\;3\;4\;5\;] \tag{5.59}$$

and combine the fuzzy behaviors of submodels $M1$ and $M2$ with the *fcombine* function:

$$[b7, c7] \;=\; \text{fcombine}(b5, c5, b6, c6, cs7) \tag{5.60}$$

The result of this operation will be:

$$b7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad c7 = \begin{bmatrix} 0.03 \\ 0.1 \\ 0.2 \\ 0.15 \\ 0.1 \\ 0.22 \\ 0.03 \\ 0.03 \\ 0.05 \\ 0.15 \end{bmatrix} \tag{5.61}$$

However, and contrary to the crisp reconstructions, fuzzy reconstructions are distortion–free across multiple sequential reconstructions. Thus, no distortion correction algorithm is needed as was true for the crisp case.

Once more in accordance with the theoretical work of [Cavallo and Klir, 1982], a different distance measure was used to determine the fuzzy reconstruction error, namely through the *Ambiguity Measure*.

The *ambiguity* of the original behavior $[b, c]$ (Behavior (5.52)) is computed as:

$$a = 1.0 - \left(\frac{0.03}{8} + \frac{0.02}{7} + \frac{0.05}{6} + \frac{0.05}{4} + \frac{0.05}{2} + \frac{0.02}{1}\right) = 0.9276 \tag{5.62}$$

0.03 is the smallest confidence value found in the confidence vector. All eight observed states have a confidence larger or equal than 0.03. This is how the first term between the parentheses is computed. The next higher confidence is 0.05. The difference to the previous one is 0.02. There are seven confidence values in the vector that are larger or equal to 0.05. This determines the next term, etc. It is easier to explain how the ambiguity is computed by means of an example than through a general formula. So, this way was chosen here. The ambiguity of the reconstructed behavior is computed as:

$$a7 = 1.0 - \left(\frac{0.03}{10} + \frac{0.02}{7} + \frac{0.05}{6} + \frac{0.05}{4} + \frac{0.05}{2} + \frac{0.02}{1}\right) = 0.9283 \qquad (5.63)$$

The distance function then is the difference between the ambiguities of the reconstructed and original behaviors:

$$err7 \;=\; a7 - a \;=\; 0.00075 \qquad (5.64)$$

This same error can be obtained directly using the function *fstructure* that performs all the necessary projections and recombinations for the composite structure $cs7$, and then computes the distance function error $err7$:

$$err7 \;=\; \text{fstructure}\,(b, c, cs7) \qquad (5.65)$$

Why do we use this strange distance measure? The distance measure used should be defined in accordance with the projection and recombination rules in such a way that, when a probabilistic behavior is reinterpreted as possibilistic behavior, and when a suboptimal structure is searched explaining the observed behavior, the two approaches (probabilistic and possibilistic) should lead to similar ordering sequences of structures, and thereby to the selection of a similar suboptimal structure. The distance measure proposed above satisfies this criterion.

### 5.2.6 Optimal Structure Analysis

How do we go about determining the most suitable structure hypothesis? The determination of the most suitable reconstruction hypothesis is a search problem. Enumerating all possible structure candidates is much too expensive, hence suboptimal search strategies must be found. Three such techniques were proposed in [Uyttenhove, 1978] and elaborated upon in [Cellier, 1991a], and all three have been implemented in RA. They all operate on binary structures rather than on composite structures. These three suboptimal search strategies are:

i) Structure Refinement.

ii) Structure Aggregation.

iii) Single Refinement.

The RA function that carries out the suboptimal search using the crisp behavior is:

$$cs \;=\; \text{optstruc}\,(b, p, qmin, group,'\text{algorithm}') \qquad (5.66)$$

and in the case of fuzzy behavior:

$$cs \;=\; \text{foptstruc}\,(b, c, qmin, group,'\text{algorithm}') \qquad (5.67)$$

where the input variables stand for:

| | | |
|---|---|---|
| $b$ | = | behavior the system; |
| $p$ | = | probability vector relative to behavior $b$; |
| $c$ | = | accumulated confidences relative to behavior $b$. |
| $qmin$ | = | smallest tolerated quality; |
| $group$ | = | is a grouping parameter provided by the modeler; |
| $algorithm$ | = | denotes the search strategy. |

and the output variable $cs$ is the suboptimal structure found. The parameter $group$ can be explained by means of an example. If a four variable system is grouped as:

$$group \;=\; [\; 1\; 2\; 3\; 0\; 1\; 4\; ] \qquad (5.68)$$

this instructs the optimization algorithm that, in a given six–variable system, the first and fifth variable must appear in any subsystem either together or not at all, whereas the fourth variable must not appear in any subsystem. If no *a priori* grouping knowledge exists, *group* should be coded as:

$$group \;=\; 1 : n \tag{5.69}$$

where $n$ denotes the number of variables in the system.

#### 5.2.6.1 Structure Refinement

It starts out with the totally connected binary structure, i.e., with a model without internal structure, or, which is saying the same, a composite structure with just one subsystem that includes the same variables as the original system. Evidently, the reconstruction error is zero for this structure, since there is nothing to be reconstructed.

Then one binary connection is severed at a time, and the reconstruction error is calculated in every case. In an $n$–variable system, this leads to:

$$\frac{n\,(n-1)}{2} \tag{5.70}$$

different structures to be investigated. The one with the smallest reconstruction error among these candidate structures is found as a good working hypothesis. The binary connection is thus permanently removed, and the search process continues by removing an additional binary connection from the remaining:

$$\frac{n\,(n-1)}{2} - 1 \tag{5.71}$$

candidate connections. The search stops when the smallest reconstruction error (information distance) within one set of candidate structures has become larger than the largest tolerated error determined by the modeler.

Let us call the error of the totally unconnected model $err_{\rm unc}$. We define the *structure quality* $Q_s$ as follows:

$$Q_S \;=\; 1.0 \;-\; \frac{err}{err_{\rm unc}} \tag{5.72}$$

$Q_S$ is equal to 1.0 for the totally connected model and equal to 0.0 for the totally unconnected model. The structure quality is more useful than the reconstruction error, since it is normalized. The modeler is thus asked to specify the smallest tolerated quality, $q_{\rm min}$, rather than the largest tolerated error, $err_{\rm max}$.

Let us apply the *optstruc* function with the Structure Refinement search option, i.e., *refine*, to the previously defined behavior of Behavior (5.11):

$$cs \;=\; \text{optstruc}\,(b, p, qmin, group,'\text{refine}') \tag{5.73}$$

with:

$$\begin{aligned} q_{min} &= 0.1 \\ group &= 1:5 \end{aligned}$$

The *optstruc* function gives results at each level of refinement, as can be seen in the next print–out:

```
STRUCTURE (Q = 1.0000)  (1,2,3,4,5)

          BEST STRUCTURE ON THIS LEVEL:   (1,2,3,4,5)
          THE BEST QUALITY IS   1.0000
          SEARCH IS CONTINUED WITH THIS STRUCTURE


STRUCTURE (Q = 0.8712)  (1,3,4,5) (2,3,4,5)
STRUCTURE (Q = 0.9419)  (1,2,4,5) (2,3,4,5)
STRUCTURE (Q = 1.0000)  (1,2,3,5) (2,3,4,5)
STRUCTURE (Q = 0.9195)  (1,2,3,4) (2,3,4,5)
STRUCTURE (Q = 0.8367)  (1,2,4,5) (1,3,4,5)
STRUCTURE (Q = 1.0000)  (1,2,3,5) (1,3,4,5)
STRUCTURE (Q = 0.8256)  (1,2,3,4) (1,3,4,5)
STRUCTURE (Q = 0.7674)  (1,2,3,5) (1,2,4,5)
STRUCTURE (Q = 0.6947)  (1,2,3,4) (1,2,4,5)
```

```
STRUCTURE (Q = 1.0000)  (1,2,3,4) (1,2,3,5)

          BEST STRUCTURE ON THIS LEVEL:   (1,2,3,5) (2,3,4,5)
          THE BEST QUALITY IS   1.0000
          SEARCH IS CONTINUED WITH THIS STRUCTURE


STRUCTURE (Q = 0.8712)  (1,3,5) (2,3,4,5)
STRUCTURE (Q = 0.8953)  (1,2,5) (2,3,4,5)
STRUCTURE (Q = 0.9048)  (1,2,3) (2,3,4,5)
STRUCTURE (Q = 0.8199)  (1,2,5) (1,3,5) (2,4,5) (3,4,5)
STRUCTURE (Q = 1.0000)  (3,4,5) (1,2,3,5)
STRUCTURE (Q = 0.7764)  (1,2,3) (1,3,5) (2,3,4) (3,4,5)
STRUCTURE (Q = 0.7209)  (2,4,5) (1,2,3,5)
STRUCTURE (Q = 0.6576)  (1,2,3) (1,2,5) (2,3,4) (2,4,5)
STRUCTURE (Q = 0.9366)  (2,3,4) (1,2,3,5)

          BEST STRUCTURE ON THIS LEVEL:   (3,4,5) (1,2,3,5)
          THE BEST QUALITY IS   1.0000
          SEARCH IS CONTINUED WITH THIS STRUCTURE


STRUCTURE (Q = 0.8712)  (1,3,5) (2,3,5) (3,4,5)
STRUCTURE (Q = 0.8953)  (1,2,5) (2,3,5) (3,4,5)
STRUCTURE (Q = 0.9048)  (1,2,3) (2,3,5) (3,4,5)
STRUCTURE (Q = 0.6563)  (1,2,5) (1,3,5) (3,4,5)
STRUCTURE (Q = 0.7383)  (1,2,3) (1,3,5) (3,4,5)
STRUCTURE (Q = 0.5930)  (4,5) (1,2,3,5)
STRUCTURE (Q = 0.5267)  (3,4) (4,5) (1,2,3) (1,2,5)
STRUCTURE (Q = 0.6631)  (3,4) (1,2,3,5)

          BEST STRUCTURE ON THIS LEVEL:   (1,2,3) (2,3,5) (3,4,5)
          THE BEST QUALITY IS   0.9048
          SEARCH IS CONTINUED WITH THIS STRUCTURE


STRUCTURE (Q = 0.5838)  (1,3) (2,3,5) (3,4,5)
STRUCTURE (Q = 0.5970)  (1,2) (2,3,5) (3,4,5)
STRUCTURE (Q = 0.3743)  (1,2) (1,3) (2,5) (3,4,5)
STRUCTURE (Q = 0.4947)  (1,2,3) (3,4,5)
STRUCTURE (Q = 0.5322)  (4,5) (1,2,3) (2,3,5)
STRUCTURE (Q = 0.3303)  (2,5) (3,4) (4,5) (1,2,3)
STRUCTURE (Q = 0.6156)  (3,4) (1,2,3) (2,3,5)

          BEST STRUCTURE ON THIS LEVEL:   (3,4) (1,2,3) (2,3,5)
          THE BEST QUALITY IS   0.6156
          SEARCH IS CONTINUED WITH THIS STRUCTURE


STRUCTURE (Q = 0.4695)  (1,3) (3,4) (2,3,5)
STRUCTURE (Q = 0.4297)  (1,2) (3,4) (2,3,5)
```

```
STRUCTURE (Q = 0.2934)  (1,2) (1,3) (2,5) (3,4) (3,5)
STRUCTURE (Q = 0.4105)  (3,4) (3,5) (1,2,3)
STRUCTURE (Q = 0.4640)  (1,2,3) (2,3,5) (4)
STRUCTURE (Q = 0.2629)  (2,5) (3,4) (1,2,3)

          BEST STRUCTURE ON THIS LEVEL:   (1,3) (3,4) (2,3,5)
          THE BEST QUALITY IS   0.4695
          SEARCH IS CONTINUED WITH THIS STRUCTURE


STRUCTURE (Q = 0.4544)  (3,4) (2,3,5) (1)
STRUCTURE (Q = 0.2994)  (1,3) (2,5) (3,4) (3,5)
STRUCTURE (Q = 0.2925)  (1,3) (2,3) (3,4) (3,5)
STRUCTURE (Q = 0.3475)  (1,3) (2,3,5) (4)
STRUCTURE (Q = 0.1769)  (1,3) (2,3) (2,5) (3,4)

          STRUCTURE ON THIS LEVEL:   (3,4) (2,3,5) (1)
          THE BEST QUALITY IS   0.4544
          SEARCH IS CONTINUED WITH THIS STRUCTURE


STRUCTURE (Q = 0.2958)  (2,5) (3,4) (3,5) (1)
STRUCTURE (Q = 0.3021)  (2,3) (3,4) (3,5) (1)
STRUCTURE (Q = 0.2904)  (2,3,5) (1) (4)
STRUCTURE (Q = 0.1986)  (2,3) (2,5) (3,4) (1)

          BEST STRUCTURE ON THIS LEVEL:   (2,3) (3,4) (3,5) (1)
          THE BEST QUALITY IS   0.3021
          SEARCH IS CONTINUED WITH THIS STRUCTURE


STRUCTURE (Q = 0.2726)  (3,4) (3,5) (1) (2)
STRUCTURE (Q = 0.2109)  (2,3) (3,5) (1) (4)
STRUCTURE (Q = 0.1083)  (2,3) (3,4) (1) (5)

          BEST STRUCTURE ON THIS LEVEL:   (3,4) (3,5) (1) (2)
          THE BEST QUALITY IS   0.2726
          SEARCH IS CONTINUED WITH THIS STRUCTURE


STRUCTURE (Q = 0.2232)  (3,5) (1) (2) (4)
STRUCTURE (Q = 0.1129)  (3,4) (1) (2) (5)

          BEST STRUCTURE ON THIS LEVEL:   (3,5) (1) (2) (4)
          THE BEST QUALITY IS   0.2232
          SEARCH IS CONTINUED WITH THIS STRUCTURE


STRUCTURE (Q = 0.0000)  (1) (2) (3) (4) (5)

     BEST STRUCTURE ON THIS LEVEL HAS A QUALITY OF 0.0000
```

```
    WHICH IS CONSIDERED TOO LOW, SEARCH IS TERMINATED


 TOTAL RESULTING STRUCTURE:  (3,5) (1) (2) (4)
```

This result has the following meaning: while variables 3 and 5 must be considered always together, variables 1, 2, and 4 can be considered independently.

The importance of this result can be seen if for example, variable 5 is the output of this five variable system; in that case, variables 1, 2, and 4 can be neglected because the only variable related with the output is variable 3. Thus the system is reduced to a two variable system, namely: variables 3 and 5.

Let us see what happens if the fuzzy option *foptstruct* is applied to the same example. In this case the probability vector $p$ is reinterpreted as the accumulated confidence vector $c$.

```
 TOTAL RESULTING STRUCTURE:  (3,4) (1) (2) (5)
```

A different result has been obtained. This is probably due to the small value given to the smallest tolerated quality.

#### 5.2.6.2 Structure Aggregation

This strategy is the dual to the former one. This time, the search starts with the totally unconnected binary structure, i.e., a composite structure in which each single variable corresponds to a subsystem, that is, a structure with zero quality. Evidently, the reconstruction error is very large. One binary connection is added at a time. Among the $n(n-1)/2$ candidate structures with a single binary connection, the one that reduces the reconstruction error the most is chosen, i.e., the one that offers the highest quality. This binary connection is thus made permanent, and a second one is added. The search continues until the quality has increased to a value that is larger than the smallest tolerated quality determined by the modeler.

Evidently, since the Structure Refinement and the Structure Aggregation algorithms are suboptimal search strategies, they will not necessarily lead to the same conclusion.

Let us apply the *optstruc* function with the Structure Aggregation suboptimal search strategy i.e., *aggregate*, to the same example:

$$cs \;=\; \text{optstruc}\,(b, p, qmin, group, '\text{aggregate}') \tag{5.74}$$

As in the case of the previous algorithm, the *optstruc* function gives a complete print–out of each refinement step:

```
STRUCTURE (Q = 0.0000)  (1) (2) (3) (4) (5)

          BEST STRUCTURE ON THIS LEVEL:   (1) (2) (3) (4) (5)
          THE BEST QUALITY IS   0.0000
          SEARCH IS CONTINUED WITH THIS STRUCTURE


STRUCTURE (Q = -.0211)  (1,2) (3) (4) (5)
STRUCTURE (Q = 0.0255)  (1,3) (2) (4) (5)
STRUCTURE (Q = 0.0133)  (1,4) (2) (3) (5)
STRUCTURE (Q = 0.0690)  (1,5) (2) (3) (4)
STRUCTURE (Q = -.0077)  (2,3) (1) (4) (5)
STRUCTURE (Q = 0.0618)  (2,4) (1) (3) (5)
STRUCTURE (Q = 0.1082)  (2,5) (1) (3) (4)
STRUCTURE (Q = 0.1129)  (3,4) (1) (2) (5)
STRUCTURE (Q = 0.2232)  (3,5) (1) (2) (4)
STRUCTURE (Q = 0.0201)  (4,5) (1) (2) (3)

     BEST STRUCTURE ON THIS LEVEL HAS A QUALITY OF 0.2232
     WHICH IS CONSIDERED LARGE ENOUGH
     SEARCH IS TERMINATED


TOTAL RESULTING STRUCTURE:   (3,5) (1) (2) (4)
```

This result coincides with the one obtained by the aforementioned Structure Refinement algorithm, having the same interpretation. Applying the same algorithm with the fuzzy option, the following result is obtained:

```
TOTAL RESULTING STRUCTURE:  (3,5) (1) (2) (4)
```

which is once again the same result as previously obtained.

Please, notice that $Q_s$ is *not* a *quality measure* [Cellier *et al.*, 1996a] in the strict sense, since the totally unconnected model does not necessarily have the poorest quality of all, as the above example shows.

#### 5.2.6.3 Single Refinement

This search strategy is the coarsest but also cheapest one. This algorithm starts out like the structure refinement algorithm, that is, with the totally connected binary structure, but rather than severing only one binary connection at a time, all binary connections with a sufficiently small error (determined by the modeler) are removed at once, and the algorithm stops after investigating the first $n(n-1)/2$ candidate structures.

Here, it doesn't make sense to talk about a minimal tolerated quality. Instead, the user should specify the largest tolerated error resulting from a single severed binary connection.

Let us apply the *optstruc* function with the Single Refinement suboptimal search strategy i.e., *singleref*, to the same example:

$$cs \;=\; \text{optstruc}\,(b, p, ermax, group, '\text{singleref}') \tag{5.75}$$

with $ermax = 0.01$

In this case, the *optstruc* function gives a complete print–out of all binary relations included, and their related omission error.

```
STATISTICS ON BINARY RELATIONS:
*******************************

     ERROR        BINARY RELATION
     STRENGTH     OMITTED
     --------     ---------------
     0.000000     (   1,   4)
     0.000000     (   2,   4)
     0.000000     (   4,   5)
     0.033333     (   1,   3)
     0.046154     (   1,   5)
     0.073810     (   1,   2)
     0.093617     (   2,   3)
     0.100000     (   2,   5)
```

```
     0.133333    (   3,   4)
     0.175000    (   3,   5)

FROM THE ABOVE LIST
THE 7 LOWERMOST RELATIONS ARE CONSIDERED IMPORTANT


TOTAL RESULTING STRUCTURE:  (3,4) (1,2,3,5)
```

Since the maximum tolerated error is 0.01, the first three binary relations can be neglected, whereas the other seven, with an associated error greater than 0.01 must be considered. Applying the same *singleref* option to the fuzzy case, the resulting structure is:

```
TOTAL RESULTING STRUCTURE:  (3,4) (1,2,3,5)
```

which is the same result as obtained with the crisp option. Had we chosen $ermax = 0.15$, the same structure would have resulted as with the *refine* and *aggregate* options, as can be seen from the above list of statistics on binary relations.

If we assume once again that variable 5 is the output of the system whereas all other variables are inputs, the obtained structure indicates that variable 4 can be neglected because it has no relation with the output, whereas variables 1, 2, and 3 must be considered. The original five variable system has become a four variable system.

Variations in the maximum tolerated error may lead to variations in the resulting structure, thus the way in which this error is chosen is a delicate matter.

The single refinement technique is particularly attractive due to its much reduced cost. The computational complexity of the single refinement algorithm is polynomial and of the order of $n^2$, where $n$ denotes the number of variables in the system, whereas the computational complexity of the two other search algorithms is exponential and of the order $2^n$. It turns out that for our purpose, i.e., for selecting a subset of variables to reason with, the single refinement algorithm is always the method of choice.

#### 5.2.6.4 Postoptimization

Finally, there is a way of postoptimizing the found suboptimal structures. The postoptimization function is:

$$cspopt \;=\; \text{singleref}\,(cs, b, p, ermax) \tag{5.76}$$

It performs the Single Refinement algorithm on each of the substructures found by any of the three suboptimal search strategies. For example, the command sequence:

$$\begin{aligned} cs11a &= \text{optstruc}\,(b, p, qmin, group,' \text{refine}') \\ cs11b &= \text{singleref}\,(cs11a, b, p, ermax) \end{aligned}$$

finds first a suboptimal structure, $cs11a$, using the structure refinement algorithm, and then looks at each of the submodels of $cs11a$ separately, and performs the single refinement algorithm on it. The final structure, $cs11b$, is the combination of all these refinements. Consequently, $cs11b$ may have less binary connections that $cs11a$, but never more.

If selection of the correct structure is critical, it may be a good idea to repeat the search with the three algorithms to check whether the same or similar suboptimal structures are obtained. If this is not the case initially, grouping may help to force the search algorithms to come up with more similar solutions. In this way, the user will gain more and more confidence in the structures proposed by the suboptimal search techniques.

## 5.3 Conclusions

In this chapter, the implementation of the Reconstruction Analysis methodology has been shown as a tool for subsystem identification and variable selection through causality analysis and refinement procedures. In FIR terms, a tool for selecting the minimum set of meaningful variables to reason with at any new situation has been found.

Reconstruction Analysis has proven to be the right GSPS level 4 tool to be combined with FIR. Some of its properties are:

- We have implemented RA in such a way that it is fully compatible with the previously developed Fuzzy Inductive Reasoning methodology.
- Some functions of FIR such as *recode*, *behavior*, and *fbehavior* are also used in RA.
- FIR recodes the original behavior trajectories and feeds RA with this data. RA, by means of the Optimal Structure Analysis techniques, searches for suboptimal structures that can be considered as subsystems. Finally, RA feeds FIR with these subsystems, i.e., sets of variables, which, in turn, will make qualitative models by means of the Optimal Mask Analysis, and carry out the reasoning process.
- Since both methodologies (FIR and RA) use the same recoded information and complement each other in the way described in the previous step, they can be considered as two facets of one and the same methodology. Both tools are available in SAPS–II.

In the next two chapters, it will be shown how these two methodologies are combined to construct the Fault Monitoring System described in Chapter 4, and how the combined methodology is applied to large–scale systems.

# Chapter 6

# Selection and Causal Grouping of Variables Using RA

## 6.1 Introduction

In the preceding chapter, the Reconstruction Analysis methodology has been explained in full, and an example was provided to enforce the understanding of this GSPS[1] level 4 tool. The need for such a tool was stated on the basis of the selection of minimum sets of meaningful variables that can be isolated as subsystems, among the large set of variables inherent in any large–scale system.

It could be useful at this point to remember that Reconstruction Analysis uses the Single Refinement algorithm of the Optimal Structure Analysis for the selection and causal grouping of variables. Two considerations were taken into account in order to decide the implementation of the RA methodology: a) the GSPS level 3 technique used by the FIR methodology to obtain qualitative models, i.e., Optimal Mask Analysis, may not be efficient when dealing with level 5 problems, and b) Optimal Mask Analysis, and the whole FIR methodology, suffice to deal with problems of fault monitoring in small– to medium–sized systems, but not in large–scale systems, in which the number

[1]A detailed description of the General Systems Problem Solver epistemological levels is provided in Chapter 3.

of variables and subsystems is larger and may vary with time.

Thus, the goal of the RA methodology is that of providing the Fault Monitoring System based on Fuzzy Inductive Reasoning with the subsets of variables to reason with. This means that RA must be capable of identifying clusters of causally related variables with the following characteristics:

- Each cluster constitutes an identified subsystem, and should be composed of one or several inputs and a single output.
- From each cluster, it should be possible to obtain an optimal mask. Thus, the number of subsystem variables should be kept inside the FIR methodology limits.
- Clusters should be causally related among each other, i.e., form a causal temporal hierarchy.

Some issues need to be addressed before RA can deliver this kind of results. These are:

i) Since Reconstruction Analysis is intended to perform temporal causality analysis, the concept of *time* must be included somehow.

ii) Some heuristic recipes must be proposed to introduce the concepts of input and output variables in Optimal Structure Analysis, and to interpret the relationships between input and output variables obtained by the Single Refinement algorithm.

iii) The third issue addresses the management of the large number of variables in a large–scale system. Just like FIR, also the RA methodology cannot deal with a large number of variables at the same time. This means that the original system needs to be decomposed into several subsystems in order to facilitate the refinement procedures of the Optimal Structure Analysis.

iv) A comparison between the input/output relationships obtained by *Optimal Structure Analysis* and *Correlation Analysis* on the one hand and by *Optimal Mask Analysis* on the other will be provided for short– and medium–sized systems in order to gain confidence that, if the results obtained are similar, then it can be inferred that the Optimal Structure Analysis will likely obtain reasonable results also when applied to large–scale systems.

Thus in this chapter, the problems related with the management of large numbers of variables, and the causal grouping between them will be tackled. Some of these problems will be directly solved by applying the RA methodology and interpreting its results, whereas others will be solved by means of heuristic recipes specially developed for such purposes.

## 6.2 Including Time in the RA Methodology

The original *Reconstructability Analysis*[2] methodology does not consider the passing of time. However, since in this dissertation it is intended to apply Reconstruction Analysis to the problem of *temporal causality analysis*, the variable *time* has to be incorporated into the methodology.

This is accomplished in the following way. Assuming that the dynamics of a system to be modeled can be captured covering $m$ sampling intervals or a time span of $m \cdot \Delta t$ time units, which is equivalent to saying that the system can be qualitatively represented by an optimal mask[3] of depth $(m+1)$; it is possible to duplicate the raw data matrix of the system $m$ times, and concatenate the duplicates to the original behavior trajectory matrix from the right, each of them shifted up by one row in comparison to the previous one.

Let us take as an example the linear system presented in Section 3.4. This input/output system is composed of one input $u_1$ and three outputs $y_1$, $y_2$, and $y_3$ that have already been *recoded* into three qualitative classes with a slowest time constant that can be covered by two sampling intervals, i.e., a mask of depth three had been chosen before. The matrix of class values (raw data matrix) for this system looks as follows:

$$
\begin{array}{c}
{}_t\backslash^x \\
0.0 \\
\delta t \\
2\cdot\delta t \\
3\cdot\delta t \\
4\cdot\delta t \\
\vdots \\
m\cdot\delta t
\end{array}
\begin{array}{c}
\begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\
\begin{pmatrix}
1 & 2 & 3 & 2 \\
1 & 3 & 3 & 2 \\
3 & 1 & 3 & 1 \\
3 & 1 & 3 & 3 \\
1 & 2 & 1 & 3 \\
\vdots & \vdots & \vdots & \vdots \\
3 & 3 & 2 & 3
\end{pmatrix}
\end{array}
\tag{6.1}
$$

[2]Reconstructability Analysis is composed of two stages: *Identification* and *Reconstruction*. In this dissertation, only the Reconstruction stage is used. Further details on Reconstructability Analysis can be found in [Cavallo and Klir, 1979; 1981; Klir, 1985a].

[3]See the Optimal Mask Analysis section in Chapter 3.

Thus, the expanded behavior trajectory matrix will look like:

$$\begin{array}{c} {}_t\backslash^x \\ 0.0 \\ \delta t \\ 2\cdot\delta t \\ 3\cdot\delta t \\ 4\cdot\delta t \\ \vdots \end{array}
\begin{array}{c} \begin{array}{cccccccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 & y_1 & y_2 & y_3 \end{array} \\
\begin{pmatrix} & & & & & & & & 1 & 2 & 3 & 2 \\ & & & & 1 & 2 & 3 & 2 & 1 & 3 & 3 & 2 \\ 1 & 2 & 3 & 2 & 1 & 3 & 3 & 2 & 3 & 1 & 3 & 1 \\ 1 & 3 & 3 & 2 & 3 & 1 & 3 & 1 & 3 & 1 & 3 & 3 \\ 3 & 1 & 3 & 1 & 3 & 1 & 3 & 3 & 1 & 2 & 1 & 3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \end{array} \quad (6.2)$$

This expanded matrix contains $m$ times as many variables as the original matrix, but in return, an optimal mask of depth 1 can be found that is equivalent to the formerly used optimal mask of depth $(m + 1)$. The new optimal mask matrix has the dimensions $1 \times (n \cdot m)$ instead of $m \times n$ as before, where $n$ is the number of variables (columns) of the original behavior trajectory matrix. For the example of Matrix (6.2) the translation rule between the old and the new optimal masks is:

$$\begin{array}{l} t-2\delta t \\ t-\delta t \\ t \end{array}
\begin{pmatrix} u_1 & u_2 & u_3 & u_4 \\ u_5 & u_6 & u_7 & u_8 \\ u_9 & y_1 & y_2 & y_3 \end{pmatrix} \quad (6.3)$$

In the modified system, *time* is no longer an issue. Consequently, the Reconstruction Analysis technique can be applied to the behavior of this much larger system, and the temporal causality analysis can be performed as described in the previous chapter.

## 6.3 Management of Large Numbers of Variables

The variable selection process is carried out by using the *Single Refinement* algorithm of the RA methodology in the same way it has been mentioned in Chapter 5. The purpose is to find strong qualitative (non–linear) correlations between inputs and outputs, and weak correlations among inputs, in such a way that the resulting relations for the outputs will be as deterministic as possible while avoiding unnecessary redundancy among highly correlated inputs.

The first step of this process is the computation of the system time constants from the input/output behavior as it was shown in Chapter 3. In this way, a mask depth is obtained that assures that the system dynamics will be covered by the resulting optimal mask. Then, the number of original variables (one input and three output variables, i.e., a total of four variables in the above example) is multiplied by the mask depth as it was explained in the previous section. The resulting number is the new number of variables (nine input and three output variables, i.e., a total of 12 variables in the given example) that must be considered in the Reconstruction Analysis.

The second step is the decomposition of the overall system (with the new number of variables) into sets of inputs and outputs that must include all possible binary relations between the variables. This decomposition is necessary because the new number of variables is much larger than the original one, and it has already been said that RA is not capable of managing too many variables at a time.[4]

Let us revisit the one input, three outputs example of the preceding section. Of course, this example can be managed by the RA methodology without needing any decomposition, but for comparison purposes, it will be useful to show how the decomposition is carried out. First, one output variable must be selected. As in the FIR methodology, RA treats each output separately. With each of the three outputs we will do the following:

i) The Single Refinement algorithm of RA will be directly applied to the nine input and one output variable system obtaining a list with all binary relations and their corresponding reconstruction error.

ii) The nine input one output variable system will be decomposed into subsets of variables that must be constructed in such a way that all possible binary combinations are included.

iii) RA will be applied to each of these subsets of variables obtaining a list with all binary relations and their corresponding reconstruction errors. If binary relations are present in more than one subset, their corresponding final error strengths will be averaged.

iv) From the two lists of binary relations and their corresponding reconstruction errors, two tables of correlations between the inputs and the selected output will be constructed, one for each list.

[4]The current limit is 20 variables.

v) A comparison between the two constructed tables, i.e., the one obtained by applying RA directly to the overall system and the one obtained by applying RA to the decomposed system, will be performed. The objective of this comparison is that of demonstrating that an overall system can be decomposed into several subsystems without introducing a large error.

In a first step, the Single Refinement algorithm is directly applied to the nine input and one output system. The corresponding reconstruction error, i.e., the error that results from a comparison between the original overall system and the reconstructed system with one binary relation missing, is obtained for each binary relation. At the end, a list is configured with all possible binary relations and their corresponding reconstruction errors. In this way, strong binary relations (with a large error value) are distinguished from weak binary relations (with a small error value).

The list of binary relations and their corresponding reconstruction errors for the first output variable $y_1$ of the linear system example we have dealt with is:

STATISTICS ON BINARY RELATIONS:
******************************

| ERROR STRENGTH | BINARY RELATION OMITTED | ERROR STRENGTH | BINARY RELATION OMITTED |
|---|---|---|---|
| 0.000000 | ( 1, 2) | 0.000000 | ( 1, 4) |
| 0.000000 | ( 1, 5) | 0.001678 | ( 3, 6) |
| 0.000000 | ( 1, 6) | 0.002013 | ( 4, 9) |
| 0.000000 | ( 1, 9) | 0.002013 | ( 4, 8) |
| 0.000000 | ( 2, 5) | 0.002237 | ( 3, 9) |
| 0.000000 | ( 5, 6) | 0.003915 | ( 7, 10) |
| 0.000000 | ( 5, 9) | 0.004325 | ( 4, 9) |
| 0.000000 | ( 1, 8) | 0.004698 | ( 5, 10) |
| 0.000000 | ( 2, 10) | 0.005593 | ( 6, 10) |
| 0.000000 | ( 1, 7) | 0.006264 | ( 6, 8) |
| 0.000000 | ( 5, 7) | 0.008725 | ( 8, 9) |
| 0.000000 | ( 5, 8) | 0.008949 | ( 2, 9) |
| 0.000000 | ( 1, 10) | 0.010982 | ( 3, 7) |
| 0.000000 | ( 8, 10) | 0.013591 | ( 2, 8) |
| 0.000000 | ( 6, 7) | 0.016107 | ( 2, 3) |
| 0.000000 | ( 7, 9) | 0.020550 | ( 2, 7) |
| 0.000000 | ( 3, 5) | 0.021253 | ( 4, 7) |
| 0.000000 | ( 7, 8) | 0.036753 | ( 3, 8) |
| 0.000000 | ( 1, 3) | 0.047859 | ( 2, 6) |
| 0.000000 | ( 4, 10) | 0.054407 | ( 4, 6) |
| 0.000000 | ( 3, 10) | 0.054864 | ( 2, 4) |

```
0.000000    (   4,   5)                0.142886    (   3,   4)
                                       0.374790    (   9,  10)
```

where variables 1 to 9 represent inputs to the single row mask obtained when including time in RA, and variable number 10 stands for the $y_1$ output.

For comparison purposes, RA will be applied once more to the nine input and one output variables system, but this time the system will be decomposed into subsets of variables. These subsets of variables must be constructed in such a way that all possible binary combinations are included. Thus, for output $y_1$ these subsets might be:[5]

$$
\begin{array}{lllll}
( & u_1, & u_2, & u_3, & y_1 \ ); \\
( & u_4, & u_5, & u_6, & y_1 \ ); \\
( & u_7, & u_8, & u_9, & y_1 \ ); \\
( & u_1, & u_4, & u_7, & y_1 \ ); \\
( & u_2, & u_5, & u_8, & y_1 \ ); \\
( & u_3, & u_6, & u_9, & y_1 \ ); \\
( & u_1, & u_5, & u_9, & y_1 \ ); \\
( & u_2, & u_6, & u_7, & y_1 \ ); \\
( & u_3, & u_4, & u_8, & y_1 \ ); \\
( & u_1, & u_6, & u_8, & y_1 \ ); \\
( & u_2, & u_4, & u_9, & y_1 \ ); \\
( & u_3, & u_5, & u_7, & y_1 \ );
\end{array}
$$

The Single Refinement algorithm is applied to each of these 12 subsets, and the corresponding reconstruction error is obtained for each of its binary relations. The resulting lists for each subset (subsystem) are the following:

```
STATISTICS ON BINARY RELATIONS (1st subsystem):
***********************************************

     ERROR      BINARY RELATION      REAL
     STRENGTH   OMITTED              VARIABLES
     --------   ---------------      -----------
     0.000000   (   1,   4)          (   1,  10)
     0.028625   (   3,   4)          (   3,  10)
     0.102315   (   2,   4)          (   2,  10)
     0.143469   (   1,   2)          (   1,   2)
     0.297573   (   2,   3)          (   2,   3)
     0.354060   (   1,   3)          (   1,   3)
```

[5] This choice is by no means unique, but for this example, it is the simplest one since each binary relation between input variables appears only once.

```
STATISTICS ON BINARY RELATIONS (2nd subsystem):
***********************************************

    ERROR       BINARY RELATION      REAL
   STRENGTH     OMITTED              VARIABLES
   --------     ---------------      -----------
   0.055120     (   1,   2)          (   4,   5)
   0.090853     (   1,   4)          (   4,  10)
   0.142567     (   2,   3)          (   5,   6)
   0.190655     (   3,   4)          (   6,  10)
   0.223815     (   1,   3)          (   4,   6)
   0.247297     (   2,   4)          (   5,  10)


STATISTICS ON BINARY RELATIONS (3rd subsystem):
***********************************************

    ERROR       BINARY RELATION      REAL
   STRENGTH     OMITTED              VARIABLES
   --------     ---------------      -----------
   0.095537     (   2,   3)          (   8,   9)
   0.118563     (   1,   3)          (   7,   9)
   0.163440     (   2,   4)          (   8,  10)
   0.332099     (   1,   2)          (   7,   8)
   0.345536     (   3,   4)          (   9,  10)
   0.383620     (   1,   4)          (   7,  10)


STATISTICS ON BINARY RELATIONS (4th subsystem):
***********************************************

    ERROR       BINARY RELATION      REAL
   STRENGTH     OMITTED              VARIABLES
   --------     ---------------      -----------
   0.092265     (   2,   4)          (   4,  10)
   0.112511     (   1,   4)          (   1,  10)
   0.214425     (   1,   2)          (   1,   4)
   0.289696     (   1,   3)          (   1,   7)
   0.326471     (   2,   3)          (   4,   7)
   0.418755     (   3,   4)          (   7,  10)


STATISTICS ON BINARY RELATIONS (5th subsystem):
***********************************************

    ERROR       BINARY RELATION      REAL
   STRENGTH     OMITTED              VARIABLES
   --------     ---------------      -----------
   0.076861     (   1,   2)          (   2,   5)
```

```
     0.099844    (   2,   3)          (   5,   8)
     0.117038    (   3,   4)          (   8,  10)
     0.141190    (   1,   4)          (   2,  10)
     0.194258    (   2,   4)          (   5,  10)
     0.378487    (   1,   3)          (   2,   8)


STATISTICS ON BINARY RELATIONS (6th subsystem):
***********************************************

       ERROR     BINARY RELATION      REAL
     STRENGTH    OMITTED              VARIABLES
     --------    ---------------      -----------
     0.081102    (   1,   3)          (   3,   9)
     0.187912    (   1,   4)          (   3,  10)
     0.191312    (   2,   3)          (   6,   9)
     0.351808    (   1,   2)          (   3,   6)
     0.369842    (   3,   4)          (   9,  10)
     0.376642    (   2,   4)          (   6,  10)


STATISTICS ON BINARY RELATIONS (7th subsystem):
***********************************************

       ERROR     BINARY RELATION      REAL
     STRENGTH    OMITTED              VARIABLES
     --------    ---------------      -----------
     0.000000    (   1,   2)          (   1,   5)
     0.109123    (   2,   3)          (   5,   9)
     0.124349    (   1,   3)          (   1,   9)
     0.220798    (   1,   4)          (   1,  10)
     0.377175    (   3,   4)          (   9,  10)
     0.496779    (   2,   4)          (   5,  10)


STATISTICS ON BINARY RELATIONS (8th subsystem):
***********************************************

       ERROR     BINARY RELATION      REAL
     STRENGTH    OMITTED              VARIABLES
     --------    ---------------      -----------
     0.077715    (   1,   4)          (   2,  10)
     0.110055    (   2,   4)          (   6,  10)
     0.173976    (   3,   4)          (   7,  10)
     0.224234    (   2,   3)          (   6,   7)
     0.235201    (   1,   3)          (   2,   7)
     0.303473    (   1,   2)          (   2,   6)


STATISTICS ON BINARY RELATIONS (9th subsystem):
***********************************************
```

```
    ERROR     BINARY RELATION      REAL
   STRENGTH   OMITTED              VARIABLES
   --------   ---------------      -----------
   0.095469   (   1,    4)         (   3,  10)
   0.163050   (   2,    4)         (   4,  10)
   0.232313   (   2,    3)         (   4,   8)
   0.235105   (   3,    4)         (   8,  10)
   0.303537   (   1,    2)         (   3,   4)
   0.352968   (   1,    3)         (   3,   8)


STATISTICS ON BINARY RELATIONS (10th subsystem):
************************************************

    ERROR     BINARY RELATION      REAL
   STRENGTH   OMITTED              VARIABLES
   --------   ---------------      -----------
   0.016779   (   1,    3)         (   1,   8)
   0.033221   (   1,    4)         (   1,  10)
   0.132226   (   1,    2)         (   1,   6)
   0.137559   (   2,    3)         (   6,   8)
   0.145055   (   3,    4)         (   8,  10)
   0.323846   (   2,    4)         (   6,  10)


STATISTICS ON BINARY RELATIONS (11th subsystem):
************************************************

    ERROR     BINARY RELATION      REAL
   STRENGTH   OMITTED              VARIABLES
   --------   ---------------      -----------
   0.100531   (   2,    3)         (   4,   9)
   0.106278   (   2,    4)         (   4,  10)
   0.183408   (   1,    3)         (   2,   9)
   0.220735   (   1,    4)         (   2,  10)
   0.276872   (   1,    2)         (   2,   4)
   0.309958   (   3,    4)         (   9,  10)


STATISTICS ON BINARY RELATIONS (12th subsystem):
************************************************

    ERROR     BINARY RELATION      REAL
   STRENGTH   OMITTED              VARIABLES
   --------   ---------------      -----------
   0.045101   (   1,    2)         (   3,   5)
   0.047248   (   2,    4)         (   5,  10)
   0.099459   (   2,    3)         (   5,   7)
   0.120685   (   3,    4)         (   7,  10)
   0.129241   (   1,    4)         (   3,  10)
```

```
0.232426    (   1,   3)          (   3,   7)
```

Some of these binary relations appear twice or more times (in this example only binary relations between inputs and output), and consequently, their corresponding reconstructions errors should be averaged. At the end, a list, similar to that shown earlier, is configured with the results obtained for all the variable subsets, which means that all possible binary relations and their corresponding reconstruction errors are in the list.

The list of binary relations and their corresponding reconstruction errors for the first output variable $y_1$ of the **decomposed linear system** example is:

```
STATISTICS ON BINARY RELATIONS:
*******************************

 ERROR      BINARY RELATION              ERROR      BINARY RELATION
 STRENGTH   OMITTED                      STRENGTH   OMITTED
 --------   ---------------              --------   ---------------
 0.000000   (   1,   5)                  0.183408   (   2,   9)
 0.016779   (   1,   8)                  0.191312   (   6,   9)
 0.045101   (   3,   5)                  0.214425   (   1,   4)
 0.055120   (   4,   5)                  0.223815   (   4,   6)
 0.076861   (   2,   5)                  0.224234   (   6,   7)
 0.081102   (   3,   9)                  0.232313   (   4,   8)
 0.091633   (   1,  10)                  0.232426   (   3,   7)
 0.095537   (   8,   9)                  0.235201   (   2,   7)
 0.099459   (   5,   7)                  0.246396   (   5,  10)
 0.099844   (   5,   8)                  0.250300   (   6,  10)
 0.100531   (   4,   9)                  0.274259   (   7,  10)
 0.109123   (   5,   9)                  0.276872   (   2,   4)
 0.110312   (   3,  10)                  0.289696   (   1,   7)
 0.113112   (   4,  10)                  0.297573   (   2,   3)
 0.118563   (   7,   9)                  0.303473   (   2,   6)
 0.124349   (   1,   9)                  0.303537   (   3,   4)
 0.132226   (   1,   6)                  0.326471   (   4,   7)
 0.135489   (   2,  10)                  0.332099   (   7,   8)
 0.137559   (   6,   8)                  0.350628   (   9,  10)
 0.142567   (   5,   6)                  0.351808   (   3,   6)
 0.143469   (   1,   2)                  0.352968   (   3,   8)
 0.165160   (   8,  10)                  0.354060   (   1,   3)
                                         0.378487   (   2,   8)
```

Once again, variables 1 to 9 represents inputs to the single row mask obtained when including time in RA, and variable number 10 stands for the $y_1$ output.

In the cases of outputs $y_2$ and $y_3$, the same subsets of variables will be constructed including the respective desired output replacing $y_1$.

A strong binary relation means that the link between its associated variables cannot be severed, whereas a weak binary relation means that this link can be neglected without much of an effect on the overall system behavior. Consequently, a strong binary relation points to a strong non–linear correlation between the variables of the pair, whereas a weak binary relation points to a weak correlation. For any desired output variable, (taking only one output variable at a time), it is thus possible to identify a set of candidate input variables that exhibit strong correlations with that output, yet weak correlations among each other.

To this end, several heuristic rules will be introduced. The goal is to find a set of candidate inputs that is large enough to contain the necessary inputs for constructing a good optimal mask, yet small enough to be tractable by the inductive reasoner.[6] The complexity of this intended optimal mask is the number of variables minus one (the output is already known) that need to be selected.

### 6.3.1 Heuristic Recipes for Variable Selection

Once the list with all possible binary relations and their errors is configured, a table needs to be constructed that includes on its left side all input variables from the new number of variables, i.e., with time being included in RA; and on its right side, '+' and '–' marks that are added in accordance with the following set of rules:

1. The list of binary relations and reconstruction errors is visited from the bottom to the top, i.e., starting with the most important binary connections, and ending with the least important ones.
2. When a binary connection is encountered between any input variable and the designated output, a '+' is added to the right of the table for that input variable.
3. When a binary connection is encountered between two input variables, '–' marks are added to the right of the table for both inputs.

At the end of the procedure, each row of the table, for a system with $n$ input variables, contains $(n-1)$ '–' marks, and one '+' mark.

[6] Variable limitation problems in Optimal Mask Analysis were exposed in Chapter 3.

The table that has just been constructed indicates how strong the binary relations are between the inputs and the desired output. For the example that we have been working on (first output variable of the linear system), the resulting tables for the overall and decomposed systems look as follows:

| Input Variables | Optimal Structure Analysis Relation with Output $y_1$ |
|---|---|
| 1 | − − + − − − − − − |
| 2 | − − − − − − + − − |
| 3 | − − − − − − + − − |
| 4 | − − − − − − − − + |
| 5 | + − − − − − − − − |
| 6 | − − − + − − − − − |
| 7 | − − − + − − − − − |
| 8 | − − − − − − + − − |
| 9 | + − − − − − − − − |

Figure 6.1: Relation between input variables and output $y_1$ of the overall linear system.

| Input Variables | Optimal Structure Analysis Relation with Output $y_1$ |
|---|---|
| 1 | − − − − − − + − − |
| 2 | − − − − − − − + − |
| 3 | − − − − − − + − − |
| 4 | − − − − − − + − − |
| 5 | + − − − − − − − − |
| 6 | − − + − − − − − − |
| 7 | − − − + − − − − − |
| 8 | − − − − + − − − − |
| 9 | + − − − − − − − − |

Figure 6.2: Relation between input variables and output $y_1$ of the decomposed linear system.

The input variables with the strongest relations are the ones that should be taken into account, whereas all other variables can be neglected. Thus, two more heuristic rules are needed in order to choose these important variables. They are:

4. Those variables that have a '+' mark in the first position on the right side of the table should be considered first. These are the input variables that had the greatest reconstruction error values with respect to the output variable in the list of binary relations.

5. If there are not enough of these variables, those that have a '+' mark in the second position on the right side of the table should be considered next, and so on.

There are several criteria to stop the variable search. One of them is the aforementioned complexity of the mask. Another one is that all variables with a strong relation with the output should be considered, no matter how many they are, where "strong" is defined as a function of the reconstruction error, and is set by the user. Whatever approach is used, once the required number of variables has been reached, all the remaining variables can be neglected.

From Figure 6.1, and following rules 4 and 5, there are five possible optimal structures with complexities from 3 to 6 for the overall linear system. They are:

$$
\begin{array}{r}
(\ u_5,\ \ u_9,\ \ y_1) \\
(\ u_1,\ \ u_5,\ \ u_9,\ \ y_1) \\
(\ u_1,\ \ u_5\ \ u_6,\ \ u_9,\ \ y_1) \\
(\ u_1,\ \ u_5\ \ u_7,\ \ u_9,\ \ y_1) \\
(\ u_1,\ \ u_5,\ \ u_6,\ \ u_7,\ \ u_9,\ \ y_1)
\end{array}
\tag{6.4}
$$

Each of these relationships can be translated into a mask matrix, as was shown in Mask (6.3). The complexity and depth of these masks are given by the number of variables in the relation, and by the previously obtained slowest time constant of the system, respectively. In the case of the linear example,[7] the mask depth was computed to be 3. Thus, from Structures (6.4), the following five masks with complexities ranging from 3 to 6 are obtained for the case of the overall linear system:

$$
\begin{array}{l}
{}_{t}\backslash^{x} \\
t-2\delta t \\
t-\delta t \\
t
\end{array}
\begin{array}{c}
\begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\
\begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -2 & +1 & 0 & 0 \end{pmatrix}
\end{array}
\quad Q = 0.7188
\tag{6.5}
$$

[7] See Section 3.4 for details about the characterization of the linear system example and how its optimal masks were obtained.

$$
\begin{array}{l}
{}_t\backslash^x \\
t-2\delta t \\
t-\delta t \\
t
\end{array}
\begin{array}{c}
\begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\
\begin{pmatrix} -1 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ -3 & +1 & 0 & 0 \end{pmatrix}
\end{array}
\quad Q \;=\; 0.8626 \tag{6.6}
$$

$$
\begin{array}{l}
{}_t\backslash^x \\
t-2\delta t \\
t-\delta t \\
t
\end{array}
\begin{array}{c}
\begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\
\begin{pmatrix} -1 & 0 & 0 & 0 \\ -2 & -3 & 0 & 0 \\ -4 & +1 & 0 & 0 \end{pmatrix}
\end{array}
\quad Q \;=\; 0.3977 \tag{6.7}
$$

$$
\begin{array}{l}
{}_t\backslash^x \\
t-2\delta t \\
t-\delta t \\
t
\end{array}
\begin{array}{c}
\begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\
\begin{pmatrix} -1 & 0 & 0 & 0 \\ -2 & 0 & -3 & 0 \\ -4 & +1 & 0 & 0 \end{pmatrix}
\end{array}
\quad Q \;=\; 0.3918 \tag{6.8}
$$

$$
\begin{array}{l}
{}_t\backslash^x \\
t-2\delta t \\
t-\delta t \\
t
\end{array}
\begin{array}{c}
\begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\
\begin{pmatrix} -1 & 0 & 0 & 0 \\ -2 & -3 & -4 & 0 \\ -5 & +1 & 0 & 0 \end{pmatrix}
\end{array}
\quad Q \;=\; 0.1317 \tag{6.9}
$$

The mask quality values, $Q$, were added to the right of each of the masks. They were obtained using FIR as explained in Chapter 3.[8]

For the decomposed linear system, according to Figure 6.2 and following the rules 4 and 5, there are four possible optimal structures with complexities ranging from 3 to 6. They are:

$$
\begin{array}{r}
(\; u_5,\;\; u_9,\;\; y_1) \\
(\; u_5,\;\; u_6,\;\; u_9,\;\; y_1) \\
(\; u_5,\;\; u_6\;\; u_7,\;\; u_9,\;\; y_1) \\
(\; u_5,\;\; u_6,\;\; u_7,\;\; u_8,\;\; u_9,\;\; y_1)
\end{array} \tag{6.10}
$$

From Structures (6.10), the following four masks with complexities ranging from 3 to 6 are obtained for the case of the decomposed linear system:

[8]In Chapter 3 the mask quality was denoted as $Q_M$, Equation (3.26).

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -2 & +1 & 0 & 0 \end{pmatrix} \end{array} \quad Q = 0.7188 \tag{6.11}$$

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & -2 & 0 & 0 \\ -3 & +1 & 0 & 0 \end{pmatrix} \end{array} \quad Q = 0.7954 \tag{6.12}$$

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & 0 \\ -4 & +1 & 0 & 0 \end{pmatrix} \end{array} \quad Q = 0.3126 \tag{6.13}$$

$$\begin{array}{l} {}_t\backslash^x \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -4 \\ -5 & +1 & 0 & 0 \end{pmatrix} \end{array} \quad Q = 0.1186 \tag{6.14}$$

Let us now compare the tables, optimal structures, and masks obtained for the overall linear system with those obtained for the decomposed linear system.

From the tables of Figures 6.1 and 6.2, it can be seen that the main difference relates to variables $u_1$ and $u_8$. In the first table, $u_1$ is an important variable, whereas in the second table it is not, and instead, another variable $u_8$ appears.

This difference implies that only one of the optimal structures presented in (6.4) coincides with one of the structures presented in (6.10). However, it can also be noticed that the basic structure composed of $u_5$, $u_6$, $u_7$, and $u_9$ is common to both approaches. This fact means that, although the decomposition of a system into several subsystem adds an error that may be viewed as a loss of information, this error is not unacceptably large.

This statement is confirmed when comparing the qualities of the resulting masks obtained for the overall linear system, Masks (6.5), (6.6), (6.7), (6.8), and (6.9), with those obtained for the decomposed linear system, Masks (6.11), (6.12), (6.13), and (6.14). The qualities of the masks belonging to the same complexity levels are all quite similar. Hence the quality reduction due to decomposition is small and therefore acceptable.

Now that it has been shown that the decomposition of a system into several subsystems can be used without introducing an unacceptably large error, the following question can be raised: Will the optimal structures obtained by the mixed RA/heuristic–recipes methodology be always translatable directly into optimal masks?

The answer depends on the number of selected variables. There are two possibilities. On the one hand, if the number of selected variables does not exceed the allowed complexity of the intended mask, then those variables directly constitute suboptimal masks proposed by the mixed RA/heuristic–recipes methodology, as is the case for Masks (6.5) to (6.8) and (6.11) to (6.13), since a maximum complexity of 5 was originally computed for this system. On the other hand, if the number of variables is greater than the allowed complexity of the intended mask, as is the case for Masks (6.9) and (6.14), Optimal Mask Analysis should now be applied to the already reduced system, in which the input variables proposed by the RA methodology are chosen as potential inputs in the mask candidate matrix.

At this point, it is interesting to make a comparison between the results obtained here by Optimal Structure Analysis and the heuristic recipes constructed around it on the one side, and the results obtained by Optimal Mask Analysis on the other. Moreover, it may also be of interest to make a comparison against a typical Correlation Analysis to see if what has been proposed in this chapter is reasonable.

## 6.4 Optimal Structure and Correlation *vs.* Optimal Mask Analyses

Optimal Structure and Correlation Analyses can be seen as alternatives to FIR (Optimal Mask Analysis) for determining high–quality masks. To assert this statement, a comparison should be performed between Optimal Structure and Correlation Analyses on the one hand, and Optimal Mask Analysis on the other.

To make these comparisons, let us apply Reconstruction Analysis and the aforementioned recipes to some of the examples that have been presented in previous chapters, namely, the linear system described in Chapter 3, and the aircraft model introduced in Chapter 4. These short– and medium–sized systems have been chosen because of two reasons. The first one is that both of them can be treated by FIR and RA methodologies; and the second one is that the results of applying Optimal Mask Analysis to both system have already been presented in previous chapters.

The goal is to demonstrate, by means of the aforementioned examples, that the relationships between inputs and outputs obtained with the mixed RA/heuristic–recipes methodology proposed here are very similar to those obtained with Optimal Mask Analysis, and somewhat better than those obtained by Correlation Analysis. It can be assumed that if the RA methodology gives accurate results when applied to short– and medium–sized systems, it will also give accurate results when applied to large–scale systems, i.e., the methodological set of tools explained along this chapter constitutes a powerful technique for the selection and causal grouping of variables in large–scale systems.

The comparison against Correlation Analysis will be carried out to demonstrate that a methodology based on non–linear relations, such as RA, is better suited to perform a reasonable selection and grouping of variables than one based on linear relations, such as Correlation Analysis. The correlations will be computed by using appropriate Matlab [MathWorks, 1992a] functions as will be explained in the following section.

### 6.4.1 Correlation Analysis

Correlation Analysis is a well known statistical technique used to discover linear relationships (patterns) between variables. For this reason, only a brief explanation will be given.

Let us suppose that we have the three inputs and one output raw data matrix shown in Matrix (6.1), where each row is an observation, and each column a variable. The "covariance" of a single variable is a number that indicates its variance. For the entire matrix, the covariance is another matrix the diagonal of which is a vector of variances for each column, or which is saying the same, a vector of squared standard deviations.

The correlations between a set of variables is computed using the *corrcoef* function of Matlab. The application of this function to the raw data matrix returns a matrix of correlation coefficients $S$ whose $(i, j)$ element is related to the aforementioned covariance matrix $C$ by:

$$S(i,j) \;=\; \frac{C(i,j)}{\sqrt{C(i,i)\;C(j,j)}} \tag{6.15}$$

The Matlab code to perform this operation for the first output variable of the linear example is the following:

```
>>  mcan = [ -1 -2 -3 -4
             -5 -6 -7 -8
             -9 +1  0  0 ]
>>  io = iomodel(raw,mcan);
>>  coef = corrcoef(io);
>>  coefabs = abs(coef);
>>  [order,I] = sort(coefabs);
```

The mask candidate matrix *mcan* indicates to the FIR command *iomodel*, how many times the original raw data matrix *raw* should be duplicated and concatenated to the right shifted up by one row (the inclusion of time in RA). Then, the Matlab command *corrcoef* computes the correlation coefficients, the absolute values of which are calculated by means of the Matlab command *abs*. Finally, the coefficients are sorted in increasing order by means of the *sort* command, and are stored in the *order* matrix. The correspondence between the sorted correlation coefficients and the variables they belong to is stored in the matrix *I*. In this way, each column of this matrix indicates, in a bottom–up approach, how strong the relationships are between variables.

As in the cases of Optimal Structure and Optimal Mask Analyses, what is important are the relationships between inputs and the desired output. The *I* matrix contains those relationships. Let us revisit the example that we have been dealing with all along this chapter, i.e., the first output variable of the linear system presented in Chapter 3. The *I* matrix of the $y_1$ output is:

$$I = \begin{pmatrix} 5 & 9 & 9 & 9 & 9 & 9 & 9 & 2 & 2 & 3 \\ 9 & 8 & 5 & 5 & 1 & 7 & 6 & 9 & 5 & 4 \\ 10 & 5 & 2 & 10 & 2 & 10 & 3 & 5 & 1 & 6 \\ 4 & 3 & 10 & 1 & 3 & 2 & 4 & 10 & 3 & 1 \\ 7 & 6 & 7 & 7 & 4 & 4 & 8 & 7 & 6 & 2 \\ 2 & 10 & 4 & 6 & 8 & 5 & 1 & 4 & 7 & 8 \\ 8 & 7 & 6 & 3 & 6 & 8 & 2 & 6 & 4 & 9 \\ 3 & 1 & 8 & 8 & 10 & 3 & 10 & 1 & 8 & 7 \\ 6 & 4 & 1 & 2 & 7 & 1 & 5 & 3 & 10 & 5 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix} \tag{6.16}$$

where the last row indicates that the maximum possible correlation of each variable is with itself. Variables 1 to 9 are the input variables whereas variable 10 is the output variable. Thus, we are interested in those columns in which variable 10 appears at the lowermost positions, i.e., variables 5, 7, and 9.

In the preceding section, five heuristic recipes were proposed in order to interpret the resulting list of binary relations and corresponding reconstruction errors obtained by the application of the Single Refinement Algorithm. The result of the recipes was the construction of the table filled with '–' and '+' marks presented in Figure 6.2 that clearly indicates which, among all input variables, had strong relationships with the desired output.

A similar table can be constructed here to interpret the results obtained by Correlation Analysis, or saying the same, to interpret the $I$ matrix. As a matter of fact, the $I$ matrix presented above, Matrix (6.16), can be converted directly into a table by simply rotating it 90 degrees clockwise. In this way, the columns of the $I$ matrix, except for the last one that corresponds to the output variable and is thus eliminated, are now the rows of the table. In each row, all input variables are substituted by a '–' mark, whereas the desired output variable is substituted by a '+' mark. The resulting table is depicted in Figure 6.3.

| Input Variables | Correlation Analysis Relation with Output $y_1$ |
|---|---|
| 1 | − − − − − − + − − |
| 2 | − − − + − − − − − |
| 3 | − − − − − + − − − |
| 4 | − − − − − − + − − |
| 5 | − + − − − − − − − |
| 6 | − − − − − − + − − |
| 7 | − + − − − − − − − |
| 8 | − − − − − + − − − |
| 9 | + − − − − − − − − |

Figure 6.3: Correlation between input variables and output $y_1$ of the decomposed linear system.

From this table, the following relationships can be extracted:

$$\begin{array}{rrrrr} & & & ( u_9, & y_1) \\ & & ( u_5, & u_9, & y_1) \\ & & ( u_7, & u_9, & y_1) \\ & ( u_5, & u_7, & u_9, & y_1) \\ ( u_2, & u_5, & u_7, & u_9, & y_1) \end{array} \quad (6.17)$$

All of them can be translated into optimal masks as was done in the preceding section.

In this way, equivalent relationships that can be compared between them are obtained by the three methodologies (Optimal Structure, Optimal Mask, and Correlation Analyses).

### 6.4.2 The Linear System Example

Along this chapter, the first output variable of the linear system presented in Chapter 4 has been used as an example to demonstrate, step by step, firstly, how Reconstruction Analysis and the heuristic rules constructed around it work, and secondly, that the error introduced by the decomposition of the overall system into subsystems that altogether include all possible binary relations is very small and acceptable.

In this section, the three output variables of this example will be analyzed using the mixed RA/heuristic–recipes methodology previously explained, and their results will be compared with those obtained by Optimal Mask and Correlation Analyses. In all cases, the resulting structures of the former of these methodologies will correspond to the decomposed linear system,[9] and the comparison will be carried out using the quality values of their corresponding masks.

#### 6.4.2.1 Output Variable $y_1$

For the $y_1$ output variable, the results obtained by the Optimal Structure Analysis are summarized by the following list of relationships. Since there is

[9]The detailed results of applying RA to each of the subsystems (resulting from the decomposed system) will not be presented due to space limitations. Instead, only the table of relations between the input variables and the desired output will be shown.

no need any more to translate these relationships into masks, let us include the complexity and the quality of the mask that each of these relations represents. The list is:

$$
\begin{array}{rlll}
(\ u_5,\ u_9,\ y_1) & C = 3, & Q = 0.7188 & \\
(\ u_5,\ u_6,\ u_9,\ y_1) & C = 4, & Q = 0.7954 & \\
(\ u_5,\ u_6\ u_7,\ u_9,\ y_1) & C = 5, & Q = 0.3126 & \\
(\ u_5,\ u_6,\ u_7,\ u_8,\ u_9,\ y_1) & C = 6, & Q = 0.1186 & (6.18)
\end{array}
$$

where the relationships are those from Structures (6.10), and the qualities are those from Masks (6.11), (6.12), (6.13), and (6.14). Notice that the two first structures have a high quality value, whereas the third structure shows a much lower value, and the fourth one exhibits an even more dramatic quality drop. It seems that, for the Optimal Structure Analysis, the complexity level 4 coincides with the inflection point of the quality function, and thus, the best possible mask is that of complexity 4. As will be immediately shown, this conclusion is indeed correct.

Following the same way of expressing relationships and qualities that has been established above, let us include, for comparison purposes, those relationships obtained by Optimal Mask Analysis for all complexity levels (from 2 to 6). They were:

$$
\begin{array}{rlll}
(\ u_5,\ y_1) & C = 2, & Q = 0.4447 & \\
(\ u_5,\ u_9,\ y_1) & C = 3, & Q = 0.7188 & \\
(\ u_2,\ u_5,\ u_9,\ y_1) & C = 4, & Q = 0.8772 & \\
(\ u_1,\ u_2,\ u_5,\ u_9,\ y_1) & C = 5, & Q = 0.8509 & \\
(\ u_1,\ u_2,\ u_3,\ u_5,\ u_9,\ y_1) & C = 6, & Q = 0.3189 & (6.19)
\end{array}
$$

Notice that these relationships include one more than those presented in (6.18), since the Optimal Structure Analysis did not obtain any relationship of complexity 2. As in the previous case, the best mask is the one of complexity 4. Notice that, in spite of having a higher quality value, relationships of complexity levels 3 and 4 of this set are not far apart from relationships of the same complexity levels of the previous set. As a matter of fact, one of them is exactly the same.

Finally, and once more for comparison purposes, let us include the results previously obtained by the Correlation Analysis and presented in Structures (6.17), they were:

$$\begin{array}{rrrrrl} ( & & & u_9, & y_1) & C = 2, \quad Q = 0.2283 \\ ( & & u_5, & u_9, & y_1) & C = 3, \quad Q = 0.7188 \\ ( & & u_7, & u_9, & y_1) & C = 3, \quad Q = 0.6838 \\ ( & u_5, & u_7, & u_9, & y_1) & C = 4, \quad Q = 0.5317 \\ ( u_2, & u_5, & u_7, & u_9, & y_1) & C = 5, \quad Q = 0.3662 \end{array} \tag{6.20}$$

In this case, there are two different relationships of complexity 3. Contrary to the previous two cases, Correlation Analysis concludes that the mask of maximum quality is of complexity 3. Beside from that, and beside from finding masks with lower quality values in general, Correlation Analysis managed to obtain two relatively good masks, namely those of complexity 3. This acceptable performance of the Correlation Analysis is due to the linear characteristics of the example.

Making a comparison as a whole between the four optimal masks obtained by the Optimal Structure Analysis of the RA methodology, the five optimal masks previously obtained by the Optimal Mask Analysis of the FIR methodology, and the five masks obtained by the Correlation Analysis, it can be said that the Optimal Structure Analysis delivered better results than those obtained by Correlation Analysis, not only because the quality values are higher in the former than in the latter, but because the former found in general masks that are closer to the truly optimal masks found by FIR than the latter.

The maximum complexity level for the linear system example had been set to 5, thus no masks with complexity 6 were allowed. The results have demonstrated clearly the dependence between the quality of a mask and the complexity,[10] and have shown that limiting the search to a maximum complexity level of 5 was indeed justified.

#### 6.4.2.2 Output Variable $y_2$

Let us see what happens when analyzing the second output variable, $y_2$. The system has been decomposed in exactly the same manner as shown for the case

[10]In Chapter 3, it has been shown that the quality of a mask rises with the complexity level until a certain limit, after which the quality decays rapidly.

of the first output variable $y_1$. Once more, the mixed RA/heuristic–recipes is applied 12 times, one for each decomposed subsystem, and once more the reconstruction error of those binary relations that appear more than once are averaged. The resulting list of binary relations and their corresponding reconstruction errors for the output variable $y_2$ of the decomposed linear system example is:[11]

```
STATISTICS ON BINARY RELATIONS:
******************************

 ERROR       BINARY RELATION              ERROR       BINARY RELATION
 STRENGTH    OMITTED                      STRENGTH    OMITTED
 --------    ---------------              --------    ---------------
 0.000000       ( 1,  9)                  0.199664       ( 3,  7)
 0.011505       ( 8,  9)                  0.200533       ( 4,  8)
 0.018456       ( 3,  9)                  0.207697       ( 7, 11)
 0.042184       ( 3,  5)                  0.227968       ( 8, 11)
 0.069865       ( 2,  9)                  0.232710       ( 4,  6)
 0.072349       ( 7,  9)                  0.235710       ( 6, 11)
 0.086541       ( 4,  9)                  0.262185       ( 5,  6)
 0.097177       ( 6,  9)                  0.267260       ( 5,  7)
 0.101312       ( 1,  4)                  0.277474       ( 3,  6)
 0.101509       ( 1, 11)                  0.291460       ( 2,  3)
 0.103912       ( 3, 11)                  0.292110       ( 1,  3)
 0.106120       ( 5,  9)                  0.300586       ( 2,  6)
 0.115660       ( 2, 11)                  0.303100       ( 1,  7)
 0.122523       ( 1,  6)                  0.309033       ( 3,  4)
 0.128694       ( 2,  5)                  0.312683       ( 3,  8)
 0.138656       ( 1,  5)                  0.317011       ( 4,  7)
 0.141822       ( 4, 11)                  0.317887       ( 5,  8)
 0.146965       ( 1,  8)                  0.336150       ( 2,  4)
 0.151969       ( 4,  5)                  0.352800       ( 2,  7)
 0.154959       ( 6,  8)                  0.361337       ( 2,  8)
 0.155203       ( 1,  2)                  0.373131       ( 6,  7)
 0.166619       ( 5, 11)                  0.437980       ( 7,  8)
                                          0.452420       ( 9, 11)
```

From this list, and following the recipes presented in the previous section, the table that graphically represents the relationships between the input variables and the $y_2$ output variable can be constructed. This table is depicted in Figure 6.4.

From this table, five relationships can be extracted. They are:

[11] It has already been said that the application of the mixed RA/heuristic–recipes for output variables $y_2$ and $y_3$ will only be shown for the case of the decomposed subsystem.

| Input Variables | Optimal Structure Analysis Relation with Output $y_2$ |
|---|---|
| 1 | $- - - - - - + - -$ |
| 2 | $- - - - - - - + -$ |
| 3 | $- - - - - - + - -$ |
| 4 | $- - - - - - + - -$ |
| 5 | $- - - + - - - - -$ |
| 6 | $- - - - + - - - -$ |
| 7 | $- - - - - - + - -$ |
| 8 | $- - - - + - - - -$ |
| 9 | $+ - - - - - - - -$ |

Figure 6.4: Relation between input variables and output $y_2$ of the decomposed linear system.

$$\begin{array}{rll} (\ u_9,\ y_2) & C = 2, & Q = 0.4178 \\ (\ u_5,\ u_9,\ y_2) & C = 3, & Q = 0.6700 \\ (\ u_5,\ u_6,\ u_9,\ y_2) & C = 4, & Q = 0.7503 \\ (\ u_5,\ u_8,\ u_9,\ y_2) & C = 4, & Q = 0.8348 \\ (\ u_5,\ u_6,\ u_8,\ u_9,\ y_2) & C = 5, & Q = 0.3900 \end{array} \tag{6.21}$$

Notice that, this time around, Optimal Structure Analysis did find a relationship of complexity 2, and two relationships of complexity 4, but did not find a relationship of complexity 6. Once more the quality function seems to have an inflection point at complexity level 4, as the suboptimal structures of this level are the ones with the highest quality values. Again, this is the case for the true optimal masks as well.

The optimal masks obtained by the Optimal Mask Analysis for each complexity level were:

$$\begin{array}{rll} (\ u_9,\ y_2) & C = 2, & Q = 0.4178 \\ (\ u_6,\ u_9,\ y_2) & C = 3, & Q = 0.6853 \\ (\ u_3,\ u_5,\ u_9,\ y_2) & C = 4, & Q = 0.8696 \\ (\ u_1,\ u_2,\ u_5,\ u_9,\ y_2) & C = 5, & Q = 0.8390 \\ (\ u_1,\ u_2,\ u_4,\ u_5,\ u_9,\ y_2) & C = 6, & Q = 0.3726 \end{array} \tag{6.22}$$

It can be noticed that the quality values of the optimal masks of complexities 4 and 5 are almost the same, which means that the inflection point of the quality function is near the complexity level 5. The fact that the first relation of this set (complexity 2) coincides with the first relation of the previous set is irrelevant, because their quality value of 0.4178 is far away from the maximum quality value of 0.8696. What is really important is that, at complexity levels 3 and 4, Optimal Structure Analysis obtained very good suboptimal relations, and that the quality function behaved in very similar ways in both cases.

Finally, and once more for comparison purposes, let us include the results obtained from the Correlation Analysis:

$$\begin{array}{rlrl}
(\ u_9,\ \ y_2) & C\ =\ 2, & Q\ =\ 0.4178 \\
(\ u_5,\ \ u_9,\ \ y_2) & C\ =\ 3, & Q\ =\ 0.6700 \\
(\ u_3,\ \ u_5,\ \ u_9,\ \ y_2) & C\ =\ 4, & Q\ =\ 0.8696 \\
(\ u_4,\ \ u_5,\ \ u_9,\ \ y_2) & C\ =\ 4, & Q\ =\ 0.7602 \\
(\ u_5,\ \ u_6,\ \ u_9,\ \ y_2) & C\ =\ 4, & Q\ =\ 0.7503 \\
(\ u_3,\ \ u_4,\ \ u_5,\ \ u_6\ \ u_9,\ \ y_2) & C\ =\ 6, & Q\ =\ 0.1924
\end{array} \tag{6.23}$$

and observe that there are three different relations of complexity 4 and none of complexity 5. One of the level 4 relations coincides with the optimal mask, and the other two are good suboptimal approximations. Another good suboptimal mask is the one of complexity 3, without mentioning that the one of complexity 2 coincides with the optimal mask at that level.

In the comparison as a whole between the three sets of relationships, Correlation Analysis gives here slightly better results than those obtained by Optimal Structure Analysis. However, the performance of the latter methodology was quite acceptable as well.

#### 6.4.2.3 Output Variable $y_3$

For the third output variable of this decomposed linear system example, $y_3$, the list of binary relations and their corresponding errors is:

```
STATISTICS ON BINARY RELATIONS:
*******************************
```

| ERROR STRENGTH | BINARY RELATION OMITTED | ERROR STRENGTH | BINARY RELATION OMITTED |
|---|---|---|---|
| 0.000000 | ( 4, 5) | 0.173882 | ( 3, 7) |
| 0.000000 | ( 5, 9) | 0.177617 | ( 1, 9) |
| 0.000000 | ( 1, 8) | 0.181578 | ( 6, 7) |
| 0.000000 | ( 5, 7) | 0.206503 | ( 2, 9) |
| 0.050448 | ( 3, 9) | 0.217523 | ( 2, 7) |
| 0.064408 | ( 3, 5) | 0.219240 | ( 4, 6) |
| 0.069032 | ( 2, 5) | 0.228369 | ( 8, 12) |
| 0.091379 | ( 7, 9) | 0.240340 | ( 4, 8) |
| 0.970975 | ( 8, 9) | 0.260434 | ( 1, 7) |
| 0.106120 | ( 1, 5) | 0.264578 | ( 5, 12) |
| 0.108926 | ( 6, 9) | 0.267128 | ( 9, 12) |
| 0.118957 | ( 6, 8) | 0.280890 | ( 6, 12) |
| 0.121920 | ( 4, 12) | 0.296311 | ( 2, 6) |
| 0.126526 | ( 1, 4) | 0.298034 | ( 7, 12) |
| 0.130816 | ( 1, 12) | 0.298240 | ( 3, 4) |
| 0.136479 | ( 5, 6) | 0.302175 | ( 3, 6) |
| 0.137522 | ( 5, 8) | 0.307223 | ( 1, 3) |
| 0.138013 | ( 2, 12) | 0.309630 | ( 7, 8) |
| 0.145458 | ( 1, 6) | 0.319304 | ( 4, 7) |
| 0.148642 | ( 1, 2) | 0.319404 | ( 3, 8) |
| 0.162073 | ( 3, 12) | 0.332611 | ( 2, 3) |
| 0.167087 | ( 4, 9) | 0.376411 | ( 2, 4) |
| | | 0.407764 | ( 2, 8) |

The table that results from the applications of the heuristic recipes is depicted in Figure 6.5.

The following four optimal structures can be read out of this table:

$$\begin{array}{rll} (\; u_5,\; u_9,\; y_3) & C = 3, & Q = 0.5094 \\ (\; u_5,\; u_6,\; u_9,\; y_3) & C = 4, & Q = 0.7965 \\ (\; u_5,\; u_7,\; u_9,\; y_3) & C = 4, & Q = 0.4270 \\ (\; u_5,\; u_6,\; u_7,\; u_9,\; y_3) & C = 5, & Q = 0.3096 \end{array} \tag{6.24}$$

There are no optimal structures for complexity levels 2 and 6. Notice the low quality of all structures but the second, which is a good suboptimal approximation to the optimal mask of that level.

As in the case of the previous two output variables, the masks obtained by the Optimal Mask Analysis and the correlations obtained by the Correlation

| Input Variables | Optimal Structure Analysis Relation with Output $y_3$ |
|---|---|
| 1 | − − − − − + − − − |
| 2 | − − − − − − − + − |
| 3 | − − − − − − + − − |
| 4 | − − − − − − − + − |
| 5 | + − − − − − − − − |
| 6 | − − + − − − − − − |
| 7 | − − + − − − − − − |
| 8 | − − − − + − − − − |
| 9 | + − − − − − − − − |

Figure 6.5: Relation between input variables and output $y_3$ of the decomposed linear system.

Analysis will be included. The optimal masks found for each complexity level were:

$$
\begin{array}{rlll}
(u_7, & y_3) & C = 2, & Q = 0.4663 \\
(u_1, u_5, & y_3) & C = 3, & Q = 0.6818 \\
(u_1, u_5, u_9, & y_3) & C = 4, & Q = 0.8835 \\
(u_1, u_2, u_5, u_9, & y_3) & C = 5, & Q = 0.8270 \\
(u_1, u_2, u_4, u_5, u_9, & y_3) & C = 6, & Q = 0.3786
\end{array}
\tag{6.25}
$$

The quality values of the optimal masks are very similar to those obtained for the previous output ($y_2$), and consequently, the quality function is also very similar.

Finally, the correlations and their qualities are:

$$
\begin{array}{rlll}
(u_9, & y_3) & C = 2, & Q = 0.1311 \\
(u_5, u_9, & y_3) & C = 3, & Q = 0.5094 \\
(u_7, u_9, & y_3) & C = 3, & Q = 0.5984 \\
(u_5, u_7, u_9, & y_3) & C = 4, & Q = 0.4270 \\
(u_2, u_5, u_7, u_9, & y_3) & C = 5, & Q = 0.3226
\end{array}
\tag{6.26}
$$

The quality values are far from those of the optimal masks. None of these relationships is a good suboptimal approximation, and it seems clear that, for this case, Correlation Analysis performs rather poorly.

The comparison as a whole of the three sets of relationships shows that this third output is much more difficult to be characterized than the other two. Whereas Optimal Mask Analysis is capable of finding high quality optimal masks, Optimal Structure Analysis is capable of finding just one good suboptimal approximation, and Correlation Analysis is not capable of finding any meaningful masks.

Taking into account the results obtained for the three output variables, and in particular the relations proposed for complexity level 4, which is the level where all optimal masks were found, Optimal Structure Analysis showed a very good performance in two cases, and a decent performance in the third one, whereas Correlation Analysis exhibited one good, one decent, and one poor performance.

### 6.4.3 The Aircraft Example

As it was done with the linear system, the $B4$ aircraft model will be used to verify that RA and the heuristic recipes proposed in this chapter work fine. To this end, what was called in Chapter 3 the *shaken flight raw data matrix* is used as the original input data for model $B4$. This data is already *recoded* using the same number of qualitative classes and the same landmarks as in Section 4.3.3.3. As a matter of fact, these are exactly the same data that were used to obtain the qualitative model of the shaken flight. Thus, the same mask depth can be considered.

The shaken flight raw data matrix should be reorganized in order for the RA methodology to recognize its temporal causality. Thus, as was explained in Section 6.2 and done in the previous example, also this matrix should be triplicated. Thus, a mask with 12 input variables and 3 output variables (15 columns) will be obtained.

Then, as it was done for the first output variable of the linear system example, the mixed RA/heuristic–recipes methodology will be applied to the decomposed aircraft system example in order to demonstrate once more that no significant error is introduced with such a procedure.

#### 6.4.3.1 Output Variable *Lift*

This example, as the preceding one, can be managed by the RA methodology without needing any decomposition, but the goal is that of demonstrating that the methodology works also on a medium–sized decomposed system.

To this end, the 15 variables should be grouped together into subsets, in such a way that all possible binary combinations are included. The final effects of repeated binary relations will be averaged. As in the previous example, each subset must be composed of several inputs and one output. For the case of the first output variable *Lift*, the way in which variables have been grouped (this choice is by no means unique) is:

$$
\begin{array}{lllllll}
( & u_1, & u_2, & u_3, & u_4, & y_1 & ); \\
( & u_5, & u_6, & u_7, & u_8, & y_1 & ); \\
( & u_9, & u_{10}, & u_{11}, & u_{12}, & y_1 & ); \\
( & u_1, & u_5, & u_6, & u_9, & y_1 & ); \\
( & u_2, & u_6, & u_7, & u_{10}, & y_1 & ); \\
( & u_3, & u_7, & u_8, & u_{11}, & y_1 & ); \\
( & u_4, & u_5, & u_8, & u_4, & y_2 & ); \\
( & u_1, & u_4, & u_6, & u_{11}, & y_1 & ); \\
( & u_1, & u_2, & u_7, & u_{12}, & y_1 & ); \\
( & u_2, & u_3, & u_8, & u_9, & y_1 & ); \\
( & u_3, & u_4, & u_5, & u_{10}, & y_1 & ); \\
( & u_4, & u_7, & u_9, & u_{10}, & y_1 & ); \\
( & u_3, & u_6, & u_9, & u_{12}, & y_1 & ); \\
( & u_2, & u_5, & u_{11}, & u_{12}, & y_1 & ); \\
( & u_1, & u_8, & u_{10}, & u_{11}, & y_1 & );
\end{array}
$$

with the same subsets being used for output variables $y_2$ (Drag) and $y_3$ (the flight path angle $\gamma$). Keeping the number of variables in each subset low, the computation of the optimal structure for each subset is easier, but the number of subsets is increased. On the other hand, letting the number of variables in each subset be a little larger, the number of subsets is reduced, but the computation to obtain the optimal structure of each subset is increased.

Once more, the Single Refinement algorithm of the Optimal Structure Analysis of the Reconstruction Analysis methodology is applied to each of these subsets, and the corresponding reconstruction errors are obtained for each of its binary relations. The results of the fifteen subsystems are:

```
STATISTICS ON BINARY RELATIONS (1st subsystem):
**********************************************

     ERROR     BINARY RELATION        REAL
   STRENGTH    OMITTED                VARIABLES
   --------    ---------------        -----------
   0.022248    (   2,   4)            (   2,   4)
   0.023994    (   2,   3)            (   2,   3)
   0.028905    (   3,   5)            (   3,  13)
   0.030690    (   1,   3)            (   1,   3)
   0.033839    (   1,   2)            (   1,   2)
   0.034774    (   1,   4)            (   1,   4)
   0.039656    (   2,   5)            (   2,  13)
   0.040981    (   4,   5)            (   4,  13)
   0.200288    (   1,   5)            (   1,  13)
   0.480018    (   3,   4)            (   3,   4)


STATISTICS ON BINARY RELATIONS (2nd subsystem):
**********************************************

     ERROR     BINARY RELATION        REAL
   STRENGTH    OMITTED                VARIABLES
   --------    ---------------        -----------
   0.073966    (   3,   4)            (   7,   8)
   0.074992    (   1,   2)            (   5,   6)
   0.075631    (   1,   3)            (   5,   7)
   0.078162    (   2,   3)            (   6,   7)
   0.090480    (   2,   4)            (   6,   8)
   0.163651    (   3,   5)            (   7,  13)
   0.172568    (   1,   5)            (   5,  13)
   0.180412    (   4,   5)            (   8,  13)
   0.237464    (   1,   4)            (   5,   8)
   0.262478    (   2,   5)            (   6,  13)


STATISTICS ON BINARY RELATIONS (3rd subsystem):
**********************************************

     ERROR     BINARY RELATION        REAL
   STRENGTH    OMITTED                VARIABLES
   --------    ---------------        -----------
   0.033196    (   3,   4)            (  11,  12)
   0.034826    (   2,   3)            (  10,  11)
   0.034937    (   2,   4)            (  10,  12)
   0.038251    (   3,   5)            (  11,  13)
   0.038817    (   1,   4)            (   9,  12)
   0.040257    (   4,   5)            (  12,  13)
   0.042876    (   1,   3)            (   9,  11)
   0.051253    (   1,   5)            (   9,  13)
   0.146398    (   1,   2)            (   9,  10)
```

```
     0.185238    (   2,    5)             (  10,  13)


STATISTICS ON BINARY RELATIONS (4th subsystem):
**********************************************

       ERROR     BINARY RELATION         REAL
     STRENGTH    OMITTED                 VARIABLES
     --------    ---------------         -----------
     0.028863    (   3,    4)             (   6,   9)
     0.033494    (   1,    2)             (   1,   5)
     0.036241    (   4,    5)             (   9,  13)
     0.052801    (   2,    3)             (   5,   6)
     0.099322    (   2,    5)             (   5,  13)
     0.104712    (   1,    3)             (   1,   6)
     0.118424    (   1,    4)             (   1,   9)
     0.130818    (   2,    4)             (   5,   9)
     0.220201    (   1,    5)             (   1,  13)
     0.284460    (   3,    5)             (   6,  13)


STATISTICS ON BINARY RELATIONS (5th subsystem):
**********************************************

       ERROR     BINARY RELATION         REAL
     STRENGTH    OMITTED                 VARIABLES
     --------    ---------------         -----------
     0.027346    (   1,    3)             (   2,   7)
     0.050696    (   1,    2)             (   2,   6)
     0.073080    (   1,    5)             (   2,  13)
     0.085159    (   2,    3)             (   6,   7)
     0.094474    (   1,    4)             (   2,  10)
     0.099454    (   3,    4)             (   7,  10)
     0.130152    (   2,    4)             (   6,  10)
     0.162206    (   3,    5)             (   7,  13)
     0.253094    (   4,    5)             (  10,  13)
     0.281120    (   2,    5)             (   6,  13)


STATISTICS ON BINARY RELATIONS (6th subsystem):
**********************************************

       ERROR     BINARY RELATION         REAL
     STRENGTH    OMITTED                 VARIABLES
     --------    ---------------         -----------
     0.033744    (   2,    4)             (   7,  11)
     0.037162    (   1,    2)             (   3,   7)
     0.037336    (   3,    4)             (   8,  11)
     0.040434    (   2,    3)             (   7,   8)
     0.041266    (   1,    4)             (   3,  11)
     0.044945    (   4,    5)             (  11,  13)
```

```
      0.077684    (   1,   5)                (   3,  13)
      0.097986    (   3,   5)                (   8,  13)
      0.098886    (   1,   3)                (   3,   8)
      0.132240    (   2,   5)                (   7,  13)


 STATISTICS ON BINARY RELATIONS (7th subsystem):
 **********************************************

        ERROR     BINARY RELATION            REAL
      STRENGTH    OMITTED                    VARIABLES
      --------    ---------------            -----------
      0.044053    (   3,   4)                (   8,  12)
      0.045679    (   4,   5)                (  12,  13)
      0.046700    (   1,   4)                (   4,  12)
      0.046974    (   2,   4)                (   5,  12)
      0.061810    (   1,   5)                (   4,  13)
      0.065707    (   1,   3)                (   4,   8)
      0.137454    (   2,   5)                (   5,  13)
      0.139803    (   1,   2)                (   4,   5)
      0.149890    (   3,   5)                (   8,  13)
      0.218999    (   2,   3)                (   5,   8)


 STATISTICS ON BINARY RELATIONS (8th subsystem):
 ***********************************************

        ERROR     BINARY RELATION            REAL
      STRENGTH    OMITTED                    VARIABLES
      --------    ---------------            -----------
      0.012279    (   3,   4)                (   6,  11)
      0.015997    (   1,   4)                (   1,  11)
      0.018204    (   4,   5)                (  11,  13)
      0.023935    (   2,   4)                (   4,  11)
      0.029958    (   2,   3)                (   4,   6)
      0.039631    (   1,   2)                (   1,   4)
      0.053287    (   2,   5)                (   4,  13)
      0.113957    (   1,   3)                (   1,   6)
      0.263907    (   1,   5)                (   1,  13)
      0.286473    (   3,   5)                (   6,  13)


 STATISTICS ON BINARY RELATIONS (9th subsystem):
 ***********************************************

        ERROR     BINARY RELATION            REAL
      STRENGTH    OMITTED                    VARIABLES
      --------    ---------------            -----------
      0.019038    (   3,   4)                (   7,  12)
      0.020049    (   2,   4)                (   2,  12)
      0.020738    (   1,   4)                (   1,  12)
```

```
    0.021145    (   1,   2)          (   1,   2)
    0.026768    (   4,   5)          (  12,  13)
    0.027856    (   2,   3)          (   2,   7)
    0.030240    (   2,   5)          (   2,  13)
    0.072658    (   1,   3)          (   1,   7)
    0.139352    (   3,   5)          (   7,  13)
    0.213198    (   1,   5)          (   1,  13)


STATISTICS ON BINARY RELATIONS (10th subsystem):
************************************************

      ERROR     BINARY RELATION      REAL
    STRENGTH    OMITTED              VARIABLES
    --------    ---------------      -----------
    0.037188    (   1,   3)          (   2,   8)
    0.039851    (   4,   5)          (   9,  13)
    0.051980    (   2,   4)          (   3,   9)
    0.053131    (   1,   2)          (   2,   3)
    0.054757    (   3,   5)          (   8,  13)
    0.058021    (   1,   5)          (   2,  13)
    0.066327    (   2,   3)          (   3,   8)
    0.086346    (   2,   5)          (   3,  13)
    0.089005    (   1,   4)          (   2,   9)
    0.428033    (   3,   4)          (   8,   9)


STATISTICS ON BINARY RELATIONS (11th subsystem):
************************************************

      ERROR     BINARY RELATION      REAL
    STRENGTH    OMITTED              VARIABLES
    --------    ---------------      -----------
    0.050997    (   1,   5)          (   3,  13)
    0.051597    (   1,   3)          (   3,   5)
    0.053362    (   2,   5)          (   4,  13)
    0.057850    (   2,   3)          (   4,   5)
    0.070435    (   1,   4)          (   3,  10)
    0.074054    (   3,   5)          (   5,  13)
    0.086463    (   2,   4)          (   4,  10)
    0.141066    (   3,   4)          (   5,  10)
    0.194925    (   4,   5)          (  10,  13)
    0.443136    (   1,   2)          (   3,   4)


STATISTICS ON BINARY RELATIONS (12th subsystem):
***********************************************

      ERROR     BINARY RELATION      REAL
    STRENGTH    OMITTED              VARIABLES
    --------    ---------------      -----------
```

```
     0.039790    (   2,   3)             (   7,   9)
     0.045809    (   1,   2)             (   4,   7)
     0.065559    (   3,   5)             (   9,  13)
     0.072427    (   1,   5)             (   4,  13)
     0.074612    (   2,   4)             (   7,  10)
     0.087380    (   1,   3)             (   4,   9)
     0.146934    (   2,   5)             (   7,  13)
     0.163374    (   3,   4)             (   9,  10)
     0.163443    (   1,   4)             (   4,  10)
     0.205439    (   4,   5)             (  10,  13)


STATISTICS ON BINARY RELATIONS (13th subsystem):
************************************************

      ERROR      BINARY RELATION         REAL
     STRENGTH    OMITTED                 VARIABLES
     --------    ---------------         -----------
     0.033875    (   2,   4)             (   6,  12)
     0.036787    (   4,   5)             (  12,  13)
     0.038013    (   1,   4)             (   3,  12)
     0.041713    (   3,   4)             (   9,  12)
     0.047758    (   1,   2)             (   3,   6)
     0.060214    (   2,   3)             (   6,   9)
     0.075407    (   1,   5)             (   3,  13)
     0.083718    (   1,   3)             (   3,   9)
     0.095151    (   3,   5)             (   9,  13)
     0.239220    (   2,   5)             (   6,  13)


STATISTICS ON BINARY RELATIONS (14th subsystem):
************************************************

      ERROR      BINARY RELATION         REAL
     STRENGTH    OMITTED                 VARIABLES
     --------    ---------------         -----------
     0.027979    (   1,   4)             (   2,  12)
     0.030086    (   3,   4)             (  11,  12)
     0.032486    (   1,   3)             (   2,  11)
     0.035598    (   3,   5)             (  11,  13)
     0.037756    (   2,   4)             (   5,  12)
     0.038567    (   4,   5)             (  12,  13)
     0.038665    (   2,   3)             (   5,  11)
     0.041903    (   1,   2)             (   2,   5)
     0.044116    (   1,   5)             (   2,  13)
     0.102686    (   2,   5)             (   5,  13)


STATISTICS ON BINARY RELATIONS (15th subsystem):
************************************************
```

```
 ERROR      BINARY RELATION          REAL
STRENGTH    OMITTED                  VARIABLES
--------    ---------------          -----------
0.022136    (   1,   4)              (   1,  11)
0.033928    (   2,   4)              (   8,  11)
0.036568    (   3,   4)              (  10,  11)
0.039507    (   4,   5)              (  11,  13)
0.054311    (   2,   3)              (   8,  10)
0.070287    (   2,   5)              (   8,  13)
0.091243    (   1,   5)              (   1,  13)
0.100869    (   1,   3)              (   1,  10)
0.106511    (   3,   5)              (  10,  13)
0.152063    (   1,   2)              (   1,   8)
```

where variables 1 to 12 represent inputs, and variable 13 stands for the $Lift$ output. Some of these binary relations appear twice or more times, and consequently, their corresponding reconstructions errors should be averaged. At the end, a list is configured with the results obtained for all the variable subsets, which means that all possible binary relations and their corresponding reconstruction errors are in the list.

The list that includes all possible binary relations and their reconstruction errors is, for the first output variable, $Lift$:

```
STATISTICS ON BINARY RELATIONS:
*******************************

ERROR       BINARY RELATION          ERROR       BINARY RELATION
STRENGTH    OMITTED                  STRENGTH    OMITTED
--------    ---------------          --------    ---------------
0.012279    (   6,  11)              0.050696    (   2,   6)
0.018679    (   1,  11)              0.051597    (   3,   5)
0.019038    (   7,  12)              0.054311    (   8,  10)
0.020738    (   1,  12)              0.056373    (   4,  13)
0.022248    (   2,   4)              0.057200    (   7,   8)
0.023935    (   4,  11)              0.063868    (   3,  13)
0.024014    (   2,  12)              0.063897    (   5,   6)
0.027492    (   1,   2)              0.065707    (   4,   8)
0.027601    (   2,   7)              0.067849    (   3,   9)
0.029958    (   4,   6)              0.070435    (   3,  10)
0.030690    (   1,   3)              0.072658    (   1,   7)
0.031641    (  11,  12)              0.075631    (   5,   7)
0.032486    (   2,  11)              0.081660    (   6,   7)
0.033494    (   1,   5)              0.082607    (   3,   8)
0.033734    (   7,  11)              0.087033    (   7,  10)
0.033875    (   6,  12)              0.087380    (   4,   9)
0.034937    (  10,  12)              0.089005    (   2,   9)
0.035301    (  11,  13)              0.090480    (   6,   8)
0.035632    (   8,  11)              0.094474    (   2,  10)
```

```
0.035697    ( 10,  11)
0.037162    (  3,   7)
0.037188    (  2,   8)
0.037203    (  1,   4)
0.037612    ( 12,  13)
0.038013    (  3,  12)
0.038536    (  2,   3)
0.039656    (  2,   5)
0.038665    (  5,  11)
0.039790    (  7,   9)
0.040266    (  9,  12)
0.041266    (  3,  11)
0.042876    (  9,  11)
0.042365    (  5,  12)
0.044053    (  8,  12)
0.044539    (  6,   9)
0.045809    (  4,   7)
0.046700    (  4,  12)
0.047758    (  3,   6)
0.049023    (  2,  13)
0.098827    (  4,   5)
0.100869    (  1,  10)
0.109335    (  1,   6)
0.110667    (  8,  13)
0.117217    (  5,  13)
0.118424    (  1,   9)
0.124953    (  4,  10)
0.130152    (  6,  10)
0.130818    (  5,   9)
0.141066    (  5,  10)
0.148877    (  7,  13)
0.152063    (  1,   8)
0.154886    (  9,  10)
0.175617    (  9,  13)
0.189041    ( 10,  13)
0.197767    (  1,  13)
0.228232    (  5,   8)
0.270750    (  6,  13)
0.428033    (  8,   9)
0.461577    (  3,   4)
```

From this list, the table that relates the twelve input variables with the output variable $y_1$ that is currently being considered can be constructed. This table is depicted in Figure 6.6.

| Input Variables | Optimal Structure Analysis Relation with Output *Lift* |
|---|---|
| 1 | + − − − − − − − − − − − |
| 2 | − − − + − − − − − − − − |
| 3 | − − − − + − − − − − − − |
| 4 | − − − − − + − − − − − − |
| 5 | − − − + − − − − − − − − |
| 6 | + − − − − − − − − − − − |
| 7 | + − − − − − − − − − − − |
| 8 | − − − − + − − − − − − − |
| 9 | − + − − − − − − − − − − |
| 10 | + − − − − − − − − − − − |
| 11 | − − − − − + − − − − − − |
| 12 | − − − − − + − − − − − − |

Figure 6.6: Relation between input variables and output *Lift* of the decomposed $B4$ aircraft system.

Considering the first two qualitative columns of this table, the two following structures of complexities 5 and 6 can be proposed:

$$
\begin{array}{rlr}
(\ u_1,\ \ u_6,\ \ u_7,\ \ u_{10},\ \ y_1) & C = 5, \quad Q = 0.4746 & \\
(\ u_1,\ \ u_6,\ \ u_7,\ \ u_9,\ \ u_{10},\ \ y_1) & C = 6, \quad Q = 0.4812 & (6.27)
\end{array}
$$

As in the case of the linear example, let us revisit, for the first output variable $y_1$, the optimal masks of all complexity levels obtained for the $B4$ aircraft model in Chapter 4. They were:

$$
\begin{array}{rlr}
(\ u_6,\ \ y_1) & C = 2, \quad Q = 0.1531 & \\
(\ u_1,\ \ u_6,\ \ y_1) & C = 3, \quad Q = 0.3060 & \\
(\ u_1,\ \ u_6,\ \ u_7,\ \ y_1) & C = 4, \quad Q = 0.4041 & \\
(\ u_1,\ \ u_5,\ \ u_6,\ \ u_7,\ \ y_1) & C = 5, \quad Q = 0.4952 & \\
(\ u_1,\ \ u_2,\ \ u_5,\ \ u_6,\ \ u_7,\ \ y_1) & C = 6, \quad Q = 0.5088 & (6.28)
\end{array}
$$

As can be seen, the two relationships obtained by Optimal Structure Analysis, namely Relations (6.27), are splendid approximations to the real optimal masks of complexities 5 and 6. Notice that, this time around, the maximum of the quality function seems to be at complexity level 6. Notice also that, in comparison with the linear system example previously explained, the quality values are much lower.

The Correlation Analysis between all the inputs and the $y_1$ output gives the following four relationships:

$$
\begin{array}{rlr}
(\ u_6,\ \ u_7,\ \ y_1) & C = 3, \quad Q = 0.2187 & \\
(\ u_6,\ \ u_7,\ \ u_{10},\ \ y_1) & C = 4, \quad Q = 0.3920 & \\
(\ u_1,\ \ u_6,\ \ u_7,\ \ u_{10},\ \ y_1) & C = 5, \quad Q = 0.4746 & \\
(\ u_1,\ \ u_2,\ \ u_6,\ \ u_7,\ \ u_{10},\ \ y_1) & C = 6, \quad Q = 0.4777 & (6.29)
\end{array}
$$

In this case, Correlation Analysis obtained very good results. They are not as good as those obtained by Optimal Structure Analysis, but they are certainly acceptable.

#### 6.4.3.2 Output Variable *Drag*

Following the same standard procedure for the second output variable *Drag* of the *B*4 aircraft model, the following binary relations list is obtained:[12]

```
STATISTICS ON BINARY RELATIONS:
*******************************

 ERROR       BINARY RELATION         ERROR       BINARY RELATION
 STRENGTH    OMITTED                 STRENGTH    OMITTED
 --------    ---------------         --------    ---------------
 0.015610    (   1,  12)             0.056372    (   7,   8)
 0.015838    (   7,  12)             0.056483    (   5,   7)
 0.016791    (   1,  11)             0.057580    (   6,   9)
 0.021195    (   6,  11)             0.057626    (   4,  12)
 0.022026    (   1,   3)             0.058203    (   3,  14)
 0.024755    (   2,  12)             0.060307    (   2,   6)
 0.025144    (   1,   4)             0.061368    (   4,  14)
 0.025475    (   2,   4)             0.061910    (   6,   8)
 0.026812    (   4,  11)             0.065490    (   3,   9)
 0.027702    (   1,   5)             0.071607    (   3,  10)
 0.029080    (  11,  12)             0.072975    (   4,   8)
 0.029116    (   7,  11)             0.076937    (   6,   7)
 0.029447    (  10,  11)             0.078267    (   6,  10)
 0.029685    (   4,   6)             0.079684    (   9,  14)
 0.032269    (   6,  12)             0.080841    (   3,   8)
 0.032348    (   5,   6)             0.081501    (   2,   9)
 0.032798    (   2,  11)             0.083222    (   7,  10)
 0.033414    (   3,  12)             0.087397    (   1,   6)
 0.034534    (  11,  14)             0.089516    (   2,  14)
 0.035724    (  10,  12)             0.094233    (   4,   9)
 0.035871    (   8,  11)             0.095655    (   1,   7)
 0.038187    (  12,  14)             0.098765    (   2,  10)
 0.038281    (   5,  11)             0.101553    (   8,  14)
 0.038701    (   2,   8)             0.107455    (   5,  14)
 0.039758    (   2,   7)             0.108574    (   4,   5)
 0.040026    (   3,  11)             0.111466    (   1,  10)
 0.041101    (   3,   7)             0.125356    (   5,   9)
 0.041890    (   9,  11)             0.136239    (   4,  10)
 0.042612    (   9,  12)             0.143179    (   5,  10)
 0.043494    (   1,   2)             0.150259    (  10,  14)
 0.043858    (   2,   3)             0.155889    (   1,   8)
 0.044148    (   2,   5)             0.172664    (   9,  10)
 0.044788    (   3,   6)             0.176108    (   1,   9)
 0.045959    (   5,  12)             0.190446    (   7,  14)
 0.047865    (   7,   9)             0.196757    (   6,  14)
 0.052207    (   8,  12)             0.206106    (   1,  14)
 0.054652    (   8,  10)             0.207343    (   5,   8)
```

[12] Due to space limitations, the fifteen subsystem lists are omitted.

```
0.055531    (   4,   7)                0.432133    (   8,   9)
0.055819    (   3,   5)                0.463627    (   3,   4)
```

from which the table of Figure 6.7 can be obtained.

| Variables | Relation with Output |
|---|---|
| 1 | + − − − − − − − − − − − |
| 2 | − + − − − − − − − − − − |
| 3 | − − − + − − − − − − − − |
| 4 | − − − − − + − − − − − − |
| 5 | − − − − + − − − − − − − |
| 6 | + − − − − − − − − − − − |
| 7 | + − − − − − − − − − − − |
| 8 | − − − + − − − − − − − − |
| 9 | − − − − − + − − − − − − |
| 10 | − + − − − − − − − − − − |
| 11 | − − − − + − − − − − − − |
| 12 | − − − − + − − − − − − − |

Figure 6.7: Relation between input variables and output *Drag* of the decomposed *B*4 aircraft system.

Four optimal structures of complexities 4 to 6 can be deduced from this table. They are:

$$\begin{array}{rl} (\ u_1,\ \ u_6,\ \ u_7,\ \ y_2) & C\ =\ 4, \quad Q\ =\ 0.3550 \\ (\ u_1,\ \ u_2,\ \ u_6,\ \ u_7,\ \ y_2) & C\ =\ 5, \quad Q\ =\ 0.3889 \\ (\ u_1,\ \ u_6,\ \ u_7,\ \ u_{10},\ \ y_2) & C\ =\ 5, \quad Q\ =\ 0.3684 \\ (\ u_1,\ \ u_2,\ \ u_6,\ \ u_7,\ \ u_{10},\ \ y_2) & C\ =\ 6, \quad Q\ =\ 0.3977 \end{array} \tag{6.30}$$

The structure of complexity 4 coincides with the optimal mask at that level, and the first structure of complexity 5 as well as the one with complexity 6 are good approximations to the optimal masks at those levels. The optimal masks found for this output were:

$$(\ u_1,\ \ y_2) \qquad C\ =\ 2, \quad Q\ =\ 0.1233$$

$$\begin{array}{rlll} (\,u_1, & u_7, & y_2) & C = 3, \quad Q = 0.2282 \\ (\,u_1, & u_6, & u_7, \; y_2) & C = 4, \quad Q = 0.3550 \\ (\,u_1, & u_6, & u_7, \; u_9, \; y_2) & C = 5, \quad Q = 0.4351 \\ (\,u_1, & u_3, & u_6, \; u_7, \; u_9, \; y_2) & C = 6, \quad Q = 0.4442 \end{array} \tag{6.31}$$

Once more, the mask of highest quality is that of complexity 6. Finally, the structures obtained by Correlation Analysis are:

$$\begin{array}{ll} (\,u_6, \; u_7, \; y_2) & C = 3, \quad Q = 0.1898 \\ (\,u_1, \; u_6, \; u_7, \; y_2) & C = 4, \quad Q = 0.3550 \\ (\,u_2, \; u_6, \; u_7, \; y_2) & C = 4, \quad Q = 0.2028 \\ (\,u_1, \; u_2, \; u_6, \; u_7, \; y_2) & C = 5, \quad Q = 0.2889 \\ (\,u_1, \; u_2, \; u_6, \; u_7, \; u_{11}, \; y_2) & C = 6, \quad Q = 0.3242 \end{array} \tag{6.32}$$

Except for the first relationship of complexity 4 that coincides with the optimal mask at that level, all other relationships exhibit low quality values that are far from the quality values of the optimal masks. For this output variable, it is clear that Optimal Structure Analysis exhibited a considerably better performance than Correlation Analysis.

#### 6.4.3.3 Output Variable $\gamma$ (Flight Path Angle)

Finally, let us take a look at the third output variable, namely $y_3$ (flight path angle). Following the same procedure as for the other two variables, the list of binary relationships and their corresponding reconstruction errors are:

```
STATISTICS ON BINARY RELATIONS:
*******************************

 ERROR       BINARY RELATION            ERROR       BINARY RELATION
 STRENGTH    OMITTED                    STRENGTH    OMITTED
 --------    ---------------            --------    ---------------
 0.020733    (   1,  12)                0.048992    (   3,  10)
 0.021852    (   6,  11)                0.050031    (   7,  10)
 0.022403    (   7,  12)                0.050515    (   4,   7)
 0.024361    (   2,  12)                0.054269    (   3,   6)
```

| | | | |
|---|---|---|---|
| 0.026217 | ( 1, 11) | 0.055951 | ( 3, 5) |
| 0.028838 | ( 4, 11) | 0.059847 | ( 4, 8) |
| 0.029000 | ( 7, 11) | 0.061115 | ( 8, 10) |
| 0.029012 | ( 6, 12) | 0.062183 | ( 1, 15) |
| 0.030582 | ( 2, 5) | 0.066028 | ( 3, 9) |
| 0.030681 | ( 2, 4) | 0.066791 | ( 2, 6) |
| 0.030731 | ( 1, 3) | 0.071973 | ( 1, 5) |
| 0.031216 | ( 1, 2) | 0.072226 | ( 7, 15) |
| 0.031579 | ( 2, 7) | 0.074796 | ( 3, 8) |
| 0.032711 | ( 1, 6) | 0.075264 | ( 2, 9) |
| 0.032714 | ( 1, 7) | 0.075276 | ( 3, 15) |
| 0.033788 | ( 2, 3) | 0.075455 | ( 4, 6) |
| 0.034176 | ( 3, 12) | 0.088896 | ( 4, 5) |
| 0.034517 | ( 7, 8) | 0.093526 | ( 2, 15) |
| 0.034850 | ( 2, 11) | 0.094004 | ( 4, 15) |
| 0.034856 | ( 2, 8) | 0.096833 | ( 4, 9) |
| 0.035382 | ( 11, 12) | 0.098804 | ( 6, 8) |
| 0.035557 | ( 3, 11) | 0.098977 | ( 4, 10) |
| 0.035840 | ( 2, 10) | 0.115381 | ( 8, 15) |
| 0.037580 | ( 3, 7) | 0.116504 | ( 5, 6) |
| 0.038099 | ( 5, 11) | 0.125940 | ( 9, 10) |
| 0.038907 | ( 9, 12) | 0.125469 | ( 6, 9) |
| 0.039256 | ( 10, 12) | 0.128602 | ( 6, 10) |
| 0.039403 | ( 12, 15) | 0.138383 | ( 5, 15) |
| 0.039642 | ( 6, 7) | 0.139785 | ( 9, 15) |
| 0.039852 | ( 11, 15) | 0.141607 | ( 5, 10) |
| 0.040237 | ( 1, 4) | 0.163861 | ( 1, 9) |
| 0.040577 | ( 5, 12) | 0.164669 | ( 1, 8) |
| 0.040976 | ( 10, 11) | 0.167684 | ( 10, 15) |
| 0.042649 | ( 8, 11) | 0.199003 | ( 1, 10) |
| 0.045044 | ( 5, 7) | 0.232827 | ( 5, 8) |
| 0.046903 | ( 9, 11) | 0.233517 | ( 5, 9) |
| 0.046982 | ( 8, 12) | 0.283892 | ( 6, 15) |
| 0.047277 | ( 4, 12) | 0.431398 | ( 8, 9) |
| 0.048733 | ( 7, 9) | 0.455965 | ( 3, 4) |

Applying the first three heuristic recipes to the binary relationships in this list, the table of Figure 6.8 can be constructed.

from which four optimal structures of complexities 4 to 6 can be proposed:

$$
\begin{array}{rllll}
(u_2,\ u_6,\ u_7,\ y_3) & C = 4, & Q = 0.2291 \\
(u_2,\ u_3,\ u_6,\ u_7,\ y_3) & C = 5, & Q = 0.2486 \\
(u_2,\ u_6,\ u_7,\ u_{10},\ y_3) & C = 5, & Q = 0.4230 \\
(u_2,\ u_3,\ u_6,\ u_7,\ u_{10},\ y_3) & C = 6, & Q = 0.4382
\end{array}
\tag{6.33}
$$

| Variables | Relation with Output |
|---|---|
| 1 | − − − − + − − − − − − − |
| 2 | + − − − − − − − − − − − |
| 3 | − + − − − − − − − − − − |
| 4 | − − − + − − − − − − − − |
| 5 | − − − + − − − − − − − − |
| 6 | + − − − − − − − − − − − |
| 7 | + − − − − − − − − − − − |
| 8 | − − − + − − − − − − − − |
| 9 | − − − + − − − − − − − − |
| 10 | − + − − − − − − − − − − |
| 11 | − − − + − − − − − − − − |
| 12 | − − − + − − − − − − − − |

Figure 6.8: Relation between input variables and output $\gamma$ (Flight Path Angle) of the decomposed $B4$ aircraft system.

The optimal masks obtained for this third output were:

$$\begin{array}{rll}
(\ u_6,\ y_3) & C = 2, & Q = 0.1660 \\
(\ u_6,\ u_{10},\ y_3) & C = 3, & Q = 0.3321 \\
(\ u_5,\ u_6,\ u_9,\ y_3) & C = 4, & Q = 0.4033 \\
(\ u_1,\ u_6,\ u_9,\ u_{10},\ y_3) & C = 5, & Q = 0.4304 \\
(\ u_4,\ u_5,\ u_6,\ u_7,\ u_9,\ y_3) & C = 6, & Q = 0.4466
\end{array} \tag{6.34}$$

Notice that the two last relationships (complexities 5 and 6) obtained by Optimal Structure Analysis are very good approximations to the optimal masks at those levels. Let us take a look at the relationships obtained by Correlation Analysis. They are:

$$\begin{array}{rll}
(\ u_6,\ u_7,\ y_3) & C = 3, & Q = 0.1785 \\
(\ u_6,\ u_7,\ u_{10},\ y_3) & C = 4, & Q = 0.2795 \\
(\ u_1,\ u_6,\ u_7,\ u_{10},\ y_3) & C = 5, & Q = 0.3030 \\
(\ u_6,\ u_7,\ u_{10},\ u_{11},\ y_3) & C = 5, & Q = 0.3472 \\
(\ u_1,\ u_6,\ u_7,\ u_{10},\ u_{11},\ y_3) & C = 6, & Q = 0.3650
\end{array} \tag{6.35}$$

As in the case of the previous output variable, the quality of the relationships found by Correlation Analysis is considerably lower than that of the relationships found by the other two methods.

Making a comparison as a whole, it seems clear that, except for the first output variable, Correlation Analysis showed a poor performance. In contrast, Optimal Structure Analysis always proposed good approximations to the real optimal masks.

## 6.5 Conclusions

The satisfactory comparison between Optimal Structure Analysis on the one hand, and Optimal Mask and Correlation Analyses on the other, in the two presented examples, give us confidence that the methodological set of tools presented in this chapter can be applied to a large–scale system as well. We should expect as good results as those obtained with the short and medium–sized systems presented here, since the error introduced by the decomposition of an overall system into subsystems has been shown to be acceptably small.

Reconstruction Analysis, and the heuristic recipes that have been constructed around it, have been shown to constitute a powerful tool for the selection and causal grouping of variables.

Notice that the computational complexity of the Single Refinement algorithm that determines the binary strengths between all pairs of variables is proportional to $n^2$, where $n$ denotes the number of variables in the system. Thus, even in a large–scale system, such as the nuclear reactor that will be presented in Chapter 7, with possibly as many as 500 variables, finding all binary strengths calls for roughly 125,000 structure evaluations, which is a very large, but not unacceptably large, number.

# Chapter 7

# Qualitative Fault Monitoring of a BWR Nuclear Reactor

## 7.1 Introduction

The tools and algorithms that have been presented along this thesis were all guided by the basic goal of proposing a methodology that is capable of safely and reliably fault–monitoring Large–Scale physical systems. Although quantitative fault monitoring schemes have been reported in the literature that perform rather well under certain conditions and for certain types of systems, our emphasis and focus have been on the development of qualitative approaches that mimic somehow the way in which human operators would approach the fault monitoring problem, in the hope to come up with schemes that are simpler and more robust when dealing with truly large–scale physical systems.

This chapter makes use of all the tools that were developed and described in previous chapters, that is, the Fuzzy Inductive Reasoning methodology (Chapter 3), the qualitative Fault Monitoring System (Chapter 4), the Reconstruction Analysis methodology (Chapter 5), and the heuristic recipes for the selection and causal grouping of variables (Chapter 6).

This chapter can be considered in some way an extension to Chapter 4.

The same algorithms for building a qualitative Fault Monitoring System will be used, and the different aspects of fault monitoring, i.e., detection, isolation, characterization, diagnosis, and analysis, will be carried out in the same way. The big difference stems from the way in which the variables are selected and the subsystems identified. Whereas in Chapter 4 these selection and identification processes were performed manually for small–sized systems, and by means of Optimal Mask Analysis for medium–sized systems, here, they will be performed by the Optimal Structure refinement algorithms of the Reconstruction Analysis methodology. In this way, the number of variables of a quantitative Large–Scale System is significantly reduced, and its structure is translated to an equivalent hierarchy of qualitative subsystems.

The variable reduction methodology is one of the most relevant aspects of the tools and algorithms developed in this thesis. We are not just trying to deduce which are the most important variables, but to reduce their number, in order to counter the overload problem.

Although mentioned briefly in the introductory chapter, the adverse effects of the control system architecture on the fault monitoring capabilities have not been met in practice until now. In this chapter, they will become very clear. The controllers that are responsible for plant safety prevent us from exciting the system properly in order to obtain decent data for the qualitative models, the controllers that are responsible for plant stability and disturbance suppression will attempt to filter out the small perturbances that the safety controllers allow us to apply to the system, and the feedback controllers that are responsible for tracking control (the model following controllers) will try their hardest to make us identify them rather than the plant they control.

In the first part of this chapter, a combined Fuzzy Inductive Reasoning and Reconstruction Analysis (FIR/RA) methodology is proposed, and its advantages for fault detection and troubleshooting in large–scale systems are discussed. In the second part of the chapter, the problem of human overload in monitoring the many sensors, controls, and actuators of a complex engineering large–scale system is addressed. A hierarchical automated Fault Monitoring System based on the combined Fuzzy Inductive Reasoning/Reconstruction Analysis methodology, previously explained, for high–level decision making is presented. The FMS will operate in parallel with the traditional channels. It will have no effect other than being able to display its findings to the human operators, and to point out potential problems and their perceived causes. The prospects as well as difficulties in realizing such a monitoring system are analyzed by discussing a prototypical implementation of such a system on a sophisticated quantitative large–scale model of a nuclear boiling water reactor under operational transient conditions.

## 7.2 Combined FIR / RA Methodology

In the previous chapter, Reconstruction Analysis (RA) has been introduced as an alternative tool to Fuzzy Inductive Reasoning (FIR) for building qualitative models, i.e., for identifying high–quality masks. The main advantage of this approach has been that RA can be performed in parts, which enables us to deal with large–scale systems.

In this chapter, the role of RA will be reduced to one of determining sets of relevant variables, rather than using it to find optimal masks directly. Once a sufficiently small number of relevant variables has been identified, FIR will be used to determine the optimal mask relating these variables to each other.

Both methodologies, FIR and RA, are now implemented in SAPS–II, a Fortran–coded system with interfaces to Matlab [MathWorks, 1992], CTRL–C [SCT, 1985], and ACSL [MGA, 1991].

In the light of what has been said along this thesis, the combination of FIR and RA can best be understood as a partly heuristic methodology for qualitative modeling and simulation of continuous–time processes. The main capabilities of this combined methodology are listed below and can be seen in Figure 7.1. They are:

- Conversion of quantitative information into qualitative triples that include class values, membership function values, and side values; the conversion is done in such a way that the quantitative information can be regenerated from the qualitative triples without any loss of information.

- Determination of significant sets of qualitative variables that can be treated as subsystems; the subsystems are found by means of temporal causality analysis and refinement procedures.

- Automatic generation of the best possible qualitative models, i.e., those with the best forecasting capabilities, for each of the previously identified subsystems.

- Construction of a hierarchy of qualitative models whereby each node represents an inductive reasoner, i.e., construction of a hierarchy of inductive reasoners.

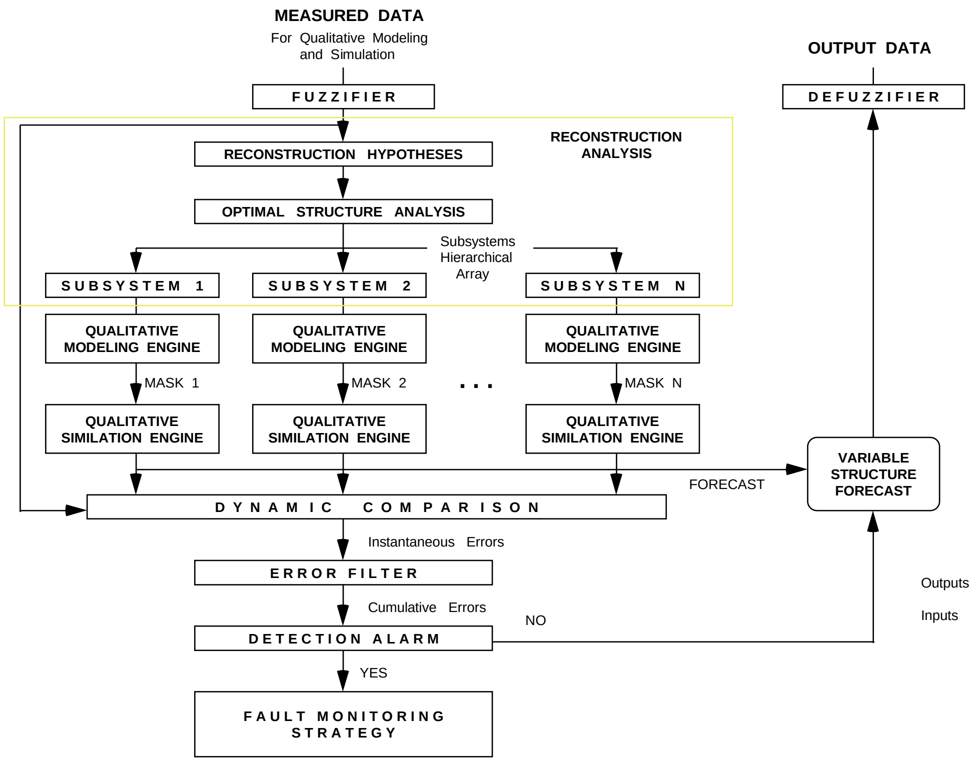


Figure 7.1: Combined FIR/RA methodology

- Qualitative simulation through inductive reasoning using all the previously found qualitative models in the hierarchy.

- Recognition of the structural state that the system is in at any point in time.

- Regeneration of quantitative results from qualitative results, if so desired, that can subsequently be used by either human operators or automatic controllers.

The capabilities of the mixed methodology as they relate to fault detection are:

- Reduction of a quantitative large–scale system to an equivalent hierarchy of qualitative subsystems.

- Comparison at sampling time between the qualitative results obtained from the inductive reasoners and the actual quantitative values obtained from the real physical system or a numerical model thereof to detect discrepancies that would indicate some sort of misbehavior having occurred.

- Inspection of the upper levels of the reasoning hierarchy to hypothesize about which subsystem might have been responsible for producing the encountered behavioral anomaly.

- Characterization of detected misbehaviors in such a way that they can be incorporated into a library of anomalous behaviors.

- Characterization and learning of the patterns that produced the detected misbehavior in such a way that they can be easily identified and prevented.

- Diagnosis of the possible causes of observed anomalous behaviors in such a way that they can also be incorporated in a library of observed faults. The idea is to be able to recognize anomalies (once they have been properly characterized and stored away in the library of anomalous behaviors) easily identified as soon as they occur again, and relate them to their most likely causes as they are already stored in the library of faults.

## 7.3 Control, Monitoring, and Safety of Nuclear Reactors

It is not our intention to provide full details about the operation of commercial nuclear reactors. However, some basic concepts should be explained in order to introduce the control and safety problems related to the normal and abnormal operation of such reactors, and the way in which faults can be prevented and mitigated by means of Fault Monitoring Systems.

A nuclear reactor consists of a core, containing the fuel, normally uranium, in which heat is released from the fission reactions of the uranium atoms, mainly as a result of the absorption of slow neutrons. Inside the core, there is also a *moderator*, the function of which is that of slowing down the high–energy neutrons liberated in the fission reaction, mainly by elastic scattering. The best moderators are materials consisting of elements of low mass number such as ordinary water, heavy water, and graphite. Surrounding the core there is a *reflector*, the purpose of which is to decrease the loss of neutrons by scattering back many of those that have escaped. Generally, the same material used as moderator serves also as reflector.

The processes that take place in the core are the following: a low–energy neutron may be absorbed by a uranium atom. If this happens, that atom will suffer a fission reaction producing two new atoms of other elements, two or three high–energy neutrons, and pure energy. If at least one of the neutrons produced can be moderated and then absorbed by another uranium atom, then another fission will be produced, and so on. Thus, the fission reaction is autosustained, and the reactor is said to have become "critical." The neutron–multiplying property is called *reactivity*.

A coolant circulating through the core is needed to extract the generated heat and transfer it to a steam generator from which steam can then be used in a turbine/generator pair to produce electricity. In most reactors, the coolant serves also as moderator and consists of ordinary water. There exist two different reactor types that fall into this category: a) those that use highly pressurized liquid water where boiling conditions are not allowed directly in the core, but take place only in an external low–pressure steam generator, these are called *Pressurized Water Reactors (PWRs)*; and b) those that use low pressure water where boiling conditions are allowed directly in the core, in which case there is no need for the external steam generator component, since the core itself is the steam generator, these are called *Boiling Water Reactors (BWRs)*. The reactor model that will be used here is of this latter type. These BWR reactors needs forced coolant recirculation to have stable operating conditions. As can be seen in Figure 7.4, two recirculation circuits and several jet pumps provide this condition.

In Figure 7.2, a simplified diagram of the nuclear and non–nuclear components of a nuclear power plant equipped with a BWR reactor are shown. As in any power plant, the heat produced by fuel burn–up is used to boil water and produce high pressure and speed steam. The steam is used to propel the turbine that in turn moves the electric generator. The resulting low pressure steam is then condensated and pumped once more into the reactor.

The unique feature of a nuclear power plant that is distinct from other power generating facilities, is the presence of large amounts of radioactive materials, primarily the fission products. Thus, the central safety problem in the design and operation of a nuclear plant is to assure that these radioactive elements will remain safely confined at all times, no matter what kind of expected or unexpected conditions are present.

The radioactive elements are confined inside the fuel bars at the core of the nuclear reactor. The core itself is confined inside the reactor vessel, and this one in turn is confined inside a primary containment building. Finally, this building is also confined inside a secondary containment building. The

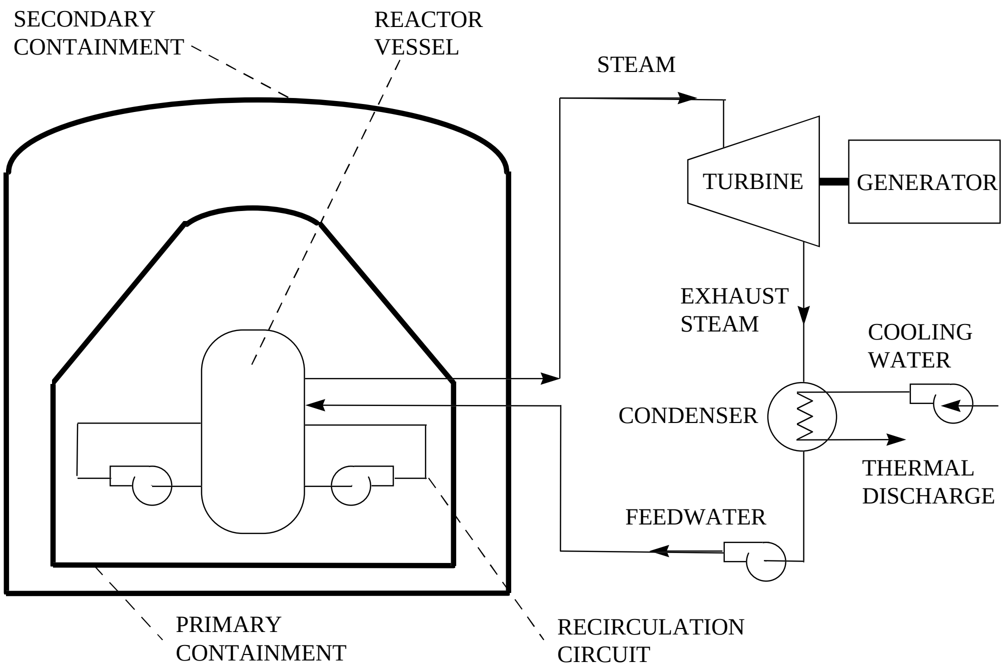


Figure 7.2: Simplified diagram of a BWR nuclear power plant.

only way in which radioactive materials can scape from the core is due to high temperature, i.e., a loss of coolant, in such a way that the hole core melts down forming a non–refrigerable geometry. The whole reactor and the whole plant are designed on the basis of this worst accident, firstly, preventing it from happening, and secondly, providing features to mitigate it once it has happened. For these reasons, the control of a nuclear power plant must be precise, reliable, robust, and safe.

The most important control system of the reactor is the Reactivity Control. It works in the following way. The rate of heat generation is proportional to the nuclear fission rate and is determined by the neutron density. Control, including startup, operation at any desired power level, and shutdown, is thus achieved by varying the neutron density in the core. This is accomplished by moving *control rods* of a material that absorbs neutrons readily (boron or cadmium). Insertion of control rods into the core results in a decrease in the reactivity and consequently, in a decrease of the neutron density. Hence, the reactor power level is reduced. Withdrawal of the control rods, on the other hand, is accompanied by an increase in the reactivity, and thus in the neutron density and power level. Thus, reactivity turns out to be the main control variable, and its control turns out to be the reactor stability control [Hetrick, 1971].

Reactivity can be affected in positive or negative ways by several factors, the most important of which are: neutron flux, coolant temperature, moderator boiling, fission products poisoning, recirculation flow rate, and control rods [Duderstadt and Hamilton, 1976]. Some of them are produced as a result of the reactor operation itself and the fuel burn–up, while others are induced by the operators in order to change the operating conditions, or by malfunctions in one or several reactor components.

In the normal operation of a nuclear reactor, the functions of the control systems may be divided broadly into three phases: startup, power operation, and shutdown. There are three different startup conditions: from "cold" condition, with a low pressure and low temperature reactor; from "hot standby" condition, with a high pressure and temperature reactor; and from "poisoned" condition. In either case, the control of the reactor requires careful attention during the startup phase, in order to prevent the possibility of an accident due to a power excursion, i.e., a tremendous increase in the reactivity of a localized core zone. Startup is accomplished by a slow withdrawal of the control rods, and by a slow increase of the recirculation flow.

During power operation, the required power level is normally maintained essentially constant by automatic controllers. An error signal, representing the difference between a particular measured operating parameter and the desired value of that parameter, is fed to a servomechanism, which then causes the necessary actions, e.g., control rod motion and/or change in recirculation flow rate, to bring the error back to zero.

Except for emergency situations, the reactor is shutdown by the slow insertion of the control rods and through recirculation pump slowdown. In principle, rapid shutdown is possible, but it is usually avoided in order to decrease mechanical stress that would result from the drastic associated temperature gradients. As in the startup phase, there are three different shutdown conditions: to hot standby, to cool condition, and to minimum allowable power condition.

All power reactors utilize both human and automatic control systems during startup, normal operation, shutdown, and emergencies. The conditions under which one control mode is preferred over the other have been explained in Chapter 4, and depend on the circumstances. In Figure 7.3, both an operator loop and an automatic loop form part of the reactor control system. A third loop corresponding to the Reactor Protection System has been introduced as the ultimate automatic protection that overrides all other human and/or

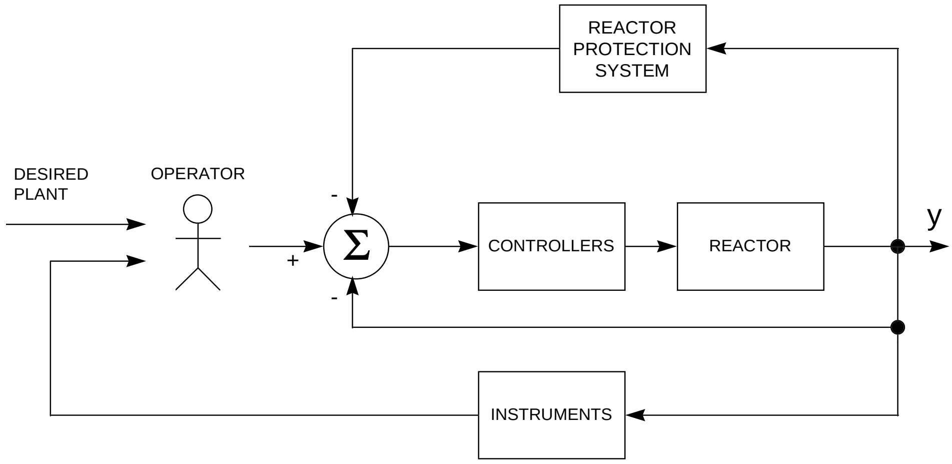


Figure 7.3: Reactor with human, automatic, and protection control systems.

automatic controllers.

Several of the reactor variables are associated, both in reality and in the quantitative reactor model used in this thesis, with built–in control loops and safety systems. Among them, the most important ones are [Glasstone and Sesonske, 1994]:

– **Pressure Control.** This control system continuously compares the amount of steam leaving the vessel with the amount of feedwater entering the vessel. It is also influenced by the turbine and steam line pressures.

– **Feedwater Control.** It is based on a comparison between the desired and actual water levels inside the vessel. It also takes into account the feedwater temperature.

– **Power Control.** This control accounts for small variations in reactivity, and consequently in the desired output power level, by changing the recirculation flow rate.

– **Reactivity Control.** As has been previously explained, the reactivity control serves to account for small variations in the neutron flux of the core, and consequently in the desired output power level, by adjusting the depth of insertion of the control rods.

– **Reactor Protection System.** It is not a control system but a protection system, in charge of the ultimate defense of the reactor, i.e., Emergency Scram, Emergency Core Cooling Systems, etc.

All of the above functions are preformed by automatic controllers. However, simulator sessions and nuclear past accidents have demonstrated that one of the most critical and unreliable factors in the control and safety of a nuclear reactor relates to the operators' override capabilities, i.e., their capabilities of interfering with the automatic control systems [Glasstone and Sesonske, 1994]. This process is highly affected by the information overload problem stated and explained in Chapters 1, 2, and 4. It could indeed be argued that in no other engineering activity human operators are as much affected by overload problems, as in the case of the control room of a nuclear power plant, specially under emergency conditions.

Since operators are in charge of unexpected and emergency conditions, all modern nuclear power plants include various features, beside from the regular instrumentation and controls, to support the operator crew in this endeavor. Most of these features are related to Fault Monitoring Systems requiring either quantitative or qualitative simulation.

Quantitative simulation of nuclear power plants is widely used. There exist not only static simulation codes for nuclear fuel analysis, but also complex dynamic codes for licensing, evaluation, and validation of operational and non–operational transients, as well as for the evaluation of severe emergencies, such as the so–called design–based accident, the worst–case scenario (loss of coolant), for the prevention of which all reactor safety systems are primarily designed. Recently, fast compact numerical real–time simulators have been installed in the control rooms of many nuclear power plants for the prediction of those reactor variables that are most useful for the early detection of abnormal conditions.

However, not just quantitative but also qualitative methods have been introduced during recent years to the control rooms of nuclear power stations, and their application has been diversified from knowledge–based advice systems, to Neural Network–based malfunction pattern recognition algorithms, as well as fuzzy controllers.

This has happened for three primary reasons: a) the need for a highly qualified and well trained staff to operate the plants calls for sophisticated *training simulators*, b) the overall complexity of the system and the difficulties in properly assessing what is going on in the plant at any point in time call for highly reliable simulation codes used as *on–line analysis tools*, and c) pressure exercised by public opinion on safety aspects of the power plants calls for *redundant control and safety systems.*

Some of these new features that were recently introduced in nuclear power plant control rooms to aid human operators in their decision making processes are [Glasstone and Sesonske, 1994]:

- A numerical simplified simulation system of the reactor capable of at least ten times real time to perform basic nuclear kinetics and thermohydraulics predictions.
- A numerical Fault Monitoring System for the 17 most significant plant variables [NSAC–EPRI, 1980].
- A qualitative simulation system that shows the tendencies, i.e., derivatives, of the most significant plant variables.
- A qualitative Expert System to relate actual symptoms back to previously learned emergencies to provide the operators with guidance regarding measures to be taken.
- A qualitative Expert System to relate developing emergencies back to previously learned emergency operation procedures, by pointing out appropriate sections and pages in the Emergency Operations Manual.
- A qualitative Neural Network Fault Monitoring System with early warning functions for recognition of transient patterns.

In subsequent sections of this chapter, a qualitative combined FIR/RA methodology–based Fault Monitoring System to tackle the information overload problem during operational transient conditions will be presented.

## 7.4 The Quantitative Large–Scale Model

The quantitative (numerical) nuclear reactor model used in this project has been developed as an infrastucture project of the Nuclear Energy Department at the Electric Research Institute in Cuernavaca, México. This model has served as the keystone for the development of the core model for the Laguna Verde Nuclear Power Station Full–Scope Training Simulator [Ramos *et al.*, 1991], and for the reactor model of the Graphic and Interactive Compact Simulator [Morales *et al.*, 1985]. This model has also been used for validation of the results obtained by a Probabilistic Risk Analysis model of the same reactor, and for thermohydraulic effects research.

The model itself is a complex differential equation model of $79^{\text{th}}$ order. It is composed of 79 state variables, 79 initial values for those state variables, 373 defined variables, and 216 parameters, which gives a total of 747 variables, representing the Boiling Water Reactor of the Laguna Verde Nuclear Power Station Unit 1, at Veracruz, México.

The model, intended for operational transient validation and thermohydraulic calculations, is very detailed in the nuclear kinetics, but even more so in the core thermohydraulics. With respect to the nuclear kinetics, it includes a point approximation with six delayed neutron precursor groups and with three reactivity feedback mechanisms due to water boiling (the voids formation), temperature change (Doppler effect), and the neutron absorber mechanisms (Scram).

With respect to the thermohydraulics, the heat generation process considers one average fuel bar with four radial nodes modeled by finite difference approximations including the residual heat. The thermohydraulics are simulated using a unidimensional axial model that takes into account all boiling phases and two phase flows, even in opposite directions. The heat conduction model has one axial node associated with each axial node of the thermohydraulics and is composed of two radial nodes, one for the fuel and the other for the cladding.

The model of the reactor vessel includes the lower plenum region, the core, the upper plenum region, the steam separators, the dome region, the bulk water region, the downcomer region, and two independent recirculation circuits as can be seen in Figure 7.4.[1] Considering all these regions, mass, energy, and pressure balances are carried out taking also into account the amount of produced steam leaving the vessel, and the amount of feedwater entering the vessel.

To perform these balances, the dome of the vessel is considered as divided into three regions, namely the steam region, the steam and saturated water region, and the returning water region, as they are shown in Figure 7.5.[2]

Two independent recirculation circuits are modeled providing the possibility for one or both of them to trip. In this way, multiple operational transients in which only one loop fails can be simulated. Each recirculation loop includes a fully modeled motor pump (with all its operating regimes) and a very detailed generalized jet pump. Since each circuit is composed of several pressure nodes, as shown in Figure 7.6,[3] the effects of reverse flow can also be simulated.

The reactor model also includes a submodel of the steam line without

[1] This figure has been taken from [Ramos, 1991].
[2] Idem.
[3] Idem.

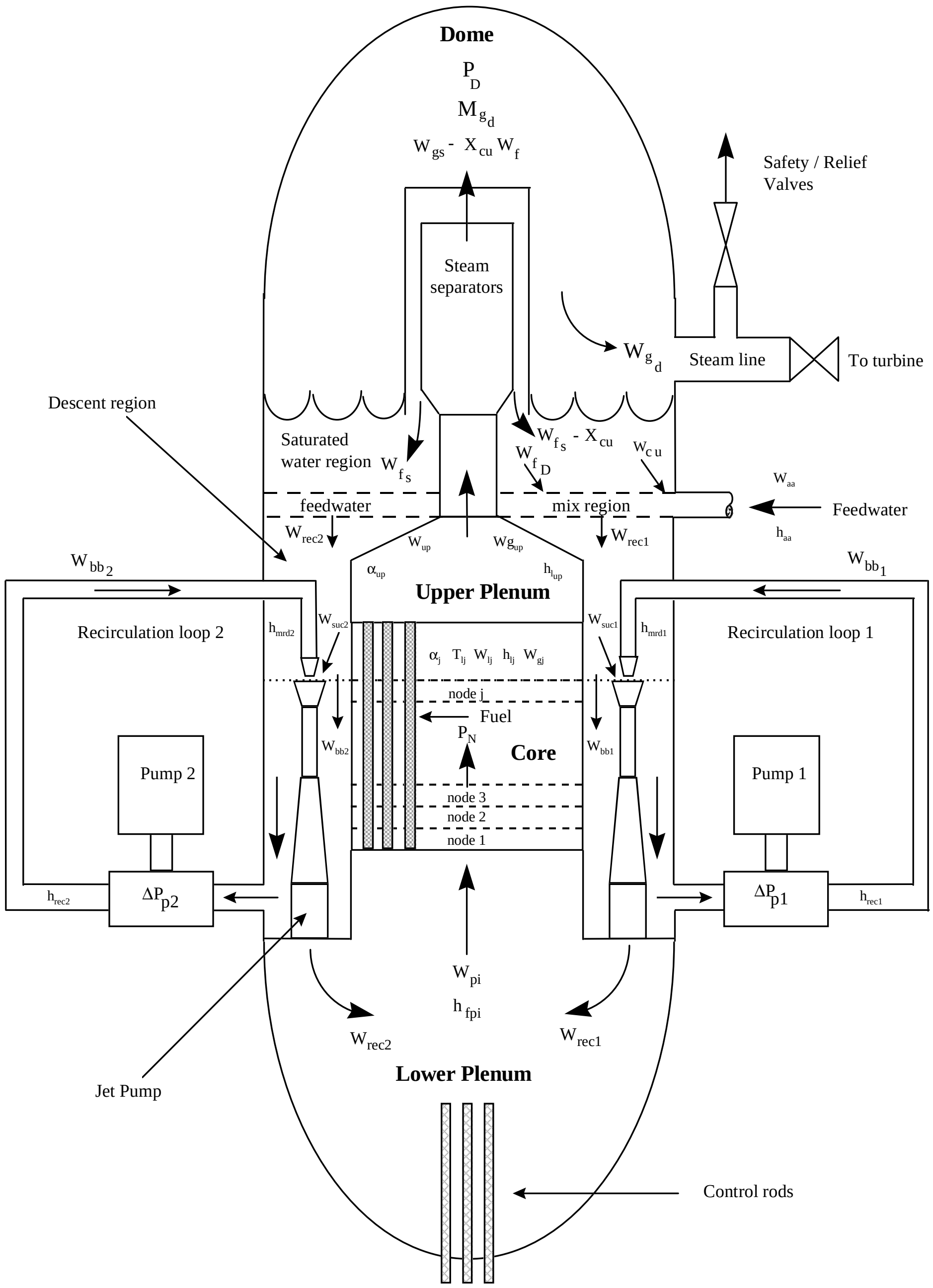


Figure 7.4: Simplified scheme of the nuclear reactor model.

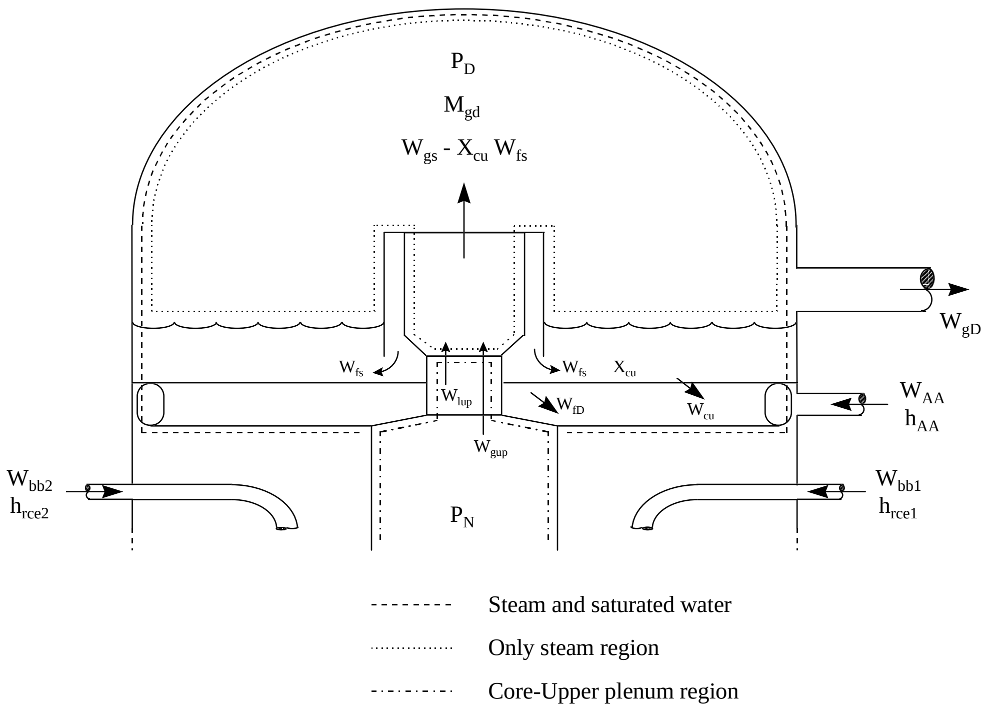


Figure 7.5: Upper plenum, steam separators, and dome of the reactor.

capacitive properties, i.e., without a pressure node. Along the steam line, several control and safety valves are placed. Belonging to the former category, there are the pressure control valves at the turbine inlet (only one is modeled since only one generalized steam line is present); and belonging to the latter class, there are the bypass channel (one modeled), as well as relief (ten modeled), safety (ten modeled), and isolation (one modeled) valves.

One generalized feedwater pump (with all its operation regimes) and one generalized feedwater preheater are modeled in order to include the feedwater controller.

The control of this reactor is accomplished by means of three simulated controllers. They are:

i) **Pressure Controller**. This controller is attached to the steam valves at the turbine inlet, and its model takes into account, on the one hand, the difference between the amount of steam leaving the vessel and its reference value, and on the other hand, the difference between the pressure at the turbine inlet and its reference value.

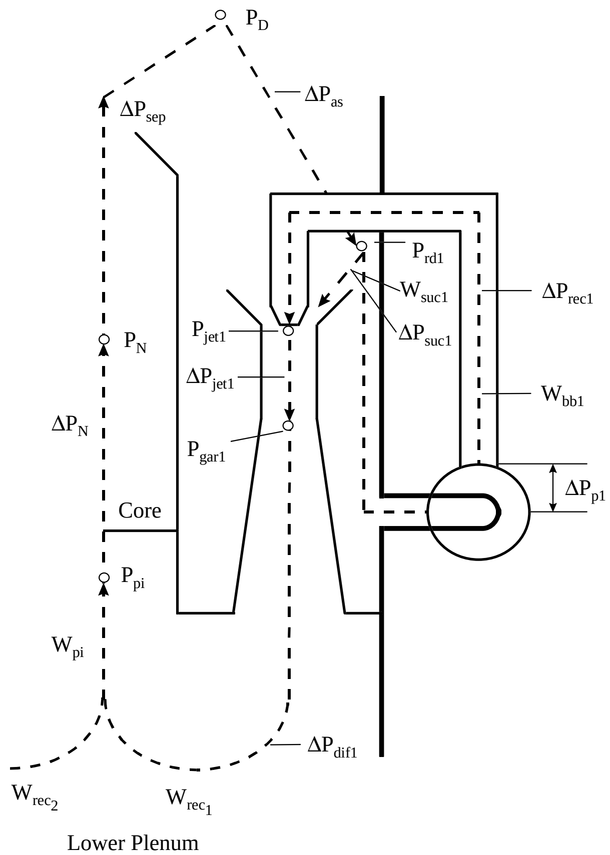


Figure 7.6: Recirculation Circuit.

ii) **Feedwater Controller**. This controller, also known as the level controller, is attached to the feedwater valve, and its model takes into account, on the one hand, the difference between the amount of steam leaving the vessel and the amount of feedwater entering the vessel, and on the other hand, the difference between the reactor water level and its reference value.

iii) **Recirculation Controller**. This controller, also known as the power controller, is attached to the recirculation valves (one for each loop). Its model takes into account, on the one hand, the difference between the power level and its reference value, and on the other hand, the difference between the amount of water entering the core and its reference value.

As will be explained subsequently in this chapter, some of these controllers were disabled to carry out the reactor excitement processes.

All the automatic transient mitigation and emergency actions and alarms are modeled in the **Reactor Protection System**. This system is in charge of the automatic opening and closing of the steam line isolation, relief, safety, and bypass channel valves. It can also reduce speed or trip one or both recirculation pumps and the feedwater pump. Finally, its most important action is the emergency shutdown. No control rods are simulated, instead the equivalent negative reactivity is inserted directly into the core through the feedback reactivity variable $Scram$. The same applies for the slow insertion or withdrawal of control rods during normal startup and shutdown.

The plant simulator can be used during all phases of plant operation (startup, steady–state, and shutdown), and it can also be used for both normal and abnormal plant operation, i.e., during operational transients. Since the mathematical description of the model takes more than 70 pages, the interested reader is friendly invited to consult the references. Full details of this nuclear reactor model have been published in [Ramos, 1991; Ramos *et al.*, 1991; Ramos and de Albornoz, 1986].

Two operational transients were built into the reactor model to alter its normal *full–power steady–state* behavior. These transients are simulated by modifying, at a scheduled time instant, a set of *ad hoc* parameters that the model includes for such purposes.

The simulated emergencies are absolutely realistic in terms of what might happen to the real reactor, this being in contrast with the aircraft emergencies simulated in Chapter 4 that were not realistic. In that case (aircraft), this was not of vital importance since the experiment was intended for demonstrating the validity of the FIR–based FMS approach only. However, in the nuclear reactor case, it is important to show that a FIR/RA–based FMS can be used to aid the human operators in their decision making process of a very complex Large–Scale System.

### 7.4.1 Operational Transients

In a Boiling Water Nuclear Reactor (BWR), safety is related to two main considerations: a) radioactivity confinement, and b) nuclear fuel cooling. These two different, but closely related, factors influence the whole plant design. The situation where the nuclear fuel can no longer be properly cooled is known as the "design–based accident." The reactor and the whole plant are designed to prevent this possibility. Beside this worst possible accident, there are lots of events with small to medium danger levels known as "transients."

A transient is an anomaly, misbehavior, or fault that causes some malfunction in the reactor or somewhere in the plant. Anomalies in a (BWR) can be divided into several categories depending on their probability, frequency of occurrence, or potential danger. For the purposes of this dissertation, we will use the latter approach. Accordingly, transients in a BWR can be classified into three categories depending on their potential danger. They are, from minor to major danger:

i) **Operational Transients**. An operational transient is a fault that does not produce a significant refrigerant leakage, and/or does not trigger any emergency reactor cooling system. The transient is usually very fast, and may be caused by internal factors such as human errors or equipment malfunctions, or by external factors such as black–out or generator load rejection. This kind of transients is well known, and human and automatic controllers are well trained to deal with them. The basic considerations when dealing with operational transients are safety and a rapid restart of the full power operation. In most cases, human operators do not know what is happening until the reactor is automatically shut down to the *hot standby condition*[4] or driven to a new steady state, i.e., until they perform a post–mortem analysis. Therefore, operators normally do obstain from taking any immediate action.

ii) **Non–Operational Transients**. These are faults that do produce a refrigerant leakage and/or trigger one or more emergency cooling systems and/or impede a rapid restart of the full power operation. The transient is normally caused by a combination of events such as, for example, operational transients, human errors, or equipment malfunctions. These types of transients are mostly discovered by means of Probabilistic Risk Analysis, and analyzed by means of Decision Trees. The most common ones are well known to operators. In most cases, human operators do need to take some actions to complement those of the automatic controllers, with the purpose of avoiding that the non–operational transients become emergencies.

iii) **Emergency Operations**. Non–Operational transients that cannot be rapidly and safely mitigated become emergency operations. In this

[4]In the hot standby condition, reactor pressure and temperature values are still near to those of operation. This permits the reactor to resume operation within minutes or very few hours. In contrast, if the reactor has been depressurized and cooled, operation cannot be resumed until several hours or even days later.

condition, almost all actions are taken manually by the human operators, and the only concern is radioactivity confinement. Since in an emergency situation, it takes some time before human operators are able to realize what is really going on, they must always follow the *Emergency Operation Procedures* no matter which fault has occurred. These procedures are intended for preserving nuclear fuel integrity[5] and are based exclusively on the monitoring of the main reactor variables (pressure, water level, thermal power, temperature, etc.), i.e., operators should forget about everything but the main reactor variables.

Since the numerical model is intended for simulation of normal operation and operational transients, only those will be considered in this analysis.

Operational transients can be classified with respect to their ultimate influence on the thermohydraulics and the neutronics of the reactor. Following [Lamarsh, 1983], their classification is:

i) Pressure Increase.

- Turbine trip.
- Generator trip.
- Sudden closure of the Main Steam Isolation valves.

ii) Pressure Decrease.

- Pressure regulator failure.
- Sudden opening of a Safety/Relief valve.

iii) Refrigerant Flow Increase.

- Recirculation pumps control failure.
- Sudden speed–up of a recirculation pump.

iv) Refrigerant Flow Decrease.

- Recirculation pumps control failure.

[5] When the heat produced by the nuclear fuel elements cannot be extracted from the core, the temperature and the power density will rise until levels at which those fuel elements may melt. A molten core will not only liberate all radioactive fission products, but it may produce a non–refrigerable nuclear fuel geometry. This is the worst accident that can ever happen in a nuclear reactor.

- Recirculation pump trip.
- Sudden slow–down of a recirculation pump.

v) Refrigerant Temperature Increase.

- Feedwater control failure.
- Feedwater pumps trip.

vi) Refrigerant Temperature Decrease.

- Feedwater control failure.
- Feedwater preheaters loss.

vii) Neutron Flux Increase.

- Control rod drop.

viii) Neutron Flux Decrease.

- Sudden total or partial Scram.

## 7.5 The Qualitative Model

Three qualitative models of the reactor will be created. One for the *full–power steady–state* normal operation, one for the *recirculation pump trip* operational transient, and one more for the *feedwater control failure* operational transient. As a matter of fact, there will be four models instead of three, since the latter transient is composed of various other transients, one of which needs also to be modeled and characterized, namely the *Scram* transient.

In the construction of these models, four modeling difficulties will be faced:

i) **Excitation of the reactor model.** It is fortunately quite impossible to excite a nuclear reactor (or even a sophisticated quantitative model thereof) in such a way that all frequencies are richly represented in the input/output behavior of the excited subsystem for the purpose of the best possible identification of an optimal mask. Sophisticated quantitative models of nuclear reactors, as the one presented here, are very detailed due to their training, validation or licensing purpose, and are designed to reflect the behavior of the real system including its automatic responses to emergency situations. For these reasons, small deviations in the expected behavior will be interpreted as anomalies that could trigger an emergency process, which may eventually lead to a shutdown of the reactor. The reactor will then behave quite differently due to the transient in comparison with its behaviour under normal operating conditions. The very test signals that are needed to excite the quantitative model for the identification of its qualitative counterpart are easily interpreted as transients by the reactor simulator. The quantitative reactor model trips over its own shoe–lace, so to speak.

ii) **Variable Selection for each operating condition.** The previously mentioned limitation on the number of input variables that a human can simultaneously process is shared by most automated reasoning algorithms. A single sequential reasoning algorithm turns slow and unwieldy when being requested to cope with too many facts at the same time. Inductive reasoners (like neural networks) may be quite efficient once they are properly trained because they are inherently parallel in nature, but their re-learning abilities degenerate quickly as the number of input variables (i.e., the dimension of their reasoning space) grows. It is therefore very important to select a minimum set of variables that meaningfully represent the system to be reasoned about, and that can be handled by the inductive reasoner in an efficient manner.

iii) **Change of meaningful variables during different stages of the same operating condition.** This problem copes with the variable selection and subsystem identification for the changing characteristics of the accident conditions, and for the post-accident conditions. Most of the operational transients end with a reactor emergency procedure that may result in a reactor shutdown. Once an emergency procedure has been initiated, there occurs a dramatic change in the minimum set of meaningful variables needed to represent the system, i.e., the valid set of variables used to describe the system prior to the transient is not the same set that is needed to represent the system during and following the transient.

iv) **Subsystems identification in a closed–loop environment.** The problem is one that has haunted for decades the researchers who are working in the identification of control systems. When identifying a subsystem within a feedback structure, it is desirable to break the system open, since otherwise, it is never fully clear whether it is really the subsystem itself that has been identified, or whether it might not be the feedback loop around the subsystem that, by its own nature, constitutes another subsystem with the same extraneous variables but exchanged inputs and outputs, or maybe a combination of both. However, inductive models are only valid within a limited range around an operating point or operating trajectory. By opening up the feedback loop, the subsystem is likely to exhibit behavioral patterns that resemble little those of the closed–loop operation. Thus, the "learned" qualitative model will be of little or no use for predicting the behavior of the subsystem in closed–loop operation. It is thus essential that the feedback loops are kept intact when identifying the subsystems. Consequently, the modeler has to live with the aforementioned difficulties.

The way in which these problems have been solved will be explained in full in the corresponding next sections.

### 7.5.1 Full–Power Steady State

A nuclear reactor is intended for full–power operation during the maximum possible time. When driven to full power, our numerical model stabilizes at 100.4 % of designed thermal output power. At this power level, with recirculation pumps operating at 100 % of their speed (80 % of their pumping capacity), and control rods fully withdrawn,[6] the reactor reaches a steady state that can be maintained during infinite time in the model, or during almost a year in a real plant, unless an unscheduled event (transient) occurs.

As has been explained before, the model includes the Reactor Protection System that is in charge of all automatic response actions when the main variables surpass critical values. This has the effect that a small deviation from the steady state will be immediately detected and pointed out to the operators through a visual alarm, and that a significant deviation will be countermeasured automatically. This gives us little chance to excite the reactor model outside the range of regularly allowed operations.

Beside normal full–power operation, two representative transients were numerically simulated. They were: a) a *recirculation pump trip*, and b) a *feedwater control failure*. These transients have been selected because neither of them produces an immediate emergency stop (Scram). The first

[6]Since the numerical model is not intended for nuclear fuel calculations, but for thermohydraulic analysis and operational transients validation, it is supposed that the reactivity due to control rods must be always zero during normal operation, i.e., the full power condition is reached with the control rods completely withdrawn.

one, recirculation pump trip, will be used to show how a small and well known transient produces important and very fast variations in the main reactor parameters, which makes it impossible for the operators to perform a characterization and identification of the transient in real–time. The second transient, feedwater control failure, is much more involved and cannot be considered a small transient, since practically all reactor systems take part in it. It is intended to demonstrate how a qualitative Fault Monitoring Systems is capable of characterizing and identifying a transient before automatic emergency actions are triggered.

#### 7.5.1.1 Excitation of the Reactor

The limitations with respect to exciting the reactor model have been stated earlier. The excitation process is limited to forcing the system to perform small oscillations around its steady state. To this end, two state variables related to the pressure control, $W_{G_{ref}}$, and to the feedwater control, $W_{FW_{ref}}$, will be used to excite the model. The excitation input variables are perturbations affecting the reference values of two control variables of the model. The reference values of these variables are preset such that the simulation starts out at some predetermined power level. In this case, full power level values are used. A change in any of these two variables will perturb the model, forcing it into a new steady state or into a transient condition.

Exciting the reactor model with random binary or ternary noise is out of the question, since the induced effects of such a violent change on the thermohydraulics will immediately trigger one or more emergency systems. Instead, the model will be excited with harmonic functions of short periods and amplitudes. Following the terminology used in the aircraft example of Chapter 4, this way of operating the model is known as the "shaken reactor." The amplitude values are chosen in such a way that the reactor reaction does not produce any alarm or emergency condition, which means that only small oscillations around the steady state are allowed. Although this restriction can be viewed, in principle, as a shortcoming since only those behavioral states in the vicinity of the steady state can be captured, i.e., predicted, it, in fact, makes the transient detection process easier since even small deviations from the expected trajectories are quickly recognized as anomalous behaviors.

The initial values of the two chosen variables, the control reference value of the amount of steam leaving the reactor vessel ($W_{G_{ref}}$) and the control reference value of the amount of feedwater entering the reactor vessel ($W_{FW_{ref}}$) are 1085.41 $kg/sec$ each. Logically, they have the same value since the total amount of water in the reactor during normal operation should be kept constant.

The maximum allowed amplitudes of the harmonic short period excitation signals are:

$$\Delta W_{G_{ref}} = 1085.41 \pm 89.5 \ \frac{kg}{sec} \quad (7.1)$$

$$\Delta W_{FW_{ref}} = 1085.41 \pm 42.0 \ \frac{kg}{sec} \quad (7.2)$$

These are applied directly to the differential equations model.

Due to the complexity of the model and its highly non–linear behavior, it is not an easy matter to compute the smallest and largest eigenfrequencies, i.e., the slowest and fastest time constants. To start, we know from preceding sections that the reactor model is divided into subsystems with different degrees of stiffness. This makes some subsystems react faster than others. The simulation uses a variable–step integration algorithm. The shortest integration step observed was 0.0125 *sec* and corresponds to the core nuclear kinetics, whereas the largest integration step was 0.517 *sec* corresponding to the pumped water and to the mixed enthalpy equations.

If these integration steps are interpreted as approximations of the time constants, $T_{fast}$ and $T_{settling}$, of the model, following Equations (3.13) and (3.14), the sampling rate $\delta t$ should be chosen as 0.006 *sec* and the mask depth should be 83. However, since neither the *full–power steady state* nor the operational transients that will be simulated are intended for nuclear elements analysis, the nuclear kinetics integration step can be substituted by the shortest thermohydraulics integration step, which happens to be 0.228 *sec*. Following the same aforementioned equations, the sampling rate will be 0.114 *sec* and the mask depth will be 5. This drastic reduction in the mask depth (from 83 to 5) leads also to a drastic reduction in the mask complexity.[7]

With these excitation parameters, a simulation was carried out along 2500 sampling points corresponding to 285 seconds. The results of this simulation are stored in the *shaken reactor raw data matrix*. The numerical reactor model is excited once more, this time with harmonic functions of fairly long periods

[7]Such a mask will not capture the highest frequencies in the model, but as long as they are irrelevant to the task at hand of identifying the two proposed transients, this may be acceptable.

(same amplitudes), along another 2500 sampling points. This way of operating the reactor is called “normal reactor operation with harmonic perturbations.” In this way, a more realistic but still dynamic reactor simulation results. This step is used to determine the limits that the variables can realistically assume. The results of this simulation are stored in the *normal reactor raw data matrix*.

The data extracted from the numerical Fortran simulations constitute the measurement data (the raw data matrices) of the qualitative models. As has been explained before, the raw data matrices are real–valued matrices in which each column represents one recorded variable, whereas each row represents a complete data record collected at one time instant. These data must then be recoded to enable the qualitative reasoning process.

#### 7.5.1.2 Variable Selection and Hierarchization

To build the qualitative models of the reactor, not all of the variables that were considered in the numerical simulation can be used, but only some subsets of variables that capture the main characteristics of the reactor during all facets of operation.

There are two main differences between the variable selection processes carried out in Chapter 4 for the aircraft and circuit examples, and the one that will be carried out here for the nuclear rector example. The first difference has to do with how many and which ones are the important variables that must be taken into account, and the second is related to the utilization of the same set of selected variables to represent the abnormal conditions. Whereas in all the examples presented in this thesis so far, the set of important variables were always known in advance, their number was kept inside the limits imposed by the FIR methodology, and this set of selected variables was kept constant during all abnormal conditions, in the example presented here, none of these three conditions applies, since the number of variables is far beyond the current software and even the methodological limits, there is not a predetermined selection of variables, and the set of variables that represent the system will vary from transient to transient.

Although in Chapters 5 and 6 a solution to these problems has been proposed and demonstrated, respectively, by means of Reconstruction Analysis, it is only in the present chapter that this methodology is extensively used, and where the hierarchies of identified qualitative subsystems, each one represented by an optimal mask, are shown. To this end let us remember how these arrays are constructed:

- Clusters of causally related variables will be identified.
- Each cluster constitutes an identified subsystem, and should be composed of one or several inputs and a single output.
- From each cluster, it should be possible to obtain an optimal mask. Thus, the number of subsystem variables should be kept inside the FIR methodology and software limits.
- Clusters should be causally related among each other, i.e., form a causal temporal hierarchy.
- At the uppermost level of the hierarchy, there is one or more executive subsystems, the input variables of which are outputs generated by subsystems located at a lower hierarchical level.

In Section 7.4, it was already explained that the numerical model is composed of 79 state variables, 79 initial values for those state variables, 373 defined variables, and 216 parameters. It does not make any sense to preform the analysis over these 747 variables since, among them there are constants, arrays, and lots of variables that the operators cannot observe or influence from the control room. This latter condition is specially relevant for the licensing and transient validation simulation programs [NSAC–EPRI, 1980], in which all output variables must coincide with instrumentation in the control room, i.e., they must be variables that the operators can directly monitor and control.

Thus for the selection process, the parameters, the initial condition values, and the array variables will be neglected. From the remaining 55 state variables and 135 defined variables, all of those that the operators cannot monitor and influence from the control room will be neglected as well. At the end, there remain 16 state variables and 35 defined variables, i.e., a total of 51 variables, a number that is much smaller than the original number of 747 variables, but that, from the point of view of the FIR methodology, is still far from being manageable.

Among these 51 variables, there are power variables, water levels, reactivities, pressures, temperatures, enthalpies, heat flows, water flows, steam flows, two phase flows, and some induced voltages. The entire list of variables is given below:

General purpose variables:

| | | |
|---|---|---|
| $Pow$ | = | Reactor thermal output power (% $age$). |
| $Level$ | = | Water level ($meters$). |

Reactivity related variables ($Dollars$):

$$
\begin{aligned}
Reac &= \text{Core total reactivity.}\\
Voids &= \text{Voids reactivity.}\\
Dopler &= \text{Doppler reactivity.}\\
Scram &= \text{Emergency stop reactivity.}
\end{aligned}
$$

Pressure related variables ($Pascals$):

$$
\begin{aligned}
P_{DM} &= \text{Dome pressure.}\\
P_{LP} &= \text{Lower plenum pressure.}\\
P_{C} &= \text{Core pressure.}\\
P_{BYP} &= \text{Bypass channel pressure.}\\
P_{TUR} &= \text{Turbine pressure.}\\
\Delta P_{C} &= \text{Core pressure drop.}\\
\Delta P_{SEP} &= \text{Steam separators pressure drop.}\\
P_{TUR_{ref}} &= \text{Turbine reference pressure.}\\
\Delta P_{REC_1} &= \text{Recirculation loop 1 pressure drop.}\\
\Delta P_{REC_2} &= \text{Recirculation loop 2 pressure drop.}\\
\Delta P_{PP_1} &= \text{Recirculation pump 1 pressure drop.}\\
\Delta P_{PP_2} &= \text{Recirculation pump 2 pressure drop.}
\end{aligned}
$$

Temperature related variables (degrees Kelvin):

$$
\begin{aligned}
T_{FUEL} &= \text{Fuel temperature.}\\
T_{MOD} &= \text{Moderator temperature.}\\
T_{MIX} &= \text{Mixed water temperature.}\\
T_{SAT} &= \text{Saturation temperature.}
\end{aligned}
$$

Enthalpy related variables ($Joules/kg$):

$$
\begin{aligned}
H_{FW} &= \text{Feedwater enthalpy.}\\
H_{MIX} &= \text{Mixed water enthalpy.}\\
H_{F_{UP}} &= \text{Upper plenum water enthalpy.}\\
H_{FG_{DM}} &= \text{Dome steam and saturated water enthalpy.}\\
H_{L_{LP}} &= \text{Lower plenum liquid enthalpy.}\\
H_{REC_1} &= \text{Recirculation loop 1 water enthalpy.}\\
H_{REC_2} &= \text{Recirculation loop 2 water enthalpy.}
\end{aligned}
$$

Heat related variables ($Joules$):

$$
Q_{RES} = \text{Residual heat.}
$$

Water, steam, or two phase flows ($kg/sec$):

$$\begin{array}{lcl}
W_{G_{DM}} & = & \text{Dome steam flow.} \\
W_{FW} & = & \text{Feedwater flow.} \\
W_{LP} & = & \text{Lower plenum flow.} \\
W_{BYP} & = & \text{Bypass channel flow.} \\
W_{SRV} & = & \text{Safety and relief valves flow.} \\
W_{GF_{BW}} & = & \text{Bulkwater steam and water flow.} \\
W_{G_{ref}} & = & \text{Steam reference flow.} \\
W_{F_{SEP}} & = & \text{Steam separators water flow.} \\
W_{G_{SEP}} & = & \text{Steam separators steam flow.} \\
W_{FW_{ref}} & = & \text{Feedwater reference flow.} \\
W_{G_{UP}} & = & \text{Upper plenum steam flow.} \\
W_{G_C} & = & \text{Core steam flow.} \\
W_{T_C} & = & \text{Core total flow.} \\
W_{REC_1} & = & \text{Recirculation loop 1 total flow.} \\
W_{REC_2} & = & \text{Recirculation loop 2 total flow.} \\
W_{SUC_1} & = & \text{Jet pump loop 1 suction flow.} \\
W_{SUC_2} & = & \text{Jet pump loop 2 suction flow.} \\
W_{PP_1} & = & \text{Recirculation loop 1 pumped flow.} \\
W_{PP_2} & = & \text{Recirculation loop 2 pumped flow.}
\end{array}$$

Voltage related variables ($Volts$):

$$\begin{array}{lcl}
Volt_1 & = & \text{Recirculation pump 1 voltage.} \\
Volt_2 & = & \text{Recirculation pump 2 voltage.}
\end{array}$$

From these 51 selected variables extracted from the *shaken reactor raw data matrix*, subsets of variables will be identified by applying Reconstruction Analysis. To this end, the variables must first be recoded using the *fuzzy recoding* process explained in Section 3.3.3. All variables will be recoded into five qualitative classes.

For the full–power steady–state case, two variables have been chosen as the ultimate hierarchy outputs. They are: *Pow* and *Level*, and will be considered as the executive subsystems. The reason for such a selection is that those variables are two of the most characteristic ones for representing operational transients. At the same time, these two variables are usually some of the first ones the operator looks at when something begins to go wrong.

Since these two variables will be the top level outputs of the Fault Monitoring System, all others will be considered as input variables (49 variables). To find out the strengths of the binary relationships between those inputs and the two selected outputs, Reconstruction Analysis is being applied, as described in Chapter 6, to each of the above subsystems. For those binary relationships that appear more than once, their corresponding reconstruction error will be averaged.

However, there is an important difference in the role that RA has played in Chapter 6, and the role that it will play here. In Chapter 6, RA was used to identify subsets of relevant variables, directly finding high–quality masks for them, whereas in this chapter, RA will be limited to identify subsets of relevant variables. This limitation has to do with the large number of variables that makes the introduction of time in the analysis unpractical. For this reason, the columns of the raw data matrix will not be duplicated shifted up by one row.

Since *time* must be introduced somehow, Optimal Mask Analysis will be applied in a subsequent step to the identified subsets of variables, in order to obtain optimal masks for them.

The goal is the reduction of the number of possible inputs while maintaining the representativeness of the system. In this way, the FIR–based FMS will have a minimum set of meaningful variables to reason with. However, since 51 variables are still far too many to be managed by RA directly, RA will be performed in parts, as proposed in the two examples of Chapter 6.

The overall system of 51 variables will be decomposed into several subsystems. Each of them must contain some input variables and the desired output, in such a way that all possible binary relations are included at least once in the set of subsystems. Since the two selected output variables are *Pow* and *Level*, one decomposition will be made for each of them (excluding the other one). Thus, for the first output variable, *Pow*, the input variables were grouped in the following way:

$$
\begin{array}{llllllllll}
( & u_1, & u_2, & u_3, & u_4, & u_5, & u_6, & u_7, & Pow & ); \\
( & u_8, & u_9, & u_{10}, & u_{11}, & u_{12}, & u_{13}, & u_{14}, & Pow & ); \\
( & u_{15}, & u_{16}, & u_{17}, & u_{18}, & u_{19}, & u_{20}, & u_{21}, & Pow & ); \\
( & u_{22}, & u_{23}, & u_{24}, & u_{25}, & u_{26}, & u_{27}, & u_{28}, & Pow & ); \\
( & u_{29}, & u_{30}, & u_{31}, & u_{32}, & u_{33}, & u_{34}, & u_{35}, & Pow & ); \\
( & u_{36}, & u_{37}, & u_{38}, & u_{39}, & u_{40}, & u_{41}, & u_{42}, & Pow & ); \\
( & u_{43}, & u_{44}, & u_{45}, & u_{46}, & u_{47}, & u_{48}, & u_{49}, & Pow & ); \\
( & u_1, & u_8, & u_{15}, & u_{22}, & u_{29}, & u_{36}, & u_{43}, & Pow & ); \\
( & u_2, & u_9, & u_{16}, & u_{23}, & u_{30}, & u_{37}, & u_{44}, & Pow & ); \\
( & u_3, & u_{10}, & u_{17}, & u_{24}, & u_{31}, & u_{38}, & u_{45}, & Pow & ); \\
( & u_4, & u_{11}, & u_{18}, & u_{25}, & u_{32}, & u_{39}, & u_{46}, & Pow & ); \\
( & u_5, & u_{12}, & u_{19}, & u_{26}, & u_{33}, & u_{40}, & u_{47}, & Pow & );
\end{array}
$$

$$
\begin{array}{lllllllll}
( & u_6, & u_{13}, & u_{20}, & u_{27}, & u_{34}, & u_{41}, & u_{48}, & Pow \ ); \\
( & u_7, & u_{14}, & u_{21}, & u_{28}, & u_{35}, & u_{42}, & u_{49}, & Pow \ ); \\
( & u_1, & u_9, & u_{17}, & u_{25}, & u_{33}, & u_{41}, & u_{49}, & Pow \ ); \\
( & u_2, & u_{10}, & u_{18}, & u_{26}, & u_{34}, & u_{42}, & u_{43}, & Pow \ ); \\
( & u_3, & u_{11}, & u_{19}, & u_{27}, & u_{35}, & u_{36}, & u_{44}, & Pow \ ); \\
( & u_4, & u_{12}, & u_{20}, & u_{28}, & u_{29}, & u_{37}, & u_{45}, & Pow \ ); \\
( & u_5, & u_{13}, & u_{21}, & u_{22}, & u_{30}, & u_{38}, & u_{46}, & Pow \ ); \\
( & u_6, & u_{14}, & u_{15}, & u_{23}, & u_{31}, & u_{39}, & u_{47}, & Pow \ ); \\
( & u_7, & u_8, & u_{16}, & u_{24}, & u_{32}, & u_{40}, & u_{48}, & Pow \ ); \\
( & u_7, & u_{13}, & u_{19}, & u_{25}, & u_{31}, & u_{37}, & u_{43}, & Pow \ ); \\
( & u_6, & u_{12}, & u_{18}, & u_{24}, & u_{30}, & u_{36}, & u_{49}, & Pow \ ); \\
( & u_5, & u_{11}, & u_{17}, & u_{23}, & u_{29}, & u_{42}, & u_{48}, & Pow \ ); \\
( & u_4, & u_{10}, & u_{16}, & u_{22}, & u_{35}, & u_{41}, & u_{47}, & Pow \ ); \\
( & u_3, & u_9, & u_{15}, & u_{28}, & u_{34}, & u_{40}, & u_{46}, & Pow \ ); \\
( & u_2, & u_8, & u_{21}, & u_{27}, & u_{33}, & u_{39}, & u_{45}, & Pow \ ); \\
( & u_1, & u_{14}, & u_{20}, & u_{26}, & u_{32}, & u_{38}, & u_{44}, & Pow \ ); \\
( & u_1, & u_{10}, & u_{19}, & u_{28}, & u_{30}, & u_{39}, & u_{48}, & Pow \ ); \\
( & u_2, & u_{11}, & u_{20}, & u_{22}, & u_{31}, & u_{40}, & u_{49}, & Pow \ ); \\
( & u_3, & u_{12}, & u_{21}, & u_{23}, & u_{32}, & u_{41}, & u_{43}, & Pow \ ); \\
( & u_4, & u_{13}, & u_{15}, & u_{24}, & u_{33}, & u_{42}, & u_{44}, & Pow \ ); \\
( & u_5, & u_{14}, & u_{16}, & u_{25}, & u_{34}, & u_{36}, & u_{45}, & Pow \ ); \\
( & u_6, & u_8, & u_{17}, & u_{26}, & u_{35}, & u_{37}, & u_{46}, & Pow \ ); \\
( & u_7, & u_9, & u_{18}, & u_{27}, & u_{29}, & u_{38}, & u_{47}, & Pow \ ); \\
( & u_1, & u_{11}, & u_{21}, & u_{24}, & u_{34}, & u_{37}, & u_{47}, & Pow \ ); \\
( & u_2, & u_{12}, & u_{15}, & u_{25}, & u_{35}, & u_{38}, & u_{48}, & Pow \ ); \\
( & u_3, & u_{13}, & u_{16}, & u_{26}, & u_{29}, & u_{39}, & u_{49}, & Pow \ ); \\
( & u_4, & u_{14}, & u_{17}, & u_{27}, & u_{30}, & u_{40}, & u_{43}, & Pow \ ); \\
( & u_5, & u_8, & u_{18}, & u_{28}, & u_{31}, & u_{41}, & u_{44}, & Pow \ ); \\
( & u_6, & u_9, & u_{19}, & u_{22}, & u_{32}, & u_{42}, & u_{45}, & Pow \ ); \\
( & u_7, & u_{10}, & u_{20}, & u_{23}, & u_{33}, & u_{36}, & u_{46}, & Pow \ ); \\
( & u_1, & u_{12}, & u_{16}, & u_{17}, & u_{31}, & u_{42}, & u_{46}, & Pow \ ); \\
( & u_2, & u_{13}, & u_{17}, & u_{28}, & u_{32}, & u_{36}, & u_{47}, & Pow \ ); \\
( & u_3, & u_{14}, & u_{18}, & u_{22}, & u_{33}, & u_{37}, & u_{48}, & Pow \ ); \\
( & u_4, & u_8, & u_{19}, & u_{23}, & u_{34}, & u_{38}, & u_{49}, & Pow \ ); \\
( & u_5, & u_9, & u_{20}, & u_{24}, & u_{35}, & u_{39}, & u_{43}, & Pow \ ); \\
( & u_6, & u_{10}, & u_{21}, & u_{25}, & u_{29}, & u_{40}, & u_{44}, & Pow \ ); \\
( & u_7, & u_{11}, & u_{15}, & u_{26}, & u_{30}, & u_{41}, & u_{45}, & Pow \ ); \\
( & u_1, & u_{13}, & u_{18}, & u_{23}, & u_{35}, & u_{40}, & u_{45}, & Pow \ ); \\
( & u_2, & u_{14}, & u_{19}, & u_{24}, & u_{29}, & u_{41}, & u_{46}, & Pow \ ); \\
( & u_3, & u_8, & u_{20}, & u_{25}, & u_{30}, & u_{42}, & u_{47}, & Pow \ ); \\
( & u_4, & u_9, & u_{21}, & u_{26}, & u_{31}, & u_{36}, & u_{48}, & Pow \ );
\end{array}
$$

$$( \; u_5, \; u_{10}, \; u_{15}, \; u_{27}, \; u_{32}, \; u_{37}, \; u_{49}, \; Pow \; );$$
$$( \; u_6, \; u_{11}, \; u_{16}, \; u_{28}, \; u_{33}, \; u_{38}, \; u_{43}, \; Pow \; );$$
$$( \; u_7, \; u_{12}, \; u_{17}, \; u_{22}, \; u_{34}, \; u_{39}, \; u_{44}, \; Pow \; );$$

For the case of the output variable, *Level*, the subsystems are exactly the same with *Level* instead of *Pow*. However, since different qualitative models should be obtained for each one of the three analyzed cases, i.e., *full–power steady state*, *recirculation pump trip*, and *feedwater controller failure*,[8] a Reconstruction Analysis must be performed among all possible input variables for each output variable of each case. That means that each output variable will be considered independently of the others.

The procedure for each case is governed by the following rules:

1. The Single Refinement algorithm of RA is applied to each output subsystem (excluding all other output subsystems) determining the input variables it depends on.

2. If the input variables that the system depends on are more than five,[9] the variables with the strongest binary relationships (the limit value is set by the modeler) with the selected output are considered as intermediate subsystem of the next hierarchical level.

3. RA is applied to each intermediate subsystem determining the input variables it depends on.

4. If there remain more than five input variables, return to step 3, otherwise proceed to step 5.

5. Once the analysis has been applied to all intermediate subsystems of all levels, the variables that were not considered as intermediate subsystems are now considered as true input variables.

Let us analyze the three cases one by one. For the *full–power steady–state* case, RA is applied to the first output variable, *Pow*, excluding the other output variable, *Level*, as potential input. After a first refinement, the dependencies of *Pow* can be seen in Figures 7.7 and 7.8.

[8]The *feedwater controller failure* operational transient includes one more operation regime that also needs to be characterized, i.e., the emergency shutdown (Scram).

[9]Five input and one output variables constitute a reasonable limit for a mask when working with Optimal Mask Analysis.

| Input Variables | Optimal Structure Analysis Relation with Output *Pow* |
|---|---|
| $Reac$ | − − − − − − − − − − − − − − − ··· |
| $Voids$ | − − − − − − + − − − − − − − − ··· |
| $Dopler$ | − − − − − − − − − − − − − − − ··· |
| $Scram$ | − − − − − − − − − − − − − − − ··· |
| $P_{DM}$ | − + − − − − − − − − − − − − − ··· |
| $P_{LP}$ | − − − − − − + − − − − − − − − ··· |
| $P_C$ | − − − − − − − − − − − − − − + ··· |
| $P_{BYP}$ | − − − − − − − − − − − − − − − ··· |
| $P_{TUR}$ | − − − + − − − − − − − − − − − ··· |
| $\Delta P_C$ | − − − − − − − − − − − − − + − ··· |
| $\Delta P_{SEP}$ | − − − − − − − − − + − − − − − ··· |
| $P_{TUR_{ref}}$ | − − − − − + − − − − − − − − − ··· |
| $\Delta P_{REC_1}$ | − − − − − − − − − − − − − − − ··· |
| $\Delta P_{REC_2}$ | − − − − − − − − − − − − − − − ··· |
| $\Delta P_{PP_1}$ | − − − − − − − − − − − − − − − ··· |
| $\Delta P_{PP_2}$ | − − − − − − − − − − − − − − − ··· |
| $T_{FUEL}$ | − − − − − − − − − − − − − − − ··· |
| $T_{MOD}$ | − − − − − − − − − − − − − − − ··· |
| $T_{MIX}$ | − − − − − − − − − − − − − − + ··· |
| $T_{SAT}$ | − − − − − − − − − − − − − − − ··· |
| $H_{FW}$ | − − − − − − − + − − − − − − − ··· |
| $H_{MIX}$ | − − − − − − − − − − − − − − + ··· |
| $H_{F_{UP}}$ | − − − − − − − − − − − − − − − ··· |
| $H_{FG_{DM}}$ | − − − − − − − − + − − − − − − ··· |
| $H_{L_{LP}}$ | − − − − − − − − − − − − − − − ··· |
| $H_{REC_1}$ | − − − − − − − − − − − − − − − ··· |
| $H_{REC_2}$ | − − − − − − − − − − − − − − − ··· |
| $Q_{RES}$ | − − − − − − − − − − − − − − − ··· |
| $W_{G_{DM}}$ | − − − − − − − − − − − − + − − ··· |
| $W_{FW}$ | − − − − − − + − − − − − − − − ··· |
| $W_{LP}$ | − − + − − − − − − − − − − − − ··· |
| $W_{BYP}$ | − − − − − − − − − − − − − − − ··· |
| $W_{SRV}$ | − − − − − − − − − − − − − − − ··· |
| $W_{GF_{BW}}$ | − − − − − − − − − − − − − − − ··· |
| $W_{G_{ref}}$ | + − − − − − − − − − − − − − − ··· |
| $W_{F_{SEP}}$ | − − − − − − − − − + − − − − − ··· |
| $W_{G_{SEP}}$ | − − − − − − − − − − − − − − + ··· |
| $W_{FW_{ref}}$ | − − − − − − − − − − − − − − − ··· |
| $W_{G_{UP}}$ | − − − − − − − − − − − − − − − ··· |

Figure 7.7: Relation between input variables and output *Pow* of the decomposed nuclear reactor system.

| Input Variables | Optimal Structure Analysis Relation with Output *Pow* (Cont) |
|---|---|
| $W_{G_C}$ | – – – – – – – – – – – – – – – ··· |
| $W_{T_C}$ | – – – – – – – – – – – – – – – ··· |
| $W_{REC_1}$ | – – – – – – – – – – – – – – – ··· |
| $W_{REC_2}$ | – – – – – – – – – – – – – – – ··· |
| $W_{SUC_1}$ | – – – – – – – – – – – – – – – ··· |
| $W_{SUC_2}$ | – – – – – – – – – – – – – – – ··· |
| $W_{PP_1}$ | – – – – – – – – – – – – – – – ··· |
| $W_{PP_2}$ | – – – – – – – – – – – – – – – ··· |
| $Volt_1$ | – – – – – – – – – – – – – – – ··· |
| $Volt_2$ | – – – – – – – – – – – – – – – ··· |

Figure 7.8: Relation between input variables and output *Pow* of the decomposed nuclear reactor system (continued).

From the table of Figure 7.7, and taking into account those input variables that present a '+' mark between the first and the tenth column, It can be seen that the output variable *Pow* depends on 12 variables. They are:

$$\begin{aligned} Pow &= f\ (W_{G_{ref}}, P_{DM}, W_{LP}, P_{TUR}, P_{TUR_{ref}}, P_{LP}, W_{FW}, \\ &\quad Voids, H_{FW}, H_{FG_{DM}}, \Delta P_{SEP}, W_{F_{SEP}}) \end{aligned} \tag{7.3}$$

The most important variables, i.e., the ones with the strongest relationship with the output, have been chosen as outputs of intermediate subsystems. They are: $W_{G_{ref}}$, $P_{DM}$ and $W_{LP}$. From them, $W_{G_{ref}}$ is omitted since its only purpose was that of exciting the reactor, and in steady–state operation, it remains practically unchanged.

In order to determine a model for the intermediate output $P_{DM}$, RA is applied once more to the subsets of inputs consisting of the 12 selected variables except for *Pow* and the other intermediate output, $W_{LP}$. The resulting table is depicted in Figure 7.9.

From this table and taking into account only the first 6 columns, the input variables that $P_{DM}$ depends on are:

$$P_{DM} = f\ (P_{TUR}, P_{TUR_{ref}}, P_{LP}, Voids, \Delta P_{SEP}, H_{FW}, H_{FG_{DM}}) \tag{7.4}$$

| Input Variables | Optimal Structure Analysis Relation with Output $P_{DM}$ |
|---|---|
| $Voids$ | $- \; - \; - \; + \; - \; - \; - \; - \; -$ |
| $P_{LP}$ | $- \; - \; + \; - \; - \; - \; - \; - \; -$ |
| $P_{TUR}$ | $+ \; - \; - \; - \; - \; - \; - \; - \; -$ |
| $\Delta P_{SEP}$ | $- \; - \; - \; - \; + \; - \; - \; - \; -$ |
| $P_{TUR_{ref}}$ | $+ \; - \; - \; - \; - \; - \; - \; - \; -$ |
| $H_{FW}$ | $- \; - \; - \; - \; - \; + \; - \; - \; -$ |
| $H_{FG_{DM}}$ | $- \; - \; - \; - \; - \; + \; - \; - \; -$ |
| $W_{FW}$ | $- \; - \; - \; - \; - \; - \; - \; - \; +$ |
| $W_{F_{SEP}}$ | $- \; - \; - \; - \; - \; - \; - \; + \; -$ |

Figure 7.9: Relation between inputs and intermediate variable $P_{DM}$ of the decomposed nuclear reactor system.

It has been said that not more than five input variables are admitted in each FIR. Since there still remain more than five variables, Reconstruction Analysis is applied once again to the variables with the strongest relationships, in this case $P_{TUR}$. To this end, only the seven variables of Relation (7.4) will be considered excluding $P_{TUR}$ itself from the analysis. For $P_{TUR}$, we find the table depicted in Figure 7.10.

| Input Variables | Optimal Structure Analysis Relation with Output $P_{TUR}$ |
|---|---|
| $Voids$ | $- \; - \; - \; - \; - \; - \; - \; +$ |
| $P_{LP}$ | $- \; - \; - \; - \; - \; - \; + \; -$ |
| $\Delta P_{SEP}$ | $- \; + \; - \; - \; - \; - \; - \; -$ |
| $P_{TUR_{ref}}$ | $+ \; - \; - \; - \; - \; - \; - \; -$ |
| $H_{FW}$ | $- \; - \; - \; - \; - \; - \; - \; +$ |
| $H_{FG_{DM}}$ | $- \; - \; - \; - \; - \; + \; - \; -$ |

Figure 7.10: Relation between inputs and intermediate variable $P_{TUR}$ of the decomposed nuclear reactor system.

From this table and taking into account only the first two columns, the input variables that $P_{TUR}$ depends on are:

$$P_{TUR} \;=\; f\;(P_{TUR_{ref}}, \Delta P_{SEP}) \tag{7.5}$$

which means that $P_{DM}$ has a lowest hierarchical subsystem called $P_{TUR}$.

The variables that $P_{TUR}$ depends on are now eliminated from the list of direct inputs to the intermediate subsystem, hence:

$$P_{DM} \;=\; f\;(P_{TUR}, P_{LP}, Voids, H_{FW}, H_{FG_{DM}}) \tag{7.6}$$

The other variable, besides $P_{DM}$, with a strong binary relation with $Pow$ is $W_{LP}$. For this subsystem, a Reconstruction Analysis must be performed considering only the 12 variables that $Pow$ depends on as shown in Relation (7.3), and excluding $Pow$ and $P_{DM}$. The results obtained from the analysis are shown in the table of Figure 7.11.

| Input Variables | Optimal Structure Analysis Relation with Output $W_{LP}$ |
|---|---|
| $Voids$ | − − − − − − + − − |
| $P_{LP}$ | − − + − − − − − − |
| $P_{TUR}$ | + − − − − − − − − |
| $\Delta P_{SEP}$ | − − − − − − + − − |
| $P_{TUR_{ref}}$ | − − − − − + − − − |
| $H_{FW}$ | − − − − − − − + − |
| $H_{FG_{DM}}$ | − − − − − − − − + |
| $W_{FW}$ | + − − − − − − − − |
| $W_{F_{SEP}}$ | − + − − − − − − − |

Figure 7.11: Relation between inputs and intermediate variable $W_{LP}$ of the decomposed nuclear reactor system.

From this table and taking into account only the first three columns, a relation for $W_{LP}$ can be constructed. It is:

$$W_{LP} \;=\; f\;(W_{FW}, W_{F_{sep}}, P_{LP}) \tag{7.7}$$

In this way, the output variable *Pow* is computed using three hierarchical levels of FMSs. *Pow* depends on 9 input variables whereby 8 are different.

For the case of the other executive FIR output variable, *Level*, a Reconstruction Analysis was performed excluding variable *Pow* only. From this analysis, 11 variables were considered important:[10]

$$\begin{aligned} Level &= f\ (W_{FW_{ref}}, W_{G_{DM}}, W_{LP}, \Delta P_C, P_{LP}, Voids, \\ &\quad W_{FW}, W_{GF_{BW}}, P_C, W_{F_{SEP}}, \Delta P_{SEP}) \end{aligned} \tag{7.8}$$

The variables with the strongest binary relationship with *Level* are: $W_{FW_{ref}}$, $W_{G_{DM}}$, $\Delta P_C$, and $W_{LP}$. They will be considered as subsystems. The first of them, $W_{FW_{ref}}$, is neglected since its only purpose was that of exciting the reactor model, and during steady state it remains practically unchanged. For each of the other three variables, Reconstruction Analysis must be applied once more. The results of these operations are the following. The first of the three subsystems, $W_{G_{DM}}$, depends on 3 variables:

$$W_{G_{DM}} = f\ (Voids, W_{GF_{BW}}, W_{FW}) \tag{7.9}$$

The second subsystem, $\Delta P_C$, depends also on three variables:

$$\Delta P_C = f\ (P_{LP}, P_C, \Delta P_{SEP}) \tag{7.10}$$

The third subsystem, $W_{LP}$, depends on three variables as well. They are:

$$W_{LP} = f\ (W_{FW}, W_{F_{SEP}}, P_{LP}) \tag{7.11}$$

The output variable *Level* depends on 9 input variables (7 are different) and has two hierarchical levels, in contrast to the output variable *Pow* that depends on 9 input variables (8 different) and has three hierarchical levels.

[10] Due to space limitations, the resulting tables of each analysis will be omitted from now on, and only the derived relationships will be presented.

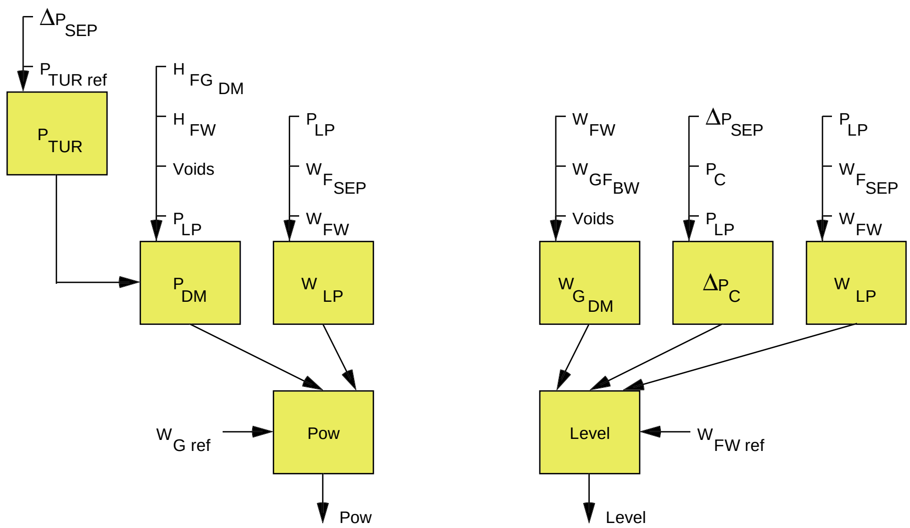


Figure 7.12: Hierarchy of subsystems and input variables for the steady state case.

Notice that the number of variables that the system depends on has been reduced from 51 to 18. However, since several variables are inputs to more than one subsystem, and even one entire subsystem is common to both output variables, the total reduction is from 51 to 10. This means that the system can be qualitatively represented with less than one fifth of the original number of variables.

Figure 7.12 represents the hierarchical arrangement of subsystems and input variables for the *full–power steady–state* case. The two excitation variables, $W_{G_{ref}}$ and $W_{FW_{ref}}$, are symbolically shown as inputs to the two executive subsystems. However, they cannot be considered true inputs, since they are constant during the considered steady states. Notice that some of the subsystems do not coincide with true physical subsystems of the nuclear reactor model, whereas others do. This is the case of subsystems $P_{TUR}$, $W_{LP}$, and $\Delta P_C$. In particular, the pressure drop in the core, $\Delta P_C$, is a derivative equation that includes in its calculation the previous, the actual, and the next pressure knots, i.e., the lower plenum pressure, $P_{LP}$, the core pressure, $P_C$, and the pressure drop of the steam separators, $\Delta P_{SEP}$, respectively.

#### 7.5.1.3 Propagation of Errors

The question may be raised why the signals of the lower–level subsystems are used as inputs to the higher level ones, if the true signals are available as measurements, as was explained earlier.

To answer this question, it should be remembered that the fault monitoring scheme presented in this thesis is based on the errors obtained by the continuous comparison between the predicted behavior of a system, and its true behavior. Thus, the reason has to do with the propagation of errors through the hierarchical ladder.

A lower– or intermediate–level subsystem detects an anomalous behavior when its error matrix contains a certain combination of saturated states and a certain number of consecutive prediction errors. The output of this lower–level subsystem contains the erroneous behavior. If this output serves as input to another intermediate subsystem or a high–level subsystem, the error condition is propagated to the upper levels of the hierarchical ladder, making the detection process of the subsystems at those levels faster and easier. In contrast, if the true signals (available as measurements at any time) were used as inputs to the higher–levels subsystems, the error condition would not be propagated, making the detection process of those subsystems longer and more difficult.

It must be considered that for the three cases presented in this chapter, i.e., *full–power steady state*, *recirculation pump trip*, and *feedwater controller failure*, only the intermediate– and higher–level subsystems make a continuous comparison between their predicted behavior and the true system behavior. The lower–level subsystems are included as a further system decomposition needed to keep the number of variables of an intermediate–level subsystem inside FIR limits, i.e., they are used to support the qualitative modeling of the variables at the intermediate level, and not to reason about faults. The output of the $P_{TUR}$ subsystem is a qualitative representation of the $P_{TUR}$ variable, whereas the outputs of the $P_{DM}$, $W_{LP}$, $W_{G_{DM}}$, and $\Delta P_C$ subsystems are local alarm signals.

#### 7.5.1.4 Optimal Masks

Every identified subsystem needs to be represented by means of an optimal mask, i.e., all of them must be qualitatively modeled by means of Optimal Mask Analysis, and in fact, the hierarchy of subsystems is a hierarchy of qualitative models generated and simulated by FIR. From the past section, we know the mask depth and the number of qualitative classes that each variable can assume. At this point two approaches are possible. On the one hand, a) all masks can be supposed to have the same depth and sampling rate, and a different number of variables, and on the other hand, b) the masks can be allowed to assume different parameters among each other. With the latter option, it can be expected that more accurate masks might be obtained at the cost of making the qualitative modeling process slower, since a new stability analysis would have to be performed for each identified subsystem to determine the most appropriate sampling interval and mask depth. With the former option, the qualitative modeling process is simplified at the cost of a somewhat reduced mask accuracy.

Since the FMS that will be used to monitor the nuclear reactor must be able to operate in a continuous way, the Forecasting All Possible Structures fault monitoring strategy[11] will be applied, propagating information up and down the hierarchical ladder. It therefore seems reasonable that all masks should at least cover the same time intervals, i.e., they all should have the same depth and sampling rate. Thus for simplicity reasons it has been decided to choose option (a), applying to all masks the same parameters. These parameters are:

| | | |
|---|---|---|
| sampling rate | = | 0.114 sec; |
| number of qualitative levels | = | 5; |
| depth of the masks | = | 5; |
| maximum mask complexity | = | 7; |
| number of variables | = | variable. |

The mask candidate matrices were built following Equation (3.15). Applying Optimal Mask Analysis to each identified subsystem, the resulting optimal masks for the output variable $Pow$ of Figure 7.12 are, for the lowest hierarchical level:

$$P_{TUR} \; = \; \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} P_{TUR_{ref}} & \Delta P_{SEP} & P_{TUR} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -2 & 0 \\ -3 & -4 & -5 \\ -6 & 0 & +1 \end{pmatrix} \end{array}$$

for the intermediate hierarchical level:

[11] c.f. Section 4.2.5.3

$$P_{DM} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccccc} P_{TUR} & P_{LP} & Voids & H_{FW} & H_{FG_{DM}} & P_{DM} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 0 \\ -2 & -3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 & -5 \\ 0 & 0 & 0 & -6 & 0 & +1 \end{pmatrix} \end{array}$$

$$W_{LP} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} W_{FW} & W_{F_{SEP}} & P_{LP} & W_{LP} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & -2 & 0 \\ -3 & -4 & 0 & -5 \\ 0 & 0 & -6 & +1 \end{pmatrix} \end{array}$$

and for the top–level (executive) $Pow$ variable itself:

$$Pow \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} P_{DM} & W_{LP} & Pow \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -2 & 0 \\ -3 & 0 & -4 \\ -5 & 0 & +1 \end{pmatrix} \end{array}$$

Notice that, at this point, another input variable has been eliminated, since $P_{DM}$ no longer depends on $H_{FG_{DM}}$.

The second subhierarchy of Figure 7.12, that of the executive output variable *Level* has two levels only. For the lower level, the optimal masks are:

$$W_{G_{DM}} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} Voids & W_{GF_{BW}} & W_{FW} & W_{G_{DM}} \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -2 & -3 & 0 & -4 \\ 0 & 0 & 0 & 0 \\ 0 & -5 & 0 & +1 \end{pmatrix} \end{array}$$

$$\Delta P_C \quad = \quad \begin{array}{c} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} P_{LP} & P_C & \Delta P_{SEP} & \Delta P_C \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -2 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 \\ 0 & -4 & -5 & +1 \end{pmatrix} \end{array}$$

$$W_{LP} \quad = \quad \begin{array}{c} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} W_{FW} & W_{F_{SEP}} & P_{LP} & W_{LP} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & -2 & 0 \\ -3 & -4 & 0 & -5 \\ 0 & 0 & -6 & +1 \end{pmatrix} \end{array}$$

and for the executive *Level* variable itself:

$$Level \quad = \quad \begin{array}{c} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} W_{G_{DM}} & \Delta P_C & W_{LP} & Level \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -2 \\ 0 & -3 & 0 & -4 \\ -5 & 0 & -6 & +1 \end{pmatrix} \end{array}$$

### 7.5.2 Recirculation Pump Trip

As can be seen in Figure 7.4, the BWR has two recirculation loops each one with one recirculation pump. A recirculation pump trip can be produced by an operator error, a failure in the pump power supply, or by the Reactor Protection System when one or several of the following conditions are reached inside the reactor: low water level, high pressure, or low feedwater flow. The primary effects that a recirculation pump trip produces are: a decrease in the water flow (refrigerant) that goes into the core of almost 50 %, and consequently, a 40 % reduction in the reactor's output thermal power. The secondary effects have to do with the reactor water level, the steam flow that leaves the reactor, the feedwater supply, and the core temperature.

During the transient initial phase, the reactor water level is increased due to the increment in the core voids produced by the reduction of refrigerant. The pressure and the water level controllers respond in an integrated way to the water level increment and the thermal power reduction by decreasing the feedwater flow and the steam flow leaving the reactor, trying to maintain the reactor pressure and the water level, while avoiding further emergency actions and smoothly driving the reactor to a new steady state. In the transient second phase, once the recirculation pump has stopped rotating, the flow through this loop is reversed, while the recirculation flow in the second loop (the one that remains operational) increases to 140 % due to the reactor pressure reduction, compensating in part for the loss of the first loop. At the final phase of the transient, a new steady state is reached at about 65 % of thermal output power[12] with a reduction in the core temperature, small oscillations in the water level around its normal position, a reduction in the feedwater flow, and a reduction in the reactor pressure.

The entire transient, from the pump trip to the new steady state, takes no longer than 25 seconds. In spite of being a well–known small operational transient that does not trigger any emergency action and that permits proceeding without shutting down the reactor, from an operator's perspective, sitting in front of the control panels, this transient will result in the triggering of three or four acoustic alarms (low pressure, high water level, low recirculation flow, low–low recirculation flow, etc.), more than thirty blinking emergency lights, and about fifteen indicators changing their values (without counting duplicated and triplicated indicators). This simply makes it impossible for him or her to discern the true causes of the transient from their consequences, and therefore, it is unrealistic to expect that the operator would be able to correctly characterize and identify the transient in real time, i.e., within a time interval, in which a real–time interaction might have been meaningful.

The transient characterization and identification can only be performed by the operators in a post–accident condition by consulting the Final Safety Analysis Report, the Transient Safety Analysis Report, the Emergency Operation Procedures, and by bringing to bear lots of previous expertise.

[12]The negative reactivity introduced by voids is compensated for by a positive reactivity introduced by the core temperature change (doppler). This null reactivity equilibrium value maintains the criticality of the reactor.

#### 7.5.2.1 Excitation

Although the time constants and consequently the sampling rate and the mask depth are the same for this transient, the excitation process was done in a completely different way. Since this is a sudden transient that cannot be inductively discovered by data analysis, i.e., cannot be prevented by a FMS of the type presented here, the reactor excitation is performed by provoking the same transient with variable intensity in the forward and reverse directions.

Let us explain how this is accomplished. The simulation is started at *full–power steady state*. Then suddenly, at time step 50, the induced voltage value of the recirculation pump number one ($Volt_1$) is set to zero. The transient takes about 25 seconds, until a new steady state is reached. At time step 320, the induced voltage value of the same recirculation pump is suddenly reset to its original value. This reverse transient takes about 15 seconds until the original steady state is reached. Then, at time step 475, the forward transient is repeated, and at time step 745, the reverse transient is repeated, and so on, in such a way that every 425 time steps a forward transient should start. The simulation was run along 3000 time steps, and the results were stored in the *zero flow raw data matrix*.

This procedure was repeated four more times, each one with a different value for the reduction of the induced voltage $Volt_1$. Since the relationship between voltage and pumped flow is non–linear, the values at which this variable will be reduced in each simulation correspond to 80 %, 60 %, 40 %, and 20 % of the pumped flow, respectively. The results of these simulations were stored in four different matrices known as the *80 % flow raw data matrix*, the *60 % flow raw data matrix*, etc. The idea behind this procedure is that of obtaining data of intermediate states of the transient trying to have the richest possible information about the reaction of the system.

Finally, the reactor is excited once more around the steady state reached by the first simulation run, i.e., the one in which the voltage value was suddenly reduced to zero, in exactly the same manner shown for the first excitation of the *full–power steady state* (same variables, same periods, and same amplitudes) along 400 time steps. In this way, the new steady state is also characterized, which permits the FMS to remain operational during and beyond the accident.

All these matrices are finally combined into a single matrix that contains the information of all simulations in such a way that, once recoded, only the occurrences of those states that were not present in the zero flow raw data matrix were included. This matrix is known as the *combined flow raw data matrix* and constitutes the measurement data of the qualitative model.

#### 7.5.2.2 Variable Selection and Hierarchization

In the case of this first transient, the process of variable selection and causal grouping of variables was done in exactly the same way as shown for the steady–state case. The same 51 variables were considered for the analysis, but this time extracted from the *combine flow raw data matrix*. Variables *Pow* and *Level* are once again chosen as the output variables of the executive subsystems, and Reconstruction Analysis is applied to each of these outputs separately.

Output *Pow* depends on the following 13 variables:

$$\begin{aligned} Pow \;&=\; f\ (P_{LP}, W_{REC_1}, Reac, Voids, Volt_1, \Delta P_{PP_1}, \\ &\quad \Delta P_C, W_{LP}, P_C, W_{PP_1}, W_{SUC_1}, P_{DM}, Dopler) \end{aligned} \tag{7.12}$$

The variables with the strongest binary relationships with the output *Pow* are: $P_{LP}$, $W_{REC_1}$, and *Reac*. They will be considered as intermediate subsystems, which means that Reconstruction Analysis should be applied to each one of them to obtain their dependencies.

Applying RA to subsystem $P_{LP}$, where only the 13 variables of Relation (7.12) are considered and variables *Pow*, $W_{REC_1}$, and *Reac* are excluded, 3 variables are considered important. They are:

$$P_{LP} \;=\; f\ (\Delta P_C, P_C, P_{DM}) \tag{7.13}$$

Applying RA to subsystem $W_{REC_1}$, excluding *Pow*, $P_{LP}$, and *Reac*, 4 variables are considered important. They are:

$$W_{REC_1} \;=\; f\ (Volt_1, \Delta P_{PP_1}, W_{PP_1}, W_{SUC_1}) \tag{7.14}$$

Finally, applying RA to subsystem *Reac* (excluding *Pow*, $P_{LP}$, and $W_{REC_1}$), 4 variables are considered important. They are:

$$Reac \;=\; f\ (Voids, W_{LP}, P_C, Dopler) \tag{7.15}$$

In this way, the executive output variable *Pow* is fed by three subsystems that depend on 10 variables.

The other executive subsystem output variable was chosen to be *Level*. RA should be applied to this variable in order to look for its dependencies. The only excluded variable in this analysis will be *Pow*. The result of this analysis shows that *Level* depends on 14 variables in the following way:

$$\begin{aligned} Level \;&=\; f\,(W_{REC_1}, W_{REC_2}, W_{G_{DM}}, Volt_1, \Delta P_{PP_1}, W_{PP_1}, \\ &\quad W_{FW}, W_{SUC_1}, \Delta P_{PP_2}, W_{PP_2}, W_{SUC_2}, Voids, P_C) \end{aligned} \tag{7.16}$$

From them, those with the strongest binary relationships with the output are chosen as intermediate subsystems. They are: $W_{REC_1}$, $W_{REC_2}$, and $W_{G_{DM}}$. RA should be applied once more to each one of these subsystems. In the case of subsystem $W_{REC_1}$, taking into account only the 14 variables that *Level* depends on and excluding *Level*, $W_{REC_2}$, and $W_{G_{DM}}$, the variables that the subsystem depends on are:

$$W_{REC_1} \;=\; f\,(Volt_1, \Delta P_{PP_1}, W_{PP_1}, W_{SUC_1}, P_C) \tag{7.17}$$

Applying RA to subsystem $W_{REC_2}$ (excluding *Level*, $W_{REC_1}$, and $W_{G_{DM}}$), 5 variables are considered important. They are:

$$W_{REC_2} \;=\; f\,(\Delta P_{PP_1}, W_{PP_2}, W_{SUC_2}, \Delta P_{PP_2}, P_C) \tag{7.18}$$

Applying RA to subsystem $W_{G_{DM}}$ (excluding *Level*, $W_{REC_1}$, and $W_{REC_2}$), 3 variables are considered important. They are:

$$W_{G_{DM}} \;=\; f\,(W_{FW}, P_{DM}, Voids) \tag{7.19}$$

Both output variables *Pow* and *Level* have two hierarchical levels. The first one depends on 11 input variables (10 different), whereas the second one depends on 13 input variables (11 different).

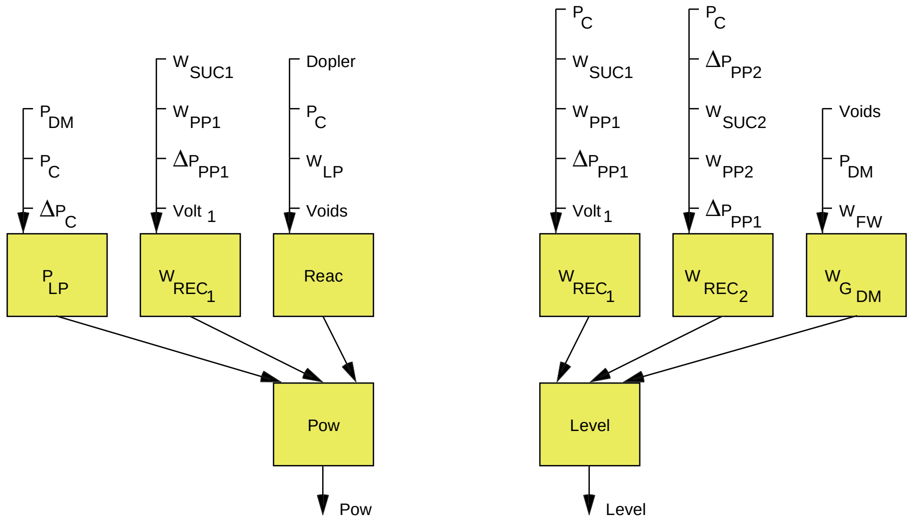


Figure 7.13: Hierarchy of subsystems and input variables for the recirculation pump trip case.

The number of variables that the system depends on has been reduced from 51 to 24. However, since several variables are inputs to more than one subsystem, and even one subsystem is common to both output variables, i.e., subsystem $W_{REC_1}$, the total reduction is from 51 to 15. This means that the system can be qualitatively represented with less than one third of the original number of variables.

Figure 7.13 represents the hierarchical arrangement of subsystems and input variables for the *recirculation pump trip* case. As in the previously analyzed *full–power steady–state* case, some subsystems coincide with true physical subsystems of the model. They are the flows in the recirculation loops number 1 and 2, $W_{REC_1}$ and $W_{REC_2}$, respectively. Their numerical equations depend on the pressure drop due to the pumps, $\Delta P_{PP_1}$ and $\Delta P_{PP_2}$, the suctioned and pumped flows in each loop, $W_{SUC_1}$, $W_{SUC_2}$, $W_{PP_1}$, and $W_{PP_2}$, and the voltage of each pump $Volt_1$ and $Volt_2$.

Notice also that the subsystem $W_{REC_2}$ (recirculation loop 2 flow) includes the input variable $\Delta P_{PP_1}$ (recirculation pump 1 pressure drop). This happens because, in this transient, the tripped pump influences the remaining pump in such a way that, while the former shows reversed flow, the latter increases its pumping capacity trying to compensate for the lost flow.

#### 7.5.2.3 Optimal Masks

It has already been said in the preceding section that the mask parameters will be the same as those used in the steady state case. The two hierarchies shown in Figure 7.13 have only two levels. Thus, applying Optimal Mask Analysis to each of the identified subsystems, the resulting optimal masks are, for the lower level first subhierarchy:

$$P_{LP} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} \Delta P_C & P_C & P_{DM} & P_{LP} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & -3 & -4 & +1 \end{pmatrix} \end{array}$$

$$W_{REC1} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} Volt_1 & \Delta P_{PP_1} & W_{PP_1} & W_{SUC_1} & W_{REC_1} \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & -3 & -4 & -5 \\ 0 & -6 & 0 & 0 & +1 \end{pmatrix} \end{array}$$

$$Reac \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} Voids & W_{LP} & P_C & Dopler & Reac \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -2 & 0 & 0 \\ -3 & 0 & 0 & 0 & -4 \\ -5 & 0 & 0 & -6 & +1 \end{pmatrix} \end{array}$$

and for the upper level variable $Pow$ itself:

$$Pow \quad = \quad \begin{array}{c c} {}_t\backslash^x & \begin{array}{cccc} P_{LP} & W_{REC_1} & Reac & Pow \end{array} \\ \begin{array}{l} t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -2 & 0 & -3 & 0 \\ 0 & -4 & 0 & 0 \\ -5 & -6 & 0 & +1 \end{pmatrix} \end{array}$$

For the second subhierarchy lower level, they are:

$$W_{REC_1} \quad = \quad \begin{array}{c c} {}_t\backslash^x & \begin{array}{cccccc} Volt_1 & \Delta P_{PP_1} & W_{PP_1} & W_{SUC_1} & P_C & W_{REC_1} \end{array} \\ \begin{array}{l} t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} & \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & -4 & 0 & 0 \\ 0 & -5 & 0 & 0 & -6 & +1 \end{pmatrix} \end{array}$$

$$W_{REC_2} \quad = \quad \begin{array}{c c} {}_t\backslash^x & \begin{array}{cccccc} \Delta P_{PP_1} & W_{PP_2} & W_{SUC_2} & \Delta P_{PP_2} & P_C & W_{REC_2} \end{array} \\ \begin{array}{l} t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & -3 & -4 & -5 & 0 \\ 0 & 0 & 0 & -6 & 0 & +1 \end{pmatrix} \end{array}$$

$$W_{G_{DM}} \quad = \quad \begin{array}{c c} {}_t\backslash^x & \begin{array}{cccc} W_{FW} & P_{DM} & Voids & W_{G_{DM}} \end{array} \\ \begin{array}{l} t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -2 & -3 & 0 & 0 \\ -4 & 0 & 0 & +1 \end{pmatrix} \end{array}$$

and for the upper level executive variable *Level*:

$$
Level \;=\; \begin{array}{c} t\backslash^{x} \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array}
\begin{array}{c} \begin{array}{cccc} W_{REC_1} & W_{REC_2} & W_{G_{DM}} & Level \end{array} \\
\left(\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -2 & -3 & 0 & 0 \\ -4 & -5 & 0 & +1 \end{array}\right) \end{array}
$$

Notice that once again, two additional variables, $W_{LP}$ and $W_{PP_2}$, have been dropped in the process.

### 7.5.3 Feedwater Controller Failure

This transient begins when the feedwater controller suffers a failure that causes the feedwater pumps to inject 110 % of their nominal value into the reactor. The main difference between the previously analyzed recirculation pump trip and this transient is that, whereas the former produced a new reactor steady state without any emergency actions, the latter is progressive and concludes with a reactor emergency shutdown.

The feedwater control failure is produced by an error in its set point values, or by the control itself when interpreting the signal of the feedwater demand. The primary effects that this transient produces are: an increase in the core inlet water subcooling, and an increase in the reactor water level due to the increment in the feedwater flow, and also an increase in the reactor thermal output power due to a reduction in the core voids produced by the subcooled refrigerant. Secondary effects have to do with the core temperature, the reactor pressure, an emergency stop, a core power excursion, and a series of emergency actions in the steam line such as: turbine trip, opening of the steam bypass, and opening of the relief valves.

During the transient initial phase, the water level and the pressure controllers react in an integrated way by sending a reduction signal to the feedwater controller, trying to reduce pressure, water level, and thermal output power. However, since the feedwater controller is stuck at 110 % of its nominal value and no automatic actions are available that would counter this effect, all these variables will continue rising. At this point the operator could take some manual actions to overcome the problem, or at least to reduce its impact, if only he or she were able to know what is really going on. During the second phase, pressure, water level, core temperature, and thermal output power will continue rising. In the control room, the operator will probably be concentrating his or her attention on the main variable indicators (water level, core temperature, and thermal output power), while ignoring the acoustic alarms, the blinking lights, and the remaining indicators. The operator knows that somehow the reactor is getting filled with water, but runs out of time and has no way of knowing that the real cause is a failure in the feedwater controller. Thus, the operator will not take any action at all.

In the final phase of the transient, the reactor will be automatically shut down due to one or both of the following two conditions: thermal output power equivalent to 120 %, and reactor water level reaching the maximum tolerable level, depending on the error magnitude of the feedwater controller. In this case, with an error of 10 %, the latter condition is reached first. At this point, the Reactor Protection System takes the following emergency actions:

- Turbine trip. The inlet turbine valves are closed impeding the steam from being vented.
- Reactor emergency shutdown (Scram). The control rods are pushed into the core injecting a great amount of negative reactivity, thereby shutting down the reactor.
- Steam bypass opening. Since the steam cannot be vented through the turbine anymore, the reactor pressure undergoes a significant increase that forces the steam bypass valves to open at their maximum capacity.
- Relief valves opening. The steam bypass alone is not capable of venting the accumulated steam. Thus, to reduce the pressure, the steam relief valves open.

With these conditions, the reactor water level is first reduced due to the turbine trip, and then augmented due to the swelling produced by the relief valves sudden reactor depressurization. The reactor thermal output power suffers a continuous increase during the initial and second phases of the transient due to an increase in the core voids produced by the subcooled water. Then suddenly, in the final phase and due to the turbine trip, it suffers a tremendous increase of about 200 % (power excursion) that is mitigated by means of the emergency shutdown.

The entire transient takes about 60 seconds, the last 10 seconds of which correspond to the final phase, i.e., the stabilization time after the emergency shutdown. The transient analysis is normally carried out by the operators in a post–accident condition. Each acoustic alarm and each blinking emergency light must be individually recognized and reset to know if the alarm condition is over. At the end of this process, the operators will notice that the lights corresponding to the feedwater controller cannot be turned off, i.e., the condition that turned them on is not over, and will identify this controller as the transient triggering component. In this way, the operators finally conclude that the reactor suffered a feedwater controller failure, a known operational transient that allows a fast reactor restart, once the faulty component has been fixed or exchanged.

However, if during the first 30 to 40 seconds, the operators, or some FMS that reports its findings to them, were able to identify, characterize, and analyze the transient, it would be possible for the operators to avoid the emergency shutdown by taking manual actions directly on the feedwater control.

#### 7.5.3.1 Excitation

The excitation process of this transient is very similar to that of the recirculation pump trip, and consists of provoking the same transient with variable intensity in the forward and reverse directions. Once more, the time constants are considered to be the same that were used in the simulation of the *full–power steady state*. However, this is not a sudden transient, but a slowly increasing transient that can indeed be inductively discovered by data analysis, and consequently, that can be prevented by a FMS of the type presented here.

Things start out as in the previous case, i.e., at full–power steady state. At time step 50, the reference value of the feedwater flow control is set to, and stuck at 1193.95 $kg/sec$, representing a 10 % increase of the original steady state value of 1085.41 $kg/sec$, causing an increase in the feedwater flow that reaches 110 %. The transient takes about 50 seconds from the beginning to immediately before the emergency shutdown is initialized. Thus, at time step 485 (3 time steps before the emergency shutdown), the reference value of the feedwater flow controller is reset to its original value. This reverse transient takes another 50 seconds until the original steady state is reached. Then, at time step 952, the forward transient is repeated, and at time step 1390 the reverse transient is repeated, and so on, in such a way that every 902 time steps a forward transient should start. The simulation was run along 3000 time steps, and the results were stored in the *110 % failure raw data matrix*.

This procedure was repeated twice, each with a different value for the increment of the feedwater flow control reference $W_{FW_{ref}}$. The values are 12 % and 8 %, corresponding to feedwater flows of 1215.66 $kg/sec$ (112 %) and 1172.24 $kg/sec$ (108 %), respectively. The results of these simulations were stored in the *112 % failure raw data matrix* and in the *108 % failure raw data matrix*. The three raw data matrices were combined into one, in such a way that, once recoded, only the occurrences of those states that were not present in the 110 % failure raw data matrix were included. This matrix is known as the *combined failure raw data matrix*, and constitutes the measurement data of the qualitative model.

In the first case, the transient takes about 41 seconds, whereas in the second case it takes about 63 seconds.[13] In both cases, the reverse transient is initiated some time intervals before the Scram emergency shutdown. The Scram is avoided in all three simulations, because, once the shutdown has been initiated, no reverse transient can be attempted any longer.

The Scram is itself an operational transient, and must be separately modeled and simulated. To this end, three different manual Scram simulations were carried out (only in the forward direction, since it is impossible to do it in the reverse direction) along 500 sampling intervals, and starting at different power levels at the sampling interval 50. The first one started at 120 % power, which is the power limit at which the feedwater controller failure transient terminates. The second started at 110 % power, and the third at 100 % power. Each simulation was stored in a different raw data matrix. Then, the three matrices were combined into a single matrix as was done in the previous simulations.

#### 7.5.3.2 Variable Selection and Hierarchization

The same 51 variables used in the two previous cases will be considered here, but this time extracted from the *combined failure raw data matrix*. All these variables will once more be recoded into 5 qualitative levels in order to apply Reconstruction Analysis.

Let us keep variables *Pow* and *Level* as the output variables of the executive subsystems. Thus, as has been done for the *full–power steady state* and *recirculation pump trip* cases, RA should be applied to each of these two

[13]This is logical since more feedwater flow will produce a faster increase in the refrigerant subcooling, that in turn will cause more voids and more thermal output power, taking less time to reach the limit Scram condition. In contrast, a smaller increment in the feedwater flow will produce a transient with a larger duration.

variables separately. Applying RA to $Pow$, it is determined that it depends on 17 variables in the following way:

$$\begin{aligned} Pow \;\; = \;\; & f\ (P_{DM}, Reac, T_{FUEL}, W_{G_{DM}}, T_{MOD}, H_{FW}, Scram, \\ & H_{MIX}, Voids, P_{LP}, P_{TUR}, H_{FG_{DM}}, W_{BYP}, W_{SR}, \\ & P_C, Q_{RES}, P_{TUR_{ref}}) \end{aligned} \tag{7.20}$$

The variables with the strongest binary relationships with the output will be considered as intermediate subsystems. They are: $P_{DM}$, $Reac$, $T_{FUEL}$, and $W_{G_{DM}}$. To each of them RA should be applied to discover its dependencies.

Let us start with subsystem $P_{DM}$. In order to apply RA to this subsystem, only the 17 variables that $Pow$ depends on should be considered, and all other subsystems should be excluded, that is, $Pow$, $Reac$, $T_{FUEL}$, and $W_{G_{DM}}$. The result of this operation is that $P_{DM}$ depends on the following four variables:

$$P_{DM} \;\; = \;\; f\ (P_{TUR}, W_{BYP}, P_C, P_{TUR_{ref}}) \tag{7.21}$$

Performing the same analysis with subsystem $Reac$ (excluding $Pow$, $P_{DM}$, $T_{FUEL}$, and $W_{G_{DM}}$), 3 variables are considered important. They are:

$$Reac \;\; = \;\; f\ (Scram, H_{FW}, Voids) \tag{7.22}$$

Applying RA to subsystem $T_{FUEL}$ (excluding $Pow$, $P_{DM}$, $Reac$, and $W_{G_{DM}}$), 8 variables are considered important. They are:

$$\begin{aligned} T_{FUEL} \;\; = \;\; & f\ (T_{MOD}, H_{MIX}, H_{FW}, P_{LP}, Scram, H_{FG_{DM}}, \\ & Voids, Q_{RES}) \end{aligned} \tag{7.23}$$

Since subsystem $T_{FUEL}$ depends on more than five variables, those variables with the strongest binary relationships with $T_{FUEL}$ itself will be considered as intermediate subsystems. In this case, the only variable that shall be considered a subsystem, as a consequence of its strong binary relationship, is $T_{MOD}$.

Applying RA to the new subsystem $T_{MOD}$, where only the 8 variables that $T_{FUEL}$ depends on are considered, and $T_{FUEL}$ and $T_{MOD}$ itself are excluded, 3 variables are considered important. They are:

$$T_{MOD} \;=\; f\;(H_{MIX}, Scram, H_{FG_{DM}}) \tag{7.24}$$

This means that these three variables will be inputs to the third hierarchical level subsystem $T_{MOD}$, whereas the other four variables of Relation (7.23) will be inputs to the second hierarchical level subsystem $T_{FUEL}$:

$$T_{FUEL} \;=\; f\;(T_{MOD}, H_{FW}, P_{LP}, Voids, Q_{RES}) \tag{7.25}$$

Finally, the last subsystem of this level is $W_{G_{DM}}$. Applying RA to this subsystem while excluding $Pow$, $P_{DM}$, and $T_{FUEL}$, the following variables are obtained:

$$W_{G_{DM}} \;=\; f\;(P_{TUR}, W_{BYP}, W_{SR}) \tag{7.26}$$

This means that the output variable $Pow$ has three hierarchical levels and depends on 17 input variables (13 different).

For the other output variable, $Level$, the procedure is the same. Let us apply RA to this subsystem,

$$\begin{aligned} Level \;=\; & f\;(W_{FW}, Reac, W_{G_{DM}}, Scram, Voids, W_{FW_{ref}}, \\ & P_{TUR}, W_{BYP}, Dopler, W_{SR}, H_{FW}, P_C, W_{GF_{BW}}) \end{aligned} \tag{7.27}$$

13 variables are obtained. From them, those with the strongest binary relationships will be considered as intermediate subsystems. They are: $W_{FW}$, $Reac$, and $W_{G_{DM}}$. Thus, RA should be applied once more to these subsystems.

For the case of subsystem $W_{FW}$ (excluding *Level*, *Reac*, and $W_{G_{DM}}$), 2 variables are considered important. They are:

$$W_{FW} \;=\; f\;(W_{FW_{ref}}, P_C) \tag{7.28}$$

Applying RA to the second subsystem *Reac* (excluding *Level*, $W_{FW}$, and $W_{G_{DM}}$), 4 variables are obtained. They are:

$$Reac \;=\; f\;(Scram, Voids, Dopler, H_{FW}) \tag{7.29}$$

Finally, applying RA to the third subsystem, $W_{G_{DM}}$, (excluding *Level*, $W_{FW}$, and *Reac*), 4 variables are obtained. They are:

$$W_{G_{DM}} \;=\; f\;(P_{TUR}, W_{BYP}, W_{SR}, W_{GF_{BW}}) \tag{7.30}$$

This means that the output subsystem *Level* has two hierarchical levels and depends on 10 input variables, whereas the output subsystem *Pow* has three hierarchical levels and depends on 17 input variables (13 different).

The number of variables that the system depends on has been reduced from 51 to 27. However, since several variables are inputs to more than one subsystem, and even two subsystems are common to both output variables, i.e., subsystems *Reac* and $W_{G_{DM}}$, the total reduction is from 51 to 15. This means that the system can be qualitatively represented with less than one third of the original number of variables, i.e., an analogous reduction to that obtained in the *recirculation pump trip* case, in spite of this transient being much more involved.

Figure 7.14 represents the hierarchical arrangement of subsystems and input variables for the *feedwater control failure* case. As can be seen, there are no subsystems that coincide with true physical subsystems of the plant. However, when this transient really occurs, there are two variables that become very important due to the subcooling increase, they are: $T_{FUEL}$ and $T_{MOD}$. It can be noticed (from Figure 7.14) that these variables are two of the most important selected subsystems. They depend on enthalpies, reactivities, and the residual heat $Q_{RES}$. It can also be noticed that some other subsystems

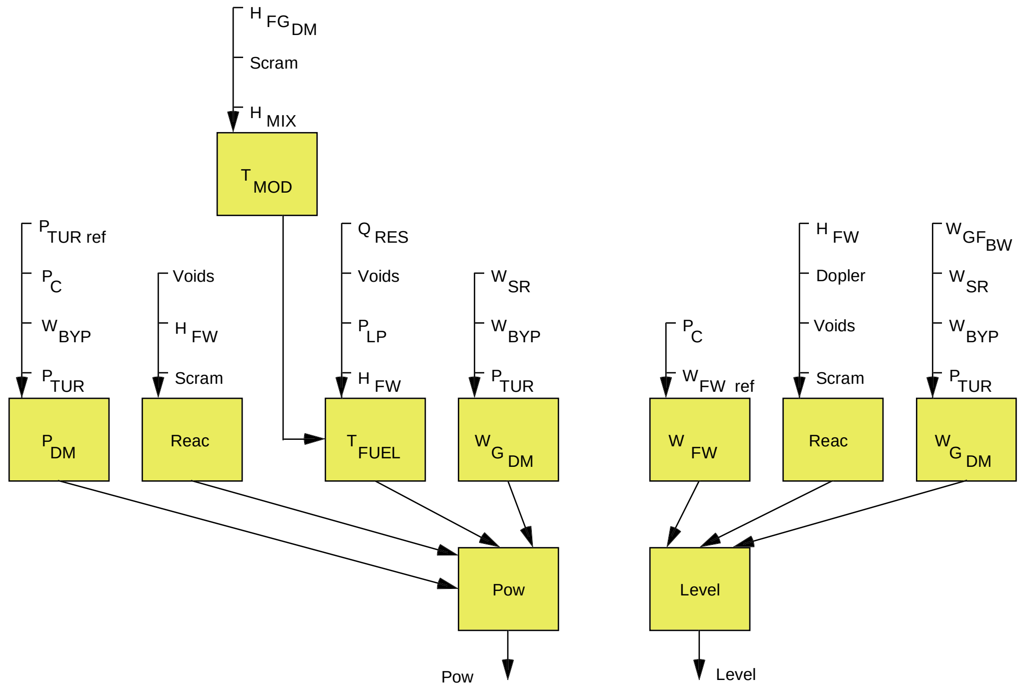


Figure 7.14: Hierarchy of subsystems and input variables for the feedwater controller failure transient.

depend on reactivity variables as well, especially on *Scram* (the reactivity introduced by the emergency shutdown).

#### 7.5.3.3 Optimal Masks

The mask parameters will be exactly the same as those used in the two previous cases. From Figure 7.14, it can be seen that one of the hierarchies has three levels, whereas the other has only two. Thus, applying Optimal Mask Analysis to each of the identified subsystems, the resulting optimal masks are, for the two first intermediate subsystems $P_{DM}$ and *Reac* of the left side hierarchy:

$$P_{DM} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} P_{TUR} & W_{BYP} & P_C & P_{TUR_{ref}} & P_{DM} \end{array} \\ \left(\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ -2 & 0 & 0 & -3 & 0 \\ -4 & 0 & 0 & -5 & 0 \\ 0 & -6 & 0 & 0 & +1 \end{array}\right) \end{array}$$

$$Reac \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} Scram & H_{FW} & Voids & Reac \end{array} \\ \left(\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & -2 & 0 & -3 \\ -4 & 0 & -5 & +1 \end{array}\right) \end{array}$$

Evidently, variable $P_C$ is being dropped. For the lowest hierarchical level subsystem $T_{MOD}$:

$$T_{MOD} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccc} H_{MIX} & Scram & H_{FG_{DM}} & T_{MOD} \end{array} \\ \left(\begin{array}{cccc} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 \\ -3 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -5 & 0 & +1 \end{array}\right) \end{array}$$

For the intermediate level subsystems $T_{FUEL}$ and $W_{G_{DM}}$:

$$T_{FUEL} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{cccccc} T_{MOD} & H_{FW} & P_{LP} & Voids & Q_{RES} & T_{FUEL} \end{array} \\ \left(\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & -3 & 0 \\ -4 & -5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & 0 & +1 \end{array}\right) \end{array}$$

$$W_{G_{DM}} \quad = \quad \begin{array}{c|ccccc} {}_t\backslash^x & P_{TUR} & W_{BYP} & W_{SR} & W_{GF_{BW}} & W_{G_{DM}} \\ \hline t-4\delta t & 0 & 0 & 0 & 0 & 0 \\ t-3\delta t & 0 & 0 & 0 & 0 & 0 \\ t-2\delta t & 0 & 0 & 0 & 0 & -1 \\ t-\delta t & -2 & 0 & 0 & 0 & 0 \\ t & 0 & -3 & -4 & 0 & +1 \end{array}$$

Variables $P_{LP}$ and $W_{GF_{BW}}$ are no longer considered important.

For the highest level executive variable $Pow$:

$$Pow \quad = \quad \begin{array}{c|ccccc} {}_t\backslash^x & P_{DM} & Reac & T_{FUEL} & W_{G_{DM}} & Pow \\ \hline t-4\delta t & 0 & 0 & 0 & 0 & 0 \\ t-3\delta t & 0 & 0 & -1 & 0 & 0 \\ t-2\delta t & -2 & 0 & -3 & 0 & 0 \\ t-\delta t & 0 & 0 & 0 & -4 & 0 \\ t & 0 & -5 & 0 & 0 & +1 \end{array}$$

Considering the right subhierarchy of Figure 7.14, the optimal masks for the subsystems at the intermediate level are:

$$W_{FW} \quad = \quad \begin{array}{c|ccc} {}_t\backslash^x & W_{FW_{ref}} & P_C & W_{FW} \\ \hline t-4\delta t & 0 & 0 & 0 \\ t-3\delta t & 0 & 0 & 0 \\ t-2\delta t & -1 & 0 & 0 \\ t-\delta t & 0 & -2 & 0 \\ t & -3 & -4 & +1 \end{array}$$

$$Reac \quad = \quad \begin{array}{c|ccccc} {}_t\backslash^x & Scram & Voids & Dopler & H_{FW} & Reac \\ \hline t-4\delta t & 0 & 0 & 0 & 0 & 0 \\ t-3\delta t & 0 & 0 & 0 & 0 & 0 \\ t-2\delta t & 0 & 0 & 0 & 0 & 0 \\ t-\delta t & 0 & -1 & -2 & -3 & -4 \\ t & -5 & -6 & 0 & 0 & +1 \end{array}$$

$$
W_{G_{DM}} \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array}
\begin{array}{c} \begin{array}{ccccc} P_{TUR} & W_{BYP} & W_{SR} & W_{GF_{BW}} & W_{G_{DM}} \end{array} \\
\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ -2 & 0 & 0 & 0 & 0 \\ 0 & -3 & -4 & 0 & +1 \end{pmatrix} \end{array}
$$

again dropping variable $W_{GF_{BW}}$.

For the top–level executive variable $Level$:

$$
Level \quad = \quad \begin{array}{l} {}_t\backslash^x \\ t-4\delta t \\ t-3\delta t \\ t-2\delta t \\ t-\delta t \\ t \end{array}
\begin{array}{c} \begin{array}{cccc} W_{FW} & Reac & W_{G_{DM}} & Level \end{array} \\
\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ -3 & -4 & -5 & +1 \end{pmatrix} \end{array}
$$

### 7.5.4 A FIR/RA–Based Continuous Fault Monitoring System

The idea of building a fully integrated fault monitoring system for the nuclear reactor was originally presented in [de Albornoz and Cellier, 1993b]. Although at that time neither the Reconstruction Analysis methodology nor the Forecasting All Possible Structures fault monitoring strategy were implemented yet, it was possible to construct:

1. A semi–continuous fault monitoring system with functions for early warning, since some potential problems can be discovered before they become emergencies.

2. A semi–continuous transient discovery system for quick detection of evolving emergencies. The transient discovery system was able to point out which of the subsystems were causing the problem, but was not able to analyze the precise nature of the anomaly. The on–line system operated with the optimal masks of the properly functioning plant exclusively, i.e., there were no masks for the transient conditions.

3. An off–line transient characterization and identification system used for post–mortem analysis.

That approach was only the first step. Since operational, non–operational, and serious transients often call for a decision to initiate a mitigation or emergency procedure within a few seconds, there is no way that a semi–continuous FIR/RA–based FMS could possibly aid the decision making process at such a time scale. Thus, it is essential for the FMS to be able to operate on–line and in a continuous–time way.

The on–line, continuous–time FIR/RA–based FMS that is presented here includes, beside from the previous three functions, procedures supporting the following tasks:

1. Early warning.
2. Fault detection.
3. Fault isolation.
4. Fault characterization.
5. Fault Diagnosis.
6. Fault Analysis.

Since plant safety is and must always be the highest priority, the only function offered by the FMS is that of presenting its findings to the human operators, trying to help them in their decision making process. This is accomplished by an early discovery of what transient is really going on (providing a confidence measure of its own prediction) while pointing out to its most probable causes.

The Forecasting All Possible Structures fault monitoring strategy will be used, thus, it is hoped that each of the three qualitative models would produce a decent prediction if the actual behavior of the reactor coincides with the behavior represented by the model, and a poor prediction when the aforementioned coincidence is not present.

The FMS will detect that a transient is taking place because the optimal masks no longer represent correctly the behavior of the system. An error threshold alarm matrix detects that the executive FIR's forecasting process contains too many errors. The comparison between the numerical variables and the forecast variables provide the values for the error matrix. The alarm matrix reads those values and triggers the alarm if a combination of consecutive incorrect forecasts and saturated states occurs. Once the executive FIR's alarm is triggered, the FMS proceeds downward to the next hierarchical level and checks the alarm matrices of the subsystems to try to determine which of them might have caused the transient to occur. Once a transient has been detected, the FMS will switch to the model that best represents the new situation.

If a single subsystem FIR triggers an alarm that is not picked up by the executive FIR's alarm matrix as well, the FMS is facing a small failure that eventually could cause the executive FIR to start the general alarm. This failure should be presented to the operators as an early warning to avoid the possibility of an operational transient later on, that might then lead to an unnecessary reactor shutdown.

The transients selected to demonstrate, in this thesis, the detection capabilities of the FMS are, beside the normal operation *full–power steady state*, a *recirculation pump trip*, and a *feedwater controller failure*. These three models have already been characterized in the preceding sections.

#### 7.5.4.1 Fault Monitoring of the Full–Power Steady State

In this section, the normal operation *full–power steady state* is maintained along 2500 sampling points. Since there is no transient scheduled, what can be seen in Figures 7.15 and 7.16 is how well the FIR/RA–based FMS has characterized the harmonic oscillations of fairly long period around the steady state. The continuous line represents the quantitative reactor model, whereas the dashed line represents the regenerated qualitative FIR model output.

These aforementioned figures represent the two executive subsystem variables *Pow* and *Level*, i.e., those that trigger the general detection alarm,[14] depicted in Figure 7.12, corresponding to the same time points of "normal reactor operation with harmonic perturbations."

Notice the extraordinary performance of the qualitative model that accurately follows its quantitative counterpart. This accuracy of the qualitative models will prove essential for the detection and identification of the reactor operational transients.

[14]Graphics for the alarms associated with the executive subsystems or lower level subsystems are not presented since no alarm will be triggered during this simulation.

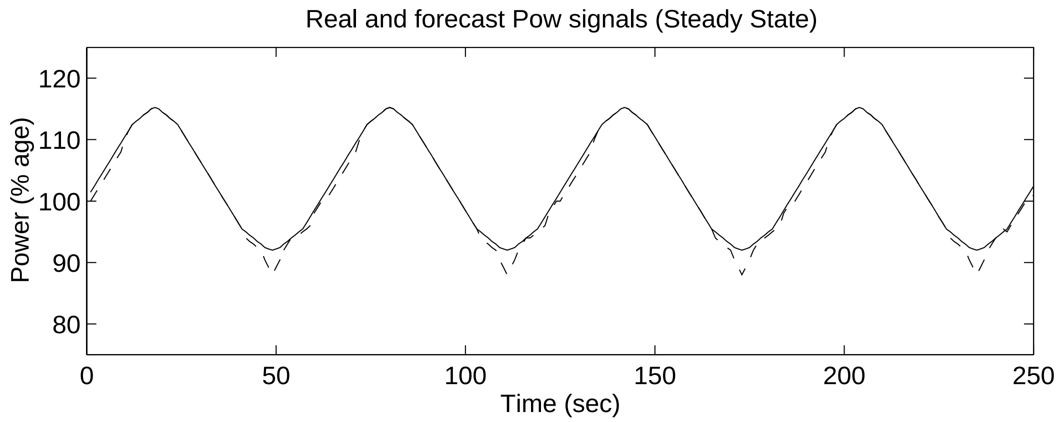


Figure 7.15: Real *vs.* forecast executive *Pow* subsystem variable.

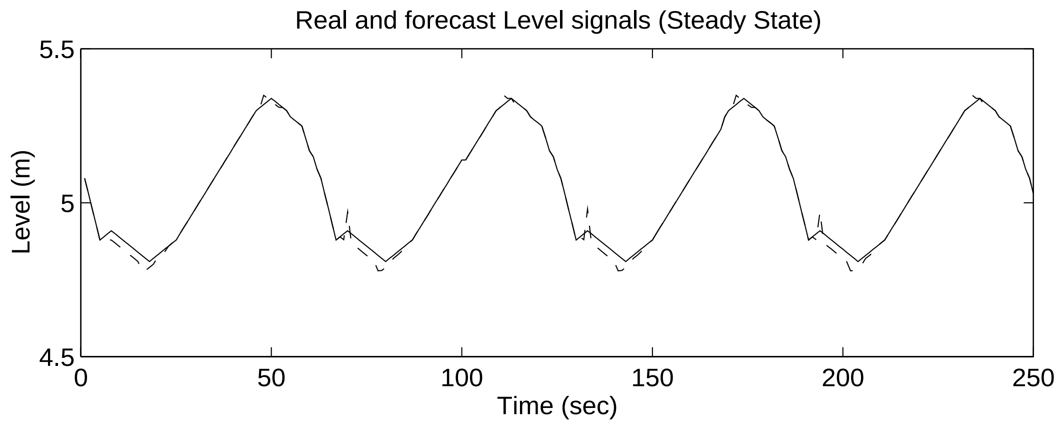


Figure 7.16: Real *vs.* forecast executive *Level* subsystem variable.

#### 7.5.4.2 Fault Monitoring of the Recirculation Pump Trip

Departing from a *full–power steady state*, a recirculation pump trip was initiated, in the numerical model, at time step 50. The graphics of Figures 7.17 and 7.18 represent the two executive subsystem variables *Pow* and *Level*, respectively, corresponding to the same time points of the "combined flow recirculation pump trip."

As can be seen in these two graphics, the quantitative variables are accurately followed by their qualitative counterparts, which means that the transient has been properly characterized. The qualitative model is able to perform splendid predictions of the thermal output power and water level during a recirculation pump trip operational transient.

In Figures 7.19 and 7.20 the real and forecast recirculation flows are

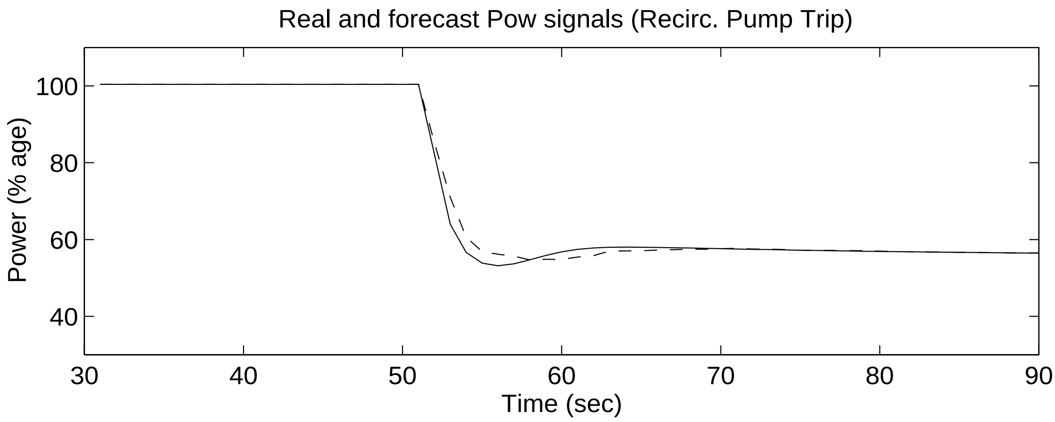


Figure 7.17: Real *vs.* forecast executive *Pow* subsystem variable.

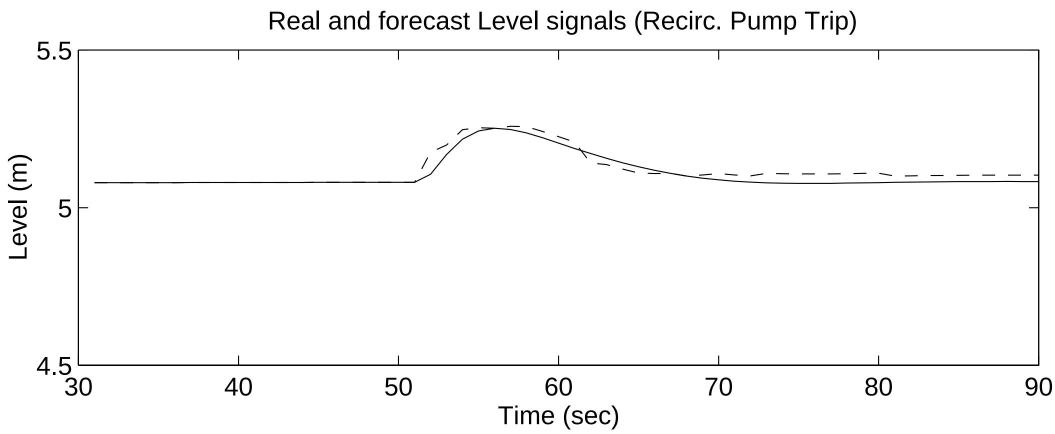


Figure 7.18: Real *vs.* forecast executive *Level* subsystem variable.

compared. Noticed that while the flow in the first loop becomes negative (reversed flow), the flow in the second loop is increased to its maximum. As a matter of fact, in these four graphics a transition between two qualitative models has been carried out, from the *full–power steady–state* model, to the *recirculation pump trip* model. The detection process is carried out by the former model, whereas the recognition process is carried out by the latter model.

From Figure 7.21, where the error and alarm matrices of the executive subsystems *Pow* and *Level* are depicted, it can be observed that the *Pow* variable detects the anomaly 11 sampling steps later; however, no alarm is triggered since the other variable *Level* reacts very slowly since the total amount of water in the reactor takes some time to change. Thus, the threshold built into the construction of the alarm matrix is not immediately reached by

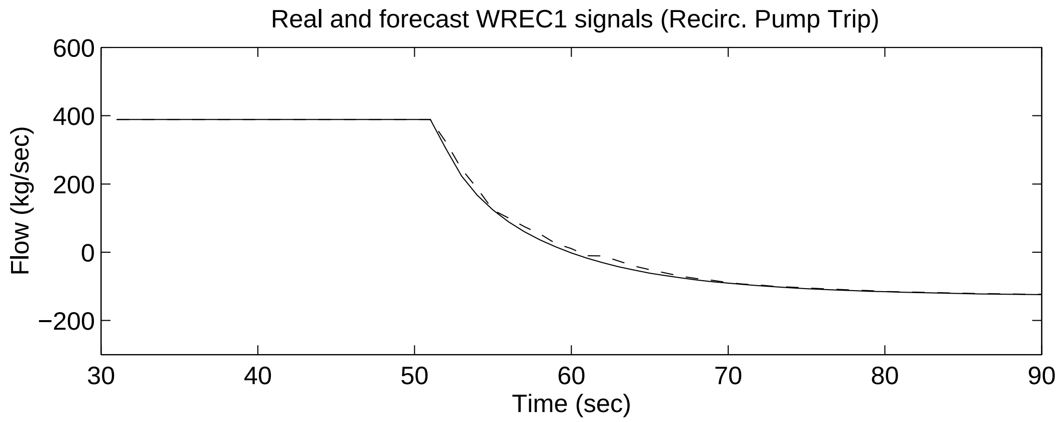


Figure 7.19: Real *vs.* forecast $W_{REC_1}$ subsystem variable.

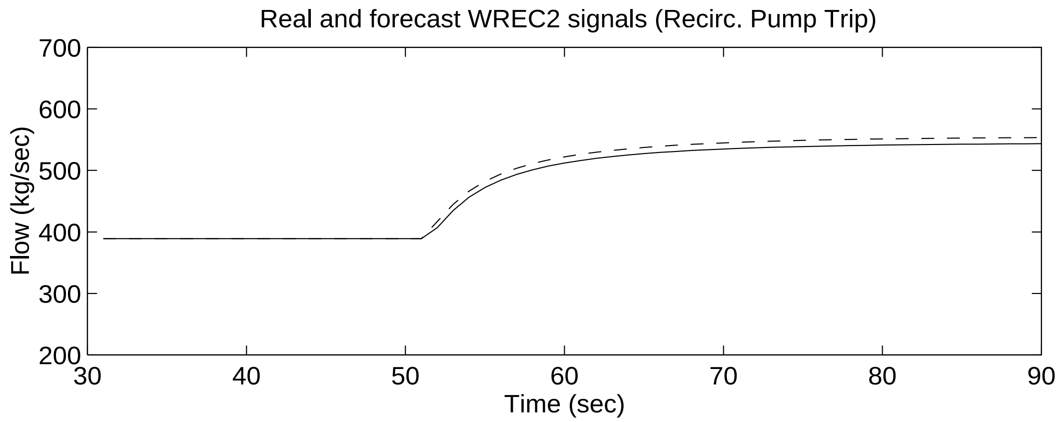


Figure 7.20: Real *vs.* forecast $W_{REC_2}$ subsystem variable.

the observed anomaly.

The second executive variable *Level* detects the transient after 19 sampling intervals, i.e., at sampling interval 69. The detection general alarm is triggered when both executive variables, at the same time, present three consecutive error predictions. Thus, the detection general alarm is triggered at time step 72. This means that the FMS is able to detect a misbehavior 2.5 seconds after it has been initiated! This represents a very good performance even compared to a quantitative FMS.

The transient was first detected by the $W_{REC_1}$ (recirculation loop 1 flow) subsystem. This subsystem is directly related to the transient through the $\Delta P_{PP_1}$ (pressure drop) and $W_{SUC_1}$ (suctioned flow) variables in the recirculation circuit. Figure 7.22 shows the error and alarm matrices of the $W_{REC_1}$ subsystem output variable, corresponding to the left subhierarchy in

$$\begin{array}{ccc}
 & \text{Error Matrix} & \text{Alarm Vector} \\
{}_{step}\backslash^{var} & \begin{array}{cc} \mathit{error\ Pow} & \mathit{error\ Level} \end{array} & \mathit{alarm} \\
\begin{array}{c} 50\\ 51\\ 52\\ 53\\ 54\\ 55\\ 56\\ 57\\ 58\\ 59\\ 60\\ 61\\ 62\\ 63\\ 64\\ 65\\ 66\\ 67\\ 68\\ 69\\ 70\\ 71\\ 72\\ 73\\ 74\\ 75 \end{array} &
\begin{pmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & -1 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
-1 & 0 \\
1 & 0 \\
1 & 0 \\
2 & 0 \\
3 & 0 \\
2 & 0 \\
2 & 0 \\
3 & 0 \\
3 & -2 \\
2 & -4 \\
2 & -3 \\
2 & -2 \\
3 & -2 \\
3 & -2 \\
3 & -1
\end{pmatrix} &
\begin{pmatrix}
0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 1\\ 1\\ 1\\ 1
\end{pmatrix}
\end{array}$$

Figure 7.21: Error and alarm matrices of the FMS executive subsystems during detection of the recirculation pump trip transient.

Figure 7.13. It can be observed that the transient is first detected only five sampling intervals after its beginning, at time steps 55 and 56, and then an obviously bad forecast accidentally produced a "good forecast" (a 0 error condition), still impeding the three consecutive errors needed to start the intermediate and high–level subsystem alarm, which is finally triggered at sampling interval 61.

At this moment, the subsystem $W_{REC_1}$ has discovered the beginning of a malfunction. If the transient would have stopped at this point, i.e., the magnitude of the reduction in the recirculated flow would have been of such a limited magnitude that no other subsystem alarm were triggered as well, the FMS would have reported its finding to the operators in the form of an early warning message.

| | Error Matrix | Alarm Vector |
|---|---|---|
| ${}_{step}\backslash^{var}$ | $error\ W_{REC_1}$ | $alarm$ |
| 50 | 0 | 0 |
| 51 | 0 | 0 |
| 52 | 0 | 0 |
| 53 | 0 | 0 |
| 54 | 0 | 0 |
| 55 | −1 | 0 |
| 56 | −2 | 0 |
| 57 | 0 | 0 |
| 58 | 3 | 0 |
| 59 | 2 | 0 |
| 60 | 3 | 0 |
| 61 | −1 | 1 |
| 62 | 4 | 1 |
| 63 | 3 | 1 |
| 64 | 3 | 1 |
| 65 | 4 | 1 |
| 66 | 4 | 1 |
| 67 | 4 | 1 |
| 68 | 3 | 1 |
| 69 | 3 | 1 |
| 70 | 3 | 1 |

Figure 7.22: Error and alarm matrices of the $W_{REC_1}$ subsystem during detection of the recirculation pump trip transient.

It should be noticed that the two subsystems $W_{REC_1}$ in the left and right subhierarchies of Figure 7.13 are not identical, since the one in the right subhierarchy depends also on the core pressure $P_C$. This makes its reaction to the transient slower and more smooth.

The detection alarm in the right subhierarchy is not produced by the $W_{REC_1}$ subsystem output variable, but by $W_{REC_2}$ that also depends on the core pressure $P_C$. This is the reason for the detection time differences between the executive variables *Pow* and *Level*. The detection process on subsystem $W_{REC_2}$ can be seen in Figure 7.23. The transient is first detected by this subsystem at the sampling interval 65, and the subsystem alarm is triggered three time intervals later, at time step 68, which in turn, triggers the executive subsystem *Level* alarm one sampling interval later, at time step 69, and the general alarm, as has been said before, at time step 72.

The subsystems *Reac* and $W_{G_{DM}}$ (reactivity and steam produced, respectively) detect the malfunction various sampling intervals later. Particularly, the reactivity is affected by a slight increase in the fuel and coolant

$$
\begin{array}{c}
 \\
{}_{step}\backslash^{var} \\
50\\51\\52\\53\\54\\55\\56\\57\\58\\59\\60\\61\\62\\63\\64\\65\\66\\67\\68\\69\\70\\71\\72
\end{array}
\begin{array}{c}
\text{Error Matrix} \\
error\ W_{REC_2} \\
\begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\-1\\-1\\-3\\-2\\-3\\-3\\-3\\-4 \end{pmatrix}
\end{array}
\qquad
\begin{array}{c}
\text{Alarm Vector} \\
alarm \\
\begin{pmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\1\\1\\1\\1\\1 \end{pmatrix}
\end{array}
$$

Figure 7.23: Error and alarm matrices of the $W_{REC_2}$ subsystem during detection of the recirculation pump trip transient.

temperatures, and also in the core voids. These two subsystems are affected through the lower plenum pressure $P_{LP}$ variable. The lower plenum pressure is affected by the flows $W_{SUC}$ and $W_{REC}$ from both recirculation loops. Thus the subsystem $P_{LP}$ is the one that transmits the transient effects into the core, the upper plenum, and the dome regions.

The transient is considered recognized when one of the qualitative models running in parallel is able to predict the new behavior of the reactor without a significant number of errors, i.e., with the general alarm being untriggered and the executive subsystems alarms untriggered during at least three consecutive time intervals. Remember that the predictions of the *full–power steady–state* model are now full of errors due to the detection of the transient.

Figure 7.24 depicts the error and alarm matrices of the executive subsystems *Pow* and *Level* of the recirculation pump trip model, during the recognition process. As can be seen, the new behavior is first recognized by the new model

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad \text{Error Matrix} \qquad\qquad\qquad\qquad \text{Alarm Vector} \\[6pt]
\begin{array}{c}
{}_{step}\backslash^{var} \\ 50\\51\\52\\53\\54\\55\\56\\57\\58\\59\\60\\61\\62\\63\\64\\65\\66\\67\\68\\69\\70\\71\\72\\73\\74\\75\\76\\77\\78\\79\\80\\81\\82\\83\\84\\85
\end{array}
\begin{array}{c}
\begin{array}{cc} \quad error\ Pow & error\ Level \end{array} \\
\begin{pmatrix}
-3 & 1 \\ -2 & 1 \\ -3 & 2 \\ -4 & -1 \\ -1 & -1 \\ -3 & 1 \\ -3 & 2 \\ -3 & 2 \\ -2 & 1 \\ -1 & -1 \\ -3 & 2 \\ -1 & 2 \\ -1 & 1 \\ -2 & 1 \\ -1 & 1 \\ -1 & 1 \\ 0 & 1 \\ 0 & 2 \\ 1 & 2 \\ -1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & -1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0
\end{pmatrix}
\end{array}
\qquad
\begin{array}{c}
alarm \\
\begin{pmatrix}
1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\0\\0\\0\\0\\0
\end{pmatrix}
\end{array}
\end{array}
$$

Figure 7.24: Error and alarm matrices of the executive *Pow* and *Level* subsystems during recognition of the recirculation pump trip transient.

no more than 16 sampling intervals after the accident, however the number of errors is not stable until 28 sampling intervals after the accident. This means that, although the error condition of the new qualitative model executive *Pow* variable disappears at time step 66 (the transient is not detected until time step 72), the general alarm, i.e., the three consecutive zero error condition in both executive subsystems, remains triggered until time step 81, that is, 31 sampling intervals after the accident, or 9.2 seconds after the transient started.

The behavior of the simulated reactor subsystems is a good approximation of what would be observed in a real reactor under similar circumstances. In a BWR nuclear reactor confronted with such a transient, $W_{REC_1}$ drops dramatically to even negative values, i.e., reversed flow in the recirculation circuit. The power *Pow* reacts immediately, whereas the pressure $P_C$ and the produced steam $W_{G_{DM}}$ decrease much more slowly.

### 7.5.4.3 Fault Monitoring of the Feedwater Controller Failure

As in the case of the previous transient, this one starts out from a *full–power steady state*. A feedwater controller failure of 110 % of normal demand is initiated, in the numerical model, at time step 50. The graphics of Figures 7.25 and 7.26 represent the two executive subsystem variables *Pow* and *Level*, respectively, corresponding to the same time points of the "combined feedwater control failure."

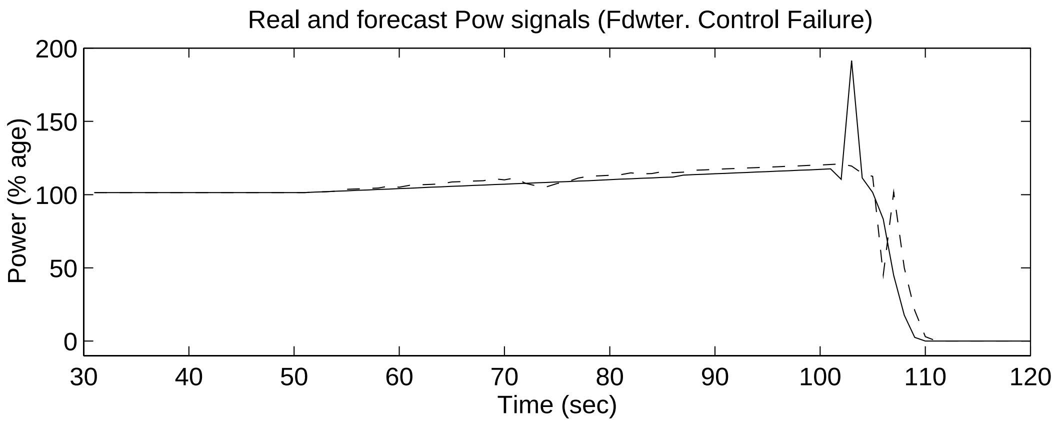


Figure 7.25: Real *vs.* forecast executive *Pow* subsystem variable.

As can be seen in these two graphics, the quantitative variables are accurately followed by their qualitative counterparts, which means that this transient has also been properly characterized. The qualitative model is once

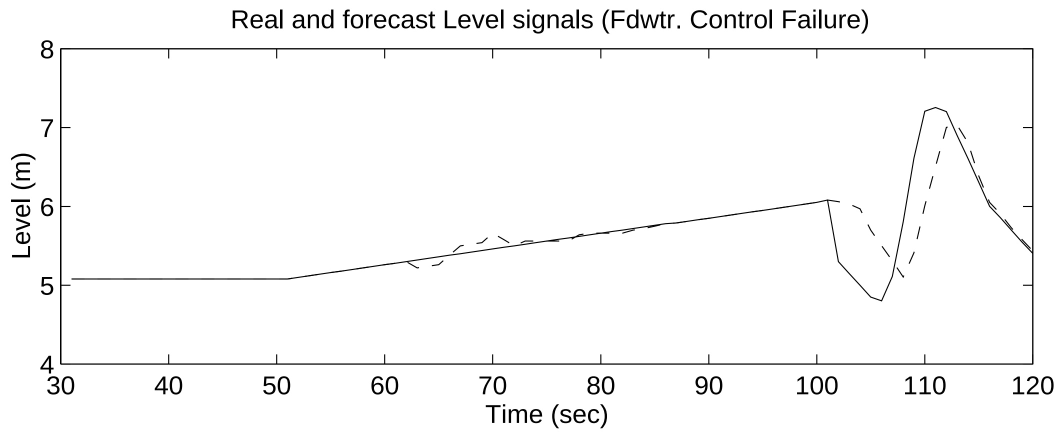


Figure 7.26: Real *vs.* forecast executive *Level* subsystem variable.

more able to provide very good predictions of the thermal output power and water level during a feedwater controller failure operational transient.

In Figures 7.27 and 7.28, the real and forecast fuel temperature and reactivity are compared. In these two graphs, two different transitions between qualitative models have been carried out. The first one was from the *full–power steady–state* model to the *feedwater controller failure* model, and the second was from this latter model to the *emergency stop Scram* model. The detection and recognition process are the following. The steady–state model detects that a transient is happening and then, the feedwater model recognizes which transient it was. In a second stage, this latter model detects a new transient happening, and the Scram model recognizes which transient it was.

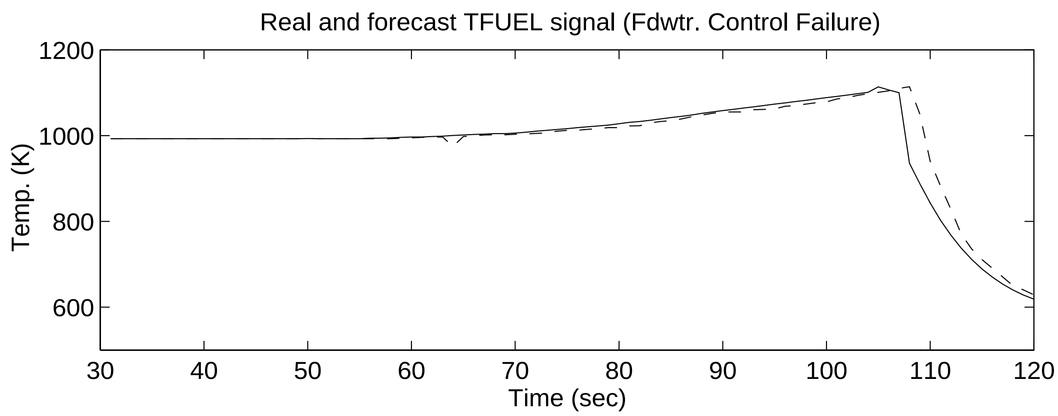


Figure 7.27: Real *vs.* forecast $T_{FUEL}$ subsystem variable.

From Figure 7.29, where the error and alarm matrices of the executive subsystems *Pow* and *Level* are depicted, it can be observed that the *Pow*

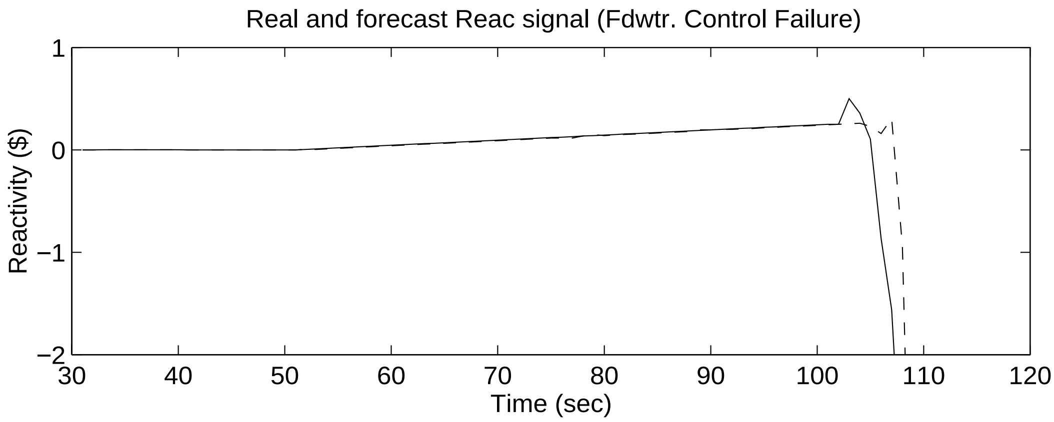


Figure 7.28: Real *vs.* forecast *Reac* subsystem variable.

variable detects the anomaly 34 sampling intervals later; however, the general alarm is not triggered because the other executive variable *Level* detects the transient only 48 sampling intervals after the accident.

Following the same rule of three consecutive error predictions of both executive subsystems, the detection general alarm is triggered 51 sampling intervals later, i.e., at time step 101. This means that the FMS is able to detect the transient 11.5 seconds after it has been initiated.

The transient was first detected by the $T_{FUEL}$ (fuel temperature) subsystem. This subsystem is directly related to the transient through the $H_{FW}$ (feedwater enthalpy) and $T_{MOD}$ (moderator–coolant temperature) variables in the core.

Figure 7.30 shows the error and alarm matrices of the $T_{FUEL}$ subsystem output variable, corresponding to the left hierarchy in Figure 7.14. It can be observed that the transient is first detected 23 sampling intervals after its beginning, i.e., at time step 73; however, the error condition is not maintained during three consecutive sampling intervals until time step 81, i.e., 9.23 seconds later.

At this moment, the subsystem $T_{FUEL}$ has discovered the beginning of the malfunction. If the transient would have stopped at this point, i.e., the magnitude of the increase in the feedwater controller set point would have been of such a limited magnitude that no other subsystem alarm were triggered as well, the FMS would have reported its finding to the operators in the form of an early warning message. Since this transient usually ends with an emergency reactor shutdown (Scram) due to a power excursion, a fast early warning of the developing anomaly may serve to prevent that shutdown.

The detection alarm in the right subhierarchy is triggered by the *Reac*

| | Error Matrix | | Alarm Vector |
|---|---|---|---|
| $_{step}\backslash^{var}$ | *error Pow* | *error Level* | *alarm* |
| 50 | 0 | 0 | 0 |
| 51 | 0 | 0 | 0 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| 75 | 1 | 0 | 0 |
| 76 | 0 | 0 | 0 |
| 77 | 0 | 0 | 0 |
| 78 | 0 | −1 | 0 |
| 79 | 0 | 0 | 0 |
| 80 | 0 | 0 | 0 |
| 81 | −1 | 0 | 0 |
| 82 | −1 | 0 | 0 |
| 83 | 0 | 0 | 0 |
| 84 | −2 | −1 | 0 |
| 85 | −2 | 0 | 0 |
| 86 | −2 | 0 | 0 |
| 87 | −1 | 0 | 0 |
| 88 | −2 | 0 | 0 |
| 89 | −2 | 0 | 0 |
| 90 | −2 | 0 | 0 |
| 91 | −2 | 0 | 0 |
| 92 | −2 | 0 | 0 |
| 93 | −1 | 0 | 0 |
| 94 | −1 | 0 | 0 |
| 95 | −3 | 0 | 0 |
| 96 | 1 | 0 | 0 |
| 97 | −2 | 0 | 0 |
| 98 | −2 | 1 | 0 |
| 99 | −2 | 1 | 0 |
| 100 | −2 | 1 | 0 |
| 101 | −2 | 1 | 1 |
| 102 | −2 | 1 | 1 |
| 103 | −2 | 2 | 1 |
| 104 | −2 | 1 | 1 |
| 105 | −2 | 1 | 1 |

Figure 7.29: Error and alarm matrices of the FMS executive subsystems during detection of the feedwater controller failure transient.

$$
\begin{array}{c}
\quad \text{Error Matrix} \qquad\qquad \text{Alarm Vector} \\
\begin{array}{c}
{}_{step}\backslash^{var} \\ 50 \\ 51 \\ \vdots \\ 70 \\ 71 \\ 72 \\ 73 \\ 74 \\ 75 \\ 76 \\ 77 \\ 78 \\ 79 \\ 80 \\ 81 \\ 82 \\ 83 \\ 84 \\ 85
\end{array}
\begin{array}{c}
error\ T_{FUEL} \\
\left(\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 2 \\ 1 \\ 0 \\ 1 \\ 2 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \end{array}\right)
\end{array}
\begin{array}{c}
alarm \\
\left(\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array}\right)
\end{array}
\end{array}
$$

Figure 7.30: Error and alarm matrices of the $T_{FUEL}$ subsystem during detection of the feedwater controller failure transient.

subsystem. Notice that this subsystem in the right subhierarchy of Figure 7.14 is not the same as the one in the left subhierarchy of the same figure. The one in the right side includes also the temperature associated reactivity *Dopler* input variable. Thus, it is much more sensitive to changes in the fuel temperature than its left side counterpart. This explains why the former triggers its subsystem alarm while the latter does not.

Once more, the transient is considered recognized when one of the qualitative models running in parallel is able to predict the new behavior of the reactor without a significant number of errors, i.e., with the general alarm untriggered and the executive subsystems alarms also untriggered during at least three consecutive time intervals. Remember that the predictions of the *full–power steady–state* model are now full of errors due to the detection of the transient.

Figure 7.31 depicts the error and alarm matrices of the executive subsystems *Pow* and *Level* of the feedwater controller failure model during the recognition process. As can be seen, the new behavior is recognized by the new model 99 sampling intervals after the accident, at time step 149.

$$
\begin{array}{c c c}
 & \text{Error Matrix} & \text{Alarm Vector} \\
{}_{step}\backslash^{var} & \begin{array}{cc} error\ Pow & error\ Level \end{array} & alarm \\
\begin{array}{c}
50\\51\\52\\53\\54\\\vdots\\105\\106\\107\\108\\109\\110\\111\\112\\113\\114\\\vdots\\140\\141\\142\\143\\144\\145\\146\\147\\148\\149\\150\\151\\152\\153\\154\\155
\end{array}
&
\begin{pmatrix}
3 & -2\\
-2 & -2\\
3 & -2\\
3 & -3\\
3 & -1\\
\vdots & \vdots\\
1 & -2\\
0 & -3\\
1 & 0\\
1 & -2\\
0 & -2\\
0 & -1\\
0 & -1\\
0 & -1\\
0 & 0\\
0 & 1\\
\vdots & \vdots\\
0 & -1\\
0 & -1\\
0 & -1\\
0 & 0\\
0 & -1\\
0 & -1\\
0 & 0\\
0 & 0\\
0 & 0\\
0 & 0\\
0 & 0\\
0 & 0\\
0 & 0\\
0 & 0\\
0 & 0\\
0 & 0
\end{pmatrix}
&
\begin{pmatrix}
1\\1\\1\\1\\1\\\vdots\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\\vdots\\1\\1\\1\\1\\1\\1\\1\\1\\1\\0\\0\\0\\0\\0\\0\\0
\end{pmatrix}
\end{array}
$$

Figure 7.31: Error and alarm matrices of the executive *Pow* and *Level* subsystems during recognition of the feedwater controller failure transient.

This means that 17.1 second after the transient began, the operators are informed about:

1. At time step 101 (11.5 sec), a developing transient has been detected.
2. At time step 149 (17.1 sec), this developing transient has been identified as a *feedwater controller failure*.
3. At time step 149, the operator is also informed that the most affected monitorable variables are the fuel temperature $T_{FUEL}$ and the core reactivity $Reac$, that in turn have affected the thermal output power $Pow$ and the water level $Level$, respectively.
4. At the same time step 149, the operator could be presented with the appropriate strategy to mitigate the transient and to avoid, if possible, the emergency shutdown.

It should also be noticed that when the emergency shutdown (Scram) is finally produced, the *feedwater controller failure* model detects that some other transient has been initiated, and then the *Scram* model recognizes the new behavior as that of an emergency shutdown. This last part of the transient has not been presented in detail since an emergency stop is easily detected by the operators in the control room, without the need of the FMS.

Let us suppose, however, that the operator takes into account the results of the FMS and follows the advice provided at time step 149 (17.1 seconds after the beginning of the accident), i.e., the correct strategy to mitigate the transient, thereby avoiding the emergency shutdown. Let us also suppose that the operator takes 25 seconds between the moment when he or she was informed and the instant when the corrective action is applied, which in this case is a change in the setpoint value of the feedwater controller.[15] 25 seconds are equivalent to 219 sampling intervals. Then, the corrective action is initiated by the operator at time step 373.

Figures 7.32 and 7.33 show the thermal output power and the water level, respectively, of the original transient with a corrective action taken by the operator at time step 373, equivalent to 42.5 seconds of simulation, and just 8 seconds before the emergency shutdown. Since the change in the feedwater controller setpoint is drastic and instantaneous, the transient is mitigated and the emergency shutdown is avoided, as can be seen when comparing this two graphs with those of Figures 7.25 and 7.26.

[15]If this corrective action does not succeed, the transient will continue until the emergency shutdown is triggered, as was to happen anyway.

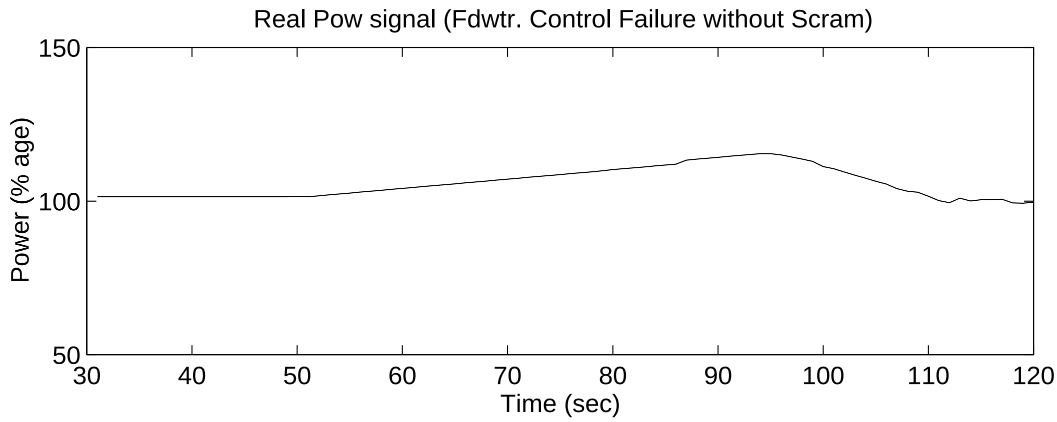


Figure 7.32: Real *vs.* forecast executive *Pow* subsystem variable.

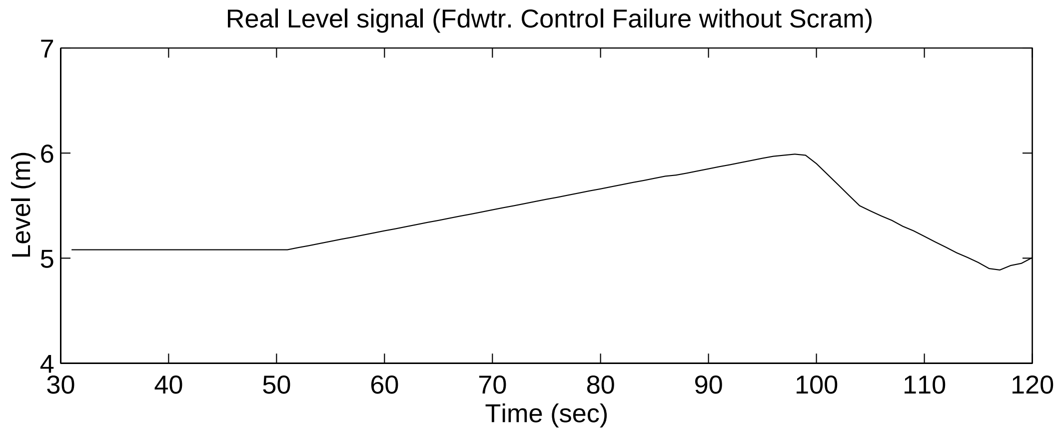


Figure 7.33: Real *vs.* forecast *Level* variable.

## 7.6 Conclusions

In this chapter, a prototypical implementation of an on–line continuous–time hierarchically structured Fault Monitoring System for large–scale systems based on Fuzzy Inductive Reasoning and Reconstruction Analysis was presented.

Two operational transients (beside the *steady–state full–power* operation and the *emergency shutdown, Scram*), namely a *recirculation pump trip* and a *feedwater controller failure*, have been detected, characterized, identified, and analyzed using the Forecasting All Possible Structures approach of the combined FIR/RA methodology. It is true that the two transients are completely different, and consequently not very much discriminatory power was needed to make distinctions between them; however, the important thing is that they are identified before a new steady state or an emergency shutdown condition are reached, i.e., when operator actions can overcome the problem or at least reduce its impact, as was demonstrated with the second transient.

It was indicated that such a Fault Monitoring System can be meaningfully used to help operators of large–scale engineering plants in their decision making process, specially under emergency conditions because of its capabilities for: **1.) early warning; 2.) fault detection; 3.) fault isolation; 4.) fault characterization; 5.) fault diagnosis; and 6.) fault analysis.**

The difficulties in constructing such an FMS were also stated. They relate to the problems of system excitation, variable selection, subsystem identification, hierarchical ordering of such identified subsystems, and the difficulties that stem from the need to identify subsystems in a closed–loop environment.

Besides the FIR/RA–based FMS itself, the other important tools that have been validated in this chapter are:

- The variable selection and subsystem identification processes by means of Reconstruction Analysis, that prove capable of drastically reducing the number of variables, as well as decomposing the overall system into a hierarchy of identified subsystems.
- The power of the qualitative modeling technique capable of representing the behavior of a nuclear reactor, including nuclear kinetics and thermohydraulic effects, during normal and abnormal operation, with just a few masks.
- The Forecasting All Possible Structures fault monitoring approach, which makes the fault detection and fault recognition processes very fast and accurate.

Not all the problems explained in this chapter have been completely overcome yet; however, they have been solved to the extent required for the task at hand. Whereas we were able to successfully fault–monitor the envisaged scenario of three different steady states (the original full–power steady state, the one reached after the recirculation pump trip, and the one reached after the emergency shutdown), two operational transients, and one emergency shutdown, we cannot claim that we have solved the problem of building a FMS that could be used to fault–monitor a wide palette of different operational transients. However, this was not the purpose. The goal was that of demonstrating that the combination of the Fuzzy Inductive Reasoning and Reconstruction Analysis techniques results in a powerful methodology (at least as powerful as Neural Networks) for qualitative simulation of real physical processes.

# Chapter 8

# Summary and Future Research

The objective of this chapter is not that of providing detailed conclusions of every single issue treated in this dissertation, since every chapter includes a conclusions section where these aspects were mentioned, but to summarize the results obtained, stress the main contributions of this research effort, mention how the techniques that were presented here can be further improved, and outline the characteristics and possible solutions to the problems that remain unsolved. Hence this chapter explains what has been accomplished, what goals have not been reached, and how the research might be continued in the future.

To this end, let us remember the motivation and objectives of this research. The motivation of this thesis was to help bridge the gap between the two worlds of quantitative computation and qualitative reasoning. Following this motivation, the objectives were stated as:

1. The development of a combined quantitative/qualitative modeling and simulation methodology to be applied to continuous–time dynamic processes.

2. The development and application of qualitative methodologies to solve information overload problems during normal and abnormal operation of quantitative complex large–scale systems.

3. The construction of a Fault Monitoring System based on the aforementioned combined quantitative and qualitative methodology, capable of mimicking the human assessment process when detecting, characterizing, identifying, diagnosing, and analyzing faults.

We argue that that these three objectives have been accomplished, since a powerful methodology capable of qualitatively fault monitoring quantitative large–scale systems in real–time has been developed and shown to be effective.

## 8.1 Summary of Results

In Chapter 3, the Fuzzy Inductive Reasoning (FIR) methodology was introduced and explained. The main characteristics of this methodology can be summarized as follows:

- No information is lost in the fuzzification and defuzzification processes.
- The qualitative models that represent the behavior of dynamic systems are always very simple.
- Inductive reasoning allows qualitative models to treat time as a continuous (quantitative) variable. This is of primary importance if modeling and simulation of mixed quantitative and qualitative systems is to be attempted.
- The technique can be applied to any system available to experimentation and observation. Inductive reasoning is fully based on behavior, thus, there is no need for knowing the internal structure of the system.
- The methodology contains an inherent model validation mechanism inside its qualitative simulation engine, which prevents it from reaching conclusions that are not justifiable on the basis of the available facts.
- Inductive reasoners operate internally in a qualitative fashion just like knowledge–based reasoners. Therefore, it is possible to apply meta–knowledge to improve the performance and quality of the inference engine, and it is also possible to trace back the reasoning process if desired.

In Chapter 4, the basis for the design of a FIR–based Fault Monitoring System (FMS) was presented. The resulting FMS proved capable of detecting, isolating, characterizing, identifying, diagnosing, and analyzing developing anomalies that can be considered faults or structural changes, depending on the dynamic system monitored. Some practical examples of the application of a FIR–based FMS were also presented to demonstrate the practicality of the *Qualitative Model Library* and the *Forecasting All Possible Structures* fault monitoring approaches when dealing with dynamic systems and variable structure systems, respectively.

The aircraft example had two objectives. On the one hand, the intention was that of demonstrating that a FIR–based FMS using the Qualitative Model Library fault monitoring approach, can be used as a watchdog autopilot to determine when a structural malfunction occurs, to hypothesize about the nature of this malfunction, and to decide upon a global strategy that allows to operate the quantitative aircraft model under the modified flying conditions. On the other hand, the intention was that of demonstrating the enhanced discriminatory power and the improved forecasting capabilities of a fuzzy inductive reasoning FMS in comparison with a crisp inductive reasoning FMS when applied to this problem.

The implementation of the *Forecasting All Possible Structures* approach demonstrated that the FIR–based FMS methodology is a powerful tool for structure characterization and identification in variable structure systems. The parallel forecasting of all possible structures enhances the previously used *Qualitative Model Library* scenario.

Two examples have been presented, a (fairly simple) two–water–tank problem, and an electrical circuit model containing three switches. The latter example was a very hard problem, and it is therefore proposed as a benchmark problem for structure identification by means of qualitative algorithms and codes. The results shown confirm that the combination of a FIR–based supervision and control system on–line with a quantitative dynamic system or VSS is a powerful tool that should be considered when qualitative fault monitoring of quantitative dynamic processes is to be attempted.

In Chapter 5, the Reconstruction Analysis (RA) methodology has been introduced, and demonstrated to be a valuable tool for subsystem identification and variable selection through causality analysis and refinement procedures. Reconstruction Analysis has been shown to be a well suited General Systems Problem Solver (GSPS) level 4 tool to be combined with FIR. Some of its properties are:

- RA is fully compatible with the previously developed FIR methodology.

- Some functions of FIR such as *recode*, *behavior*, *fbehavior*, and *iomodel* are also used in RA.

- FIR recodes the original behavior trajectories and feeds RA with this data. RA, by means of the Optimal Structure Analysis techniques, searches for suboptimal structures that can be considered as subsystems, and constructs a temporal causality hierarchy with these subsystems. Finally, RA feeds FIR with these subsystems, i.e., sets of variables, which, in turn, will construct qualitative models by means of the Optimal Mask Analysis, and carry out the reasoning process.

- Since both methodologies (FIR and RA) use the same recoded information and complement each other in the way described in the previous step, they can be considered as two facets of one and the same methodology.

In Chapter 6, a satisfactory comparison between Optimal Structure Analysis on the one hand, and Optimal Mask and Correlation Analyses on the other, gave us confidence that Reconstruction Analysis, and the heuristic recipes that have been designed around this methodology, constitute a sound and powerful tool for the selection and causal grouping of variables, i.e., Optimal Structure Analysis can be used as an alternative to Optimal Mask Analysis for finding high–quality qualitative models. It was also noted that the computational complexity of the Single Refinement algorithm used to determine the minimum set of meaningful variables is proportional to $n^2$, where $n$ denotes the number of variables in the system.

In Chapter 7, a prototypical implementation of an on–line continuous–time hierarchically structured Fault Monitoring System for large–scale systems based on Fuzzy Inductive Reasoning and Reconstruction Analysis was presented.

Two operational transients of a Boiling Water Nuclear Reactor (BWR), beside from the *full–power steady–state* operation and the *emergency shutdown, Scram*, namely a *recirculation pump trip* and a *feedwater controller failure*, have been detected, characterized, identified, and analyzed using the Forecasting All Possible Structures approach of the combined FIR/RA methodology. It is true that the two transients were completely different, and consequently not much discriminatory power was needed to make distinctions between them; however, the important point is that they were identified before a new steady state or an emergency shutdown condition were reached, i.e., when operator actions can still overcome the problem or at least reduce its impact, as was demonstrated with the second transient.

It was also indicated in the same chapter, that such a Fault Monitoring System can be meaningfully used to help operators of large–scale engineering plants in their decision making process, specially under emergency conditions, because of its capabilities for: **1.) early warning; 2.) fault detection; 3.) fault isolation; 4.) fault characterization; 5.) fault diagnosis; and 6.) fault analysis.**

## 8.2 Major Contributions of This Thesis

The major contributions of this thesis can be looked at from two different perspectives: from the point of view of advancing the methodology, and from the point of view of its applications. The contributions relating to the former perspective can be summarized as follows:

1. Enhancement of the Fuzzy Inductive Reasoning methodology by adding the necessary capabilities to deal with large–scale systems, i.e., for working with a large set of variables representing information that may be redundant and/or difficult to discriminate.

2. Development of a mixed quantitative/qualitative modeling and simulation environment. This task was done in cooperation with two other doctoral students as mentioned in the introductory section of Chapter 3.

3. Comparison between a crisp and a fuzzy inductive reasoner.

4. Implementation and application of the Reconstruction Analysis methodology to reduce, discriminate, and divide redundant information, i.e., to perform a temporal causality analysis among a large number of behavior variables, to find minimum sets of significant variables necessary to characterize different subsystems.

5. Development of a combined Fuzzy Inductive Reasoning and Reconstruction Analysis methodology for qualitative modeling and simulation of continuous–time large–scale processes.

6. Comparison of the capabilities of the Optimal Mask Analysis of the FIR methodology and the refinement procedures of the Optimal Structure Analysis of the Reconstruction Analysis for characterizing system behavior in qualitative terms, i.e., for qualitative modeling.

7. Design of a tool capable of reducing a large–scale system to a hierarchy of subsystems, each one represented by a separate FIR optimal mask.

From the point of view of the applications of the methodology, the contributions are:

1. A powerful qualitative tool for detecting, identifying, and characterizing different structural modes in quantitative variable structure systems.
2. Use of the combined Inductive Reasoning/Reconstruction Analysis methodology to detect, identify, characterize, and analyze misbehaviors related to malfunctions, faults, transients, and/or structural changes in quantitative continuous–time large–scale physical processes.
3. Combination of a qualitative fault monitoring and decision support system and a quantitative large–scale system for decision support under real–time constraints.

## 8.3 Future Research

Several problems remain unsolved. Some of them have to do with the combined methodology developed in this thesis, whereas others are concerned with the fault monitoring aspects of the application. Those problems related to the former aspect are:

- The landmarks between qualitative classes are normally set in such a way that all classes contain the same number of observations; however, this approach turned out to be inappropriate in some cases; thus, alternative techniques for establishing suitable landmarks should be proposed.
- Depending on the complexity and the non–linearity of the system under study, the optimal mask proposed by the Optimal Mask Analysis is not always the mask with the best forecasting capability. This obviously has to do with the implemented quality measure, and for this reason, complementary quality measures are currently being proposed and implemented by other students of the same research group. The true relationship between the quality of a mask and its forecasting power needs to be explored and quantified.
- One of the current lines of research in FIR deals with the introduction of additional information about qualitative knowledge of the variables. Although fuzzy recoding does not lose any information up front, as was shown, this does not guarantee that the total information available will

be utilized in the process of fuzzy inferencing. It may make sense to introduce *redundancy* into the qualitative data model in order to reduce the risk of losing information later on in the inferencing process. To this end, two different possibilities are now being explored:

i) *Causal Inductive Reasoning* (CIR), a new approach to inductive reasoning based on FIR, recodes quantitative values into qualitative *quadruples* instead of triples. Each quadruple contains the same three pieces of information as used in FIR, plus a qualitative derivative value that indicates whether the recoded variable is currently increasing, decreasing, or staying at about the same level.

ii) The potential inputs in a mask candidate matrix do not need to be numbered since their function is solely to point out, which are the $m$–input variables that may possibly affect the behavior of the $m$–output variable of the mask; however, it would make sense to introduce an additional value, e.g. $(-2)$, to denote *required inputs*. In this case, only masks would be evaluated that contain the variables identified by this marker in the mask candidate matrix, reducing the overall search effort.

With respect to the problems derived from the fault monitoring application of the combined FIR/RA methodology, the following aspects should be tackled in the future:

- The applicability of the Forecasting All Possible Structures fault monitoring approach is limited by the number of qualitative models that can be computed at the same time in parallel. One solution that could be studied is the possibility of grouping related faults in such a way that a single model may represent the behavior of more than one fault.

- Several aspects of the fault monitoring strategies still need to be automated, as for example the heuristic rules for the reduction of the number of variables proposed in Chapter 6, and the setting of the threshold values for the moving average error counters.

- It was mentioned in Chapter 2 that, beside from Expert Systems and Neural Networks, (FIR has already been successfuly compared against Expert Systems and Neural Networks in the biomedical domain), there are some other qualitative methodologies for fault monitoring of physical processes such as *QSIM* and *QUALSIM*. A set of benchmark problems should be created in order to carry out a detailed comparison between FIR/RA and the other methodologies.

Not all the problems explained in this thesis have been completely overcome yet; however, they have been solved to the extent required for the task at hand. Whereas we were able to successfully fault–monitor the envisaged scenario of four different examples, we cannot claim that we have solved the problem of building an FMS that could be used to fault–monitor a wide palette of different dynamic systems. However, this was not the purpose. The goal was that of demonstrating that the combination of the Fuzzy Inductive Reasoning and Reconstruction Analysis techniques results in a powerful methodology (at least as powerful as Neural Networks) for qualitative simulation of real physical processes.

The research effort will continue. While there remain many problems yet to be addressed, we are quite excited about the possibilities of the chosen approach. We believe strongly that the FIR/RA methodology offers great opportunities in tackling a number of "hot" issues in applied Artificial Intelligence, difficult issues that, to this date, other methodologies have not been able to master.

# Chapter A

# Bibliography

## A.1 Publications


**[Cellier, F.E., A. Nebot, F. Mugica, and A. de Albornoz (1992)]**, "Combined Qualitative/Quantitative Simulation Models of Continuous–Time Processes Using Fuzzy Inductive Reasoning Techniques," *Proc. SICICA'92, IFAC Symposium on Intelligent Components and Instruments for Control Applications*, Málaga, Spain, May 20–22, pp. 589–593.

**[Cellier, F.E., A. de Albornoz, and A. Uhrmacher (1996b)]**, "SAPS–II: A Research Tool for Reconstruction Analysis," *International Journal of General Systems*, to be submitted shortly.

**[Cellier, F.E., A. Nebot, F. Mugica, and A. de Albornoz (1996a)]**, "Combined Qualitative/Quantitative Simulation Models of Continuous–Time Processes Using Fuzzy Inductive Reasoning Techniques," in *International Journal of General Systems*, **24**(1–2), pp. 95–116.

**[de Albornoz, A., and F.E. Cellier (1993a)]**, "Qualitative Simulation Applied to Reason Inductively About the Behavior of a Quantitatively Simulated Aircraft Model," *Proc. QUARDET'93, IMACS Intl. Workshop on Qualitative Reasoning and Decision Technologies*, Barcelona, Spain, June 16–18, pp. 711–721.

**[de Albornoz, A., and F.E. Cellier (1993b)]**, "Variable Selection and Sensor Fusion in Automatic Hierarchical Fault Monitoring of Large–Scale Systems," *Proc. QUARDET'93, IMACS Intl. Workshop on Qualitative Reasoning and Decision Technologies*, Barcelona, Spain, June 16–18, pp. 722–734.

**[de Albornoz, A., and F.E. Cellier (1994)]**, "Building Intelligence into an Autopilot – Using Qualitative Simulation to Support Global Decision Making," *Simulation*, **62**(6), pp. 354–363.

**[de Albornoz, A. and F.E. Cellier (1996)]**, "The Use of Reconstruction Analysis for Obtaining Inductive Models of Dynamic Systems with High Forecasting Power," *International Journal of General Systems*, to be submitted shortly.

**[de Albornoz, A., F.E. Cellier, and J. Sardá (1994)]**, "Structure Identification in Variable Structure Systems by Means of Qualitative Simulation," *Proceedings SCS–ESM'94 Qualitative Information, Fuzzy Techniques, and Neural Networks in Simulation ICQFN'94*, Barcelona, Spain, pp. 486–491.

**[de Albornoz, A., F.E. Cellier, and R.M. Huber (1996b)]**,"A Systems–Theoretic Approach for Dealing with Large–Scale Systems," *International Journal of General Systems*, to be submitted shortly.

**[de Albornoz, A., F.E. Cellier, and J. Sardá (1996a)]**, "Structure Identification in Variable Structure Systems by Means of Qualitative Simulation," *Mathematical Modeling of Systems*, accepted for publication.

## A.2 References

**[Aguilar–Martín, J. (1991)]**, "Knowledge–Based Real Time Supervision of Dynamical Processes," in *IEEE Trans. on Expert Systems*, April 1992.

**[Air France, 1989]**, *A 310 Pilote Automatique/Directeur de Vol (PA/DV)*, Description I–22. 71. 02, 1ère Edition.

**[Aliev, R.A., F.T. Aliev, and M.D. Babaev (1992)]**, "The Synthesis of a Fuzzy Coordinate–Parametric Automatic Control System for an Oil–Refinery Unit," in *Fuzzy Sets and Systems*, **47**, pp. 157–162.

**[Allen, J.F., (1984)]**, "Towards a General Theory of Action and Time," in *Artificial Intelligence*. Vol. 23.

**[Allen, J.F and H.A. Kautz (1985)]**, "A Model of Naive Temporal Reasoning," in *Formal Theories of the Commonsense World*, (Hobbs & Moore, Eds.), Ablex Publishing

Co., Norwood, N.J., USA, pp. 251–268.

[**Babbie, E. (1989)**, *The Practice of Social Research,* 5th Edition, Wadsworth Publishing Company, Belmont, Calif., USA.

[**Bartlett, E.B. (1991)**], "Nuclear Power Plant Status Diagnostics Using an Artificial Neural Network," *Nuclear Technology*, **97**, pp. 272–281.

[**Basseville, M. and I. Nikiforov (Eds.) (1993)**], *Detection of Abrupt Changes: Theory and Applications*, Prentice Hall Information and Systems Science Series, USA.

[**Basseville, M. and A. Benveniste (Eds.) (1985)**], "Detection of Abrupt Changes in Signals and Dynamical Systems," in *Lecture Notes in Control and Information Sciences*, **77** Springer–Verlag.

[**Beard, R.V. (1971)**], *Failure Accomodation in Linear Systems Through Self–Reorganization*, Report Dept. MVT–71–1, Man Vehicle Laboratory, Cambridge, MA., USA.

[**Berkan, R.C., B.R. Upadhyaya, and L.H. Tsoukalas (1991)**], "Advanced Automation Concepts for Large-Scale Systems," *IEEE Control Systems*, October 1991, pp. 4–12.

[**Blalock, H.M. (1985)**], *Causal Models in Panel and Experimental Design*, Adline Publishing Co., USA.

[**Bobrow, D.G. Ed.(1984)**], "Qualitative Reasoning About Physical Systems," *Artificial Intelligence*, **24**.

[**Boullart, L., A. Krijgsman, and R.A. Vingerhoeds (Eds.) (1992)**] *Applications of Artificial Intelligence in Process Control*, Pergamon Press, London, U.K.

[**Brauer, F. and J.A. Nobel (1969)**], *Qualitative Theory of Ordinary Differential Equations*, W.A. Benjamin Inc.

[**Brown, J. and C.R. Rao (1985)**], *Sensor Failure Detection and Isolation in Multiengine Aircraft*, NASA Contractor Report CR–174846, General Electric Co., Aircraft Engine Group, Cincinnati, OH, USA.

[**Buchanan, B.G., and E.H. Shortliffe (1984)**], *Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, MA: Addison–Wesley, U.S.A.

[**Cacciabue, P.C., G. Mancini, and U. Bersini (1990)**], "A Model of Operator Behavior for Man–Machine System Simulation," *Automatica*, **26**(6), pp. 1025–1034.

[**Carbonell, J. and S. Minton (1985)**], "Metaphor and Commonsense Reasoning," in *Formal Theories of the Commonsense World*, (Hobbs & Moore, Eds.), Ablex Publishing Co., Norwood, N.J., USA, pp. 405–426.

[**Cavallo, R.E. and G.J. Klir (1981)**], "Reconstructability Analysis: Evaluation of Reconstruction Hypotheses," in *International Journal of General Systems*, **7**, pp. 7–32.

[**Cavallo, R.E and G.J. Klir (1982)**], "Reconstruction of Possibilistic Behavior Systems," in *Fuzzy Sets and Systems*, **8**, pp. 175–197.

[**Cellier, F.E. (1987)**], "Qualitative Simulation of Technical Systems Using the General

Systems Problem Solving Framework," in *International Journal of General Systems*, **13**(4), pp. 333–344.

**[Cellier, F.E. (1991a)]**, *Continuous System Modeling*, Springer–Verlag, New York, USA.

**[Cellier, F.E. (1991b)]**, "Qualitative Modeling and Simulation – Promise or Illusion," in *Proceedings of the SCS 1991 Winter Simulation Conference*, Phoenix, AZ, USA, pp. 1086–1090.

**[Cellier, F.E. (1991c)]**, "General System Problem Solving Paradigm for Qualitative Modeling," in *Qualitative Simulation Modeling and Analysis*, (Fishwick & Luker, Eds.), Springer–Verlag, USA, pp. 51–71.

**[Cellier, F.E. and J. López (1995)]**, "Causal Inductive Reasoning: A New Paradigm for Data–Driven Qualitative Simulation of Continuous–Time Dynamical Systems," plenary paper presented in *Proc. of the SCS ESM'95 European Simulation Multiconference,* Prague, Czech Republic, June 7–5.

**[Cellier, F.E. and F. Mugica (1992)]**, "Systematic Design of Fuzzy Controllers Using Inductive Reasoning," in *Proc. IEEE Intelligent Control Conference*, Glasgow, Scotland, U.K., August 11–13, pp. 198–203.

**[Cellier, F.E. and D.W. Yandell (1987)]**, "SAPS–II: A New Implementation of the Systems Approach Problem Solver," in *International Journal of General Systems*, **13**(4), pp. 307–322.

**[Cellier, F.E., L.C. Schooley, M.K. Sundareshan, and B.P. Zeigler (1992a)]**, "Computer–Aided Design of Intelligent Controllers: Challenge of the Nineties," in: *Recent Advances in Computer Aided Control Systems Engineering* (M. Jamshidi and C.J. Herget, eds.), Elsevier Science Publishers, Amsterdam, the Netherlands, pp. 53–80.

**[Cellier, F.E., A. Doser, G. Farrenkopf, J. Kim, Y.D. Pan, L.C. Schooley, B. Williams, and B.P. Zeigler (1992b)]**, "Watchdog Monitor Prevents Martian Oxygen Production Plant from Shutting Itself Down During Storm," in *Proceedings of the ISRAM'92 ASME Conference*, Santa Fe, N.M., USA, pp. 697–704.

**[Cellier, F.E., J. López, A. Nebot, and G. Cembrano (1996c)]**, "Means for Estimating the Forecasting Error in Fuzzy Inductive Reasoning," in *Proceedings of the ESM'96 Intl. Conf. on Qualitative Information, Fuzzy Systems, and Neural Networks in Simulation*, Budapest, Hungary.

**[Charniak, E. and D. McDermott (1987)]**, *Introduction to Artificial Intelligence*, Addison–Wesley, USA.

**[Chow, E.Y. and A.S. Willsky (1984)]**, "Analytical Redundancy and the Design ofControl Failure Detection systems," in *IEEE Trans. on Automatic Control*, **AC 29**, pp. 603–614.

**[çinar, A., T.E. Marlin, and J.F. McGregor (1992)]**, "Automated Monitoring and Assessment of Controller Performance," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 44–48.

**[Clark, Robert N. (1978a)]**, "A Simplified Instrument Failure Detection Scheme," in *IEEE Trans. on Aerospace and Electronic Systems*, **14**, pp. 558–563.

**[Clark, Robert N. (1978b)]**, "Instrument Fault Detection," in *IEEE Trans. on Aerospace and Electronic Systems*, **14**, pp. 456–465.

**[Clark, Robert N. (1989)]**, "State Estimation Schemes for Instrument Fault Detection," in (R. Patton, P. Frank, R. Clark eds.) *Fault Diagnosis in Dynamic Systems, Theory and Applications*, Prentice Hall, U.K., pp. 21–45.

**[Clark, R.N., D.C. Fosth, and V.M. Walton (1975)]**, "Detection Instrument Malfunctions in Control Systems," in *IEEE Trans. Electron. Sys.*, **11**, pp. 465–473.

**[Conant, R.C. (1981)]**, "Detection and Analysis of Dependency Structures," in *International Journal of General Systems,* **7**(1), pp. 81–91.

**[Conant, R.C. (1988a)]**, "Extended Dependency Analysis of Large Systems, Part I: Static Analysis," in *International Journal of General Systems*, **14**, pp. 97–123.

**[Conant, R.C. (1988b)]**, "Extended Dependency Analysis of Large Systems, Part II: Dynamic Analysis," in *International Journal of General Systems*, **14**, pp. 125–141.

**[Cordes, G.A.** *et al.,* **(1990)]**, "Neural Network Setpoint Control of an Advanced Test Reactor Experiment Loop Simulation," EGG-EE-8935, Sept., 1990.

**[Cortés, U., J. Béjar, and A. Moreno (1993)]**, *Inteligencia Artificial,* Colecció Politext, Edicions Universitat Politècnica de Catalunya, Barcelona, España.

**[Crespo, Alfons (1993)]**, "Real–Time Expert Systems," in *Applications of Artificial Intelligence in Process Control,* (L. Boullart, A. Krijgsman, and R.A. Vingerhoeds Eds.), Pergamon Press, UK.

**[Cuena, J., G. Fernández, R.L. de Mántaras, and M.F. Verdejo, (1986)]**, in *Inteligencia Artificial: Sistemas Expertos,* Alianza Editorial, Barcelona, España.

**[Cuena, J. (1987)]**, "Concepto y Métodos de Construcción de Sistemas Expertos," in *Inteligencia Artificial,* (J. Mompín, Ed.), Marcombo Boixareu Eds. Barcelona, España, pp. 81–91.

**[Davis, R. (1982)]**, "Diagnosis Based on Structure and Function," in *Proceedings of the National Conf. on Artificial Intelligence*, Pittsburgh, Penn., USA., pp. 137–142.

**[Davis, R. (1984)]**, "Diagnostic Reasoning Based on Structure and Behavior," in *Artificial Intelligence*, **24**.

**[Davis, R., and H. Shrobe (1982)]**, "Diagnosis Based on Structure and Function," in *Proc. od AAAI'82,* Pittsburgh, Morgan Kaufman Publishers Inc., San Mateo, CA, USA, pp. 137–142.

**[Davis, R., and W. Hamscher, (1988)]**, "Model–Based Reasoning: Troubleshooting," in *Exploring Artificial Intelligence*, (H.E. Shrobe, Ed.), Morgan Kaufman Publishers Inc., San Mateo, CA, USA.

**[Deckert, J.C., M.N. Desai, J.J Deyst, and A.S. Willsky (1977)]**, "DFBW Sensor Failure Identification Using Analytical Redundancy," in *IEEE Trans. on Automatic Control*, **22**, pp. 795–809.

**[Dempster, A. (1968)]**, "A Generalization of Bayesian Inference," in *Jour. of Stat. Soc.*, **30**(b), pp. 205–247.

**[Desai, M. and A. Ray (1981)]**, "A Fault Detection and Isolation Methodology," in *Proc. of the 20th Conference on Decision and Control*, pp. 1363–1369.

**[Dvorak, D. and B. Kuipers (1989)]**, "Model–Based Monitoring of Dynamic Systems, in *Proceedings of the 11th Intl. Conf. on Artificial Intelligence*, Detroit, Mich., USA., pp. 1238–1243.

**[Duderstadt, J.J., and L.J. Hamilton (1976)]**, *Nuclear Reactor Analysis*, John Wiley & Sons, U.S.A.

**[de Kleer, J. (1975)]**, *Qualitative and Quantitative Knowledge in Classical Mechanics,* TR-352 MIT, Artificial Intelligence Laboratory, Cambridge, MA, USA.

**[de Kleer, J. and G.J. Sussman (1980)]**, "Propagation of Constraints Applied to Circuit Synthesis," in *Circuit Theory and Applications*, **8**.

**[de Kleer, J. and J.S. Brown (1982)]**, "Foundations of Envisioning," in *Proceedings of the National Conference on Artificial Intelligence (AAAI-82)*, Aug. 18–20.

**[de Kleer, J. and J.S. Brown (1984)]**, "A Qualitative Physics Based on Confluences," in *Artificial Intelligence*, **24**.

**[de Kleer, J. and B. Williams (1987)]**, "Diagnosing Multiple Faults," in *Artificial Intelligence*, **32**(1), pp. 97–130.

**[de Kleer, J. and b. Williams (1989)]**, "Diagnosis with Behavioral Modes," in *Proc. of the 11th Intl. Conf. on Artificial Intelligence*, Detroit, Mich., USA.

**[Elmqvist H. (1993)]**, *Dymola: Dynamic Modeling Language — User's Guide*, Dynasim AB, Lund, Sweden.

**[Elmqvist H., F.E. Cellier, and M. Otter (1993)]**, "Object–Oriented Modeling of Hybrid Systems," in *Proc. ESS'93, European Simulation Symposium*, Delft, The Netherlands, October 25–28, pp. xxxi–xli.

**[Emami–Naeini, A.** *et al.*, **(1986)]**, *Robust Detection, Isolation, and Accomodation for Sensor Failures*, NASA Contractor Report CR–174825, Alphatec Inc., Burlington, Ma. USA.

**[Emelyanov, S.V. (1970)]**, "Design of Variable Structure Systems With Discontinuous Switching Functions," in *Eng. Cybern.*, **1**, pp. 156–160.

**[Enand, S. and E.A. Scarl (1989)]**, "Building Functional Models for Fault Location and Control," in *Proceedings of the 1989 Summer Computer Simulation Conference*, Austin, TX, USA.

**[Fathi, Z., W.F. Ramirez, A.P. Tavares, G. Gilliland and J. Korbicz (1991)]**, "A Knowledge–Based System with Embedded Estimation Components for Fault Detection and Isolation in Process Plants," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 32–37.

**[Feyock, S. (1987)]**, "Fault Diagnosis Based on Qualitative Models," in *Second A.I. and Simulation Workshop (AAAI'87)*, Seattle, WA, USA.

**[Firebaugh, M.W. (1988)]**, *Artificial Intelligence, a Knowledge–Based Approach*, Boyd

& Fraser Publishing Co., Boston, Mass., USA.

[**First, M.B., B.J. Weimer, and S. McLinden, (1982)**], "LOCALIZE: Computer–Assisted Localization of Peripheral Nervous System Lesions," in *Computers and Biomedical Research*, **15**(6), pp. 525–543.

[**Fishwick, P.A. (1988)**], "Qualitative Simulation: Fundamental Concepts and Issues," in *A.I. and Simulation 1988*, The Society for Computer Simulation, USA, pp. 25–31.

[**Fishwick, P.A. (1991)**], "Fuzzy Simulation: Specifying and Identifying Qualitative Models," in *International Journal of General Systems*, **19**(3), pp. 295–316.

[**Forbus, K.D. (1981)**], "Qualitative Reasoning About Physical Processes", *Proceedings of the Seventh Intl. Joint Conf. on Artificial Intelligence (IJCAI-81),* Vancouver, B.C., Canada.

[**Forbus,K.D. and A. Stevens, (1981)**], "Using Qualitative Simulation to Generate Explanations," BBN Technical Report No. 4490, Cambridge, Ma., USA. (also in *Proc. of the Third Annual Conference of the Cognitive Science Society*, Aug., 1991.)

[**Forbus, K.D. (1984)**], "Qualitative Process Theory," *Artificial Intelligence*, **24**, pp 85–167.

[**Forbus, K.D. (1985)**], "The Role of Qualitative Dynamics in Naive Physics," in *Formal Theories of the Commonsense World*, (Hobbs & Moore, Eds.), Ablex Publishing Co., Norwood, N.J., USA, pp 185–226.

[**Forbus, K.D. and J. de Kleer (1993)**], *Building Problem Solvers,* The MIT Press, Cambridge, Mass., USA.

[**Forgy, C. (1982)**], "RETE: A Fast Algorithm for Many Pattern / Many Object Pattern Matching Problem," in *Artificial Intelligence*, **19**, North–Holland.

[**Foss, B.A. and T.A. Johansen (1992)**], "An Integrated Approach to On–Line Fault Detection and Diagnosis, Including Artificial Neural Networks with Local Basis Functions," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 150–155.

[**Frank, P.M. (1987a)**], "Advanced Fault Detection and Isolation Schemes Using Nonlinear and Robust Observers," in *Proc. of the 10th IFAC World Congress*, Munich, Germany, pp. 429–434.

[**Frank, P.M. (1987b)**], "Fault Diagnosis in Dynamic Systems via State Estimation – A Survey," in *Proc. of the 1st European Workshop on Fault Duagnostics, Reliability and Related Knowledge–Based Approaches*, Rhodes, Greece, Reidel Press, Vol. 1, pp. 35–98.

[**Frank, P.M. (1990)**], "Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge–Based Redundancy—-A Survey and Some New Results," in *Automatica*, **26**(3), Pergamon Press, U.K. pp. 459–474.

[**Frank, P.M. (1992)**], "Robust Model–Based Fault Detedtion in Dynamic Systems," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 1–13.

[**Frank, P.M. and J. Wünnenberg (1989)**], "Robust Fault Diagnosis Using Unknown Observer Schemes," in (R. Patton, P. Frank, R. Clark eds.) *Fault Diagnosis in Dynamic*

*Systems, Theory and Applications*, Prentice Hall, U.K., pp. 47–97.

**[Franksen, O.I., P.Falster, and F.J. Evans (1979)]**, "Qualitative Aspects of Large Scale Systems," in *Lecture Notes in Control and Information Sciences*, **17**, (Balakrishnan & Thoma, Eds.), Springer–Verlag.

**[García, H.E., A. Ray, and R.M. Edwards (1991)]**, "Reconfigurable Control of Power Plants Using Learning Automata," in *IEEE Control Systems*, pp. 85–92.

**[Ge, W., and C.Z. Fang (1988)]**, "Detection of Faulty Components via Robust Observation," in *Intl. Journal of Control*, **47**, pp. 581–599.

**[Gensym Corporation (1990)]**, "G2 Reference Manual for Version 2.0 of the G2 Real–Time Expert System," Gensym Corporation, Cambridge, MA., USA.

**[Genesereth, M.R., (1984)]**, "The Use of Design Descriptions in Automated Diagnosis," in *Artificial Intelligence*, **24**(3), pp. 411–436.

**[Gentner, D. and D.R. Gentner (1983)]**, "Flowing Waters or Teeming Crowds: Mental Models of Electricity," in *Mental Models*, (Gentner & Stevens, Eds.), Lawrence Erlbaum Assoc.

**[Gertler, Janos (1992)]**, "Structured Residuals for Fault Isolation, Disturbance Decoupling and Modeling Error Robustness," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 111–119.

**[Giarratano, J. and G. Riley (1989)]** *Expert Systems: Principles and Programming*, PWs–KENT Publishing Co., Boston, USA.

**[Glasstone, S., and A. Sesonske (1994)]**, *Nuclear Reactor Engineering.*, fourth edition, Vol. 1 & 2 Chapman & Hall, U.S.A.

**[Glymour, C., R. Scheines, P. Spirtes, and K. Kelly (1987)]**, *Discovering Causal Structure*, Academic Press.

**[Godo, L. and R.L. de Mántaras (1993)]**, "Fuzzy Logic," in *Encyclopedia of Computer Science and Technology*, **29**(14), (A. Kent & J.G. Williams, Eds.), Marcel Dekker Inc., New York, USA.

**[Gomm, J.B., D. Williams, and P. Harris (1992)]**, "Detection of Incipient Process Faults Using Approximate Parametric Models," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 138–143.

**[Gorton, P.A. and J.O. Gray (1989)]**, "A FAult Diagnostic Approach to Signal Processing in Non–Destructive Test Instrumentation," in (R. Patton, P. Frank, R. Clark eds.) *Fault Diagnosis in Dynamic Systems, Theory and Applications*, Prentice Hall, U.K., pp. 387–410.

**[Hassberger, J.A., P. Gmytrasiewicz, and J.C. Lee (1990)]**, "Fault Tree Based Diagnostics Using Fuzzy Logic," in *IEEE Trans. on Pattern and Machine Intelligence*, **12**(11), pp. 115–1119.

**[Hayes, P.J. (1979)]**, "The Naive Physics Manifesto," in *Expert Systems in the Micro–Electronic Age,"*, (D. Mitchie, Ed.), Edinburgh University Press, Edinburgh, Scotland, UK., pp. 242–270

[**Hayes, P.J. (1985a)**], "The Second Naive Physics Manifesto," in *Formal Theories of the Commonsense World,* (Hobbs & Moore, Eds.), Ablex Publishing Co., Norwood, N.J., USA, pp. 1–36.

[**Hayes, P.J. (1985b)**], "Naive Physics I: Ontology for Liquids," in *Formal Theories of the Commonsense World,* (Hobbs & Moore, Eds.), Ablex Publishing Co., Norwood, N.J., USA, pp. 71–108.

[**Hayes–Roth, B. (1990)**], "Architectural Foundations for Real–Time Performance in Intelligent Agents," in *Real–Time Systems*, **2**.

[**Hetrick, David. L. (1971)**], *Dynamics of Nuclear Reactors*, The University of Chicago Press, Chicago, U.S.A.

[**Himmelblau, D.M. (1986)**], "Fault Detection and Diagnosis – Today and Tomorrow," in *Proc. of the IFAC Workshop on Fault Detection and Safety in Chemical Plants*, Kyoto, Japan, Oxford Pergamon Press, pp. 95–105.

[**Himmelblau, D.M. (1992)**], "Use of Artificial Neural Networks to Monitor Faults and for Troubleshooting in the Process Industries," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry,* Oxford Pergamon Press, Vol. 8, pp. 144–149.

[**Hobbs and Moore, Eds. (1985)**], *Formal Theories of the Commonsense World,* Ablex Publishing Co., Norwood, N.J., USA.

[**Hopfield, J.J. (1982)**], "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," in *Proceedings of Natl. Acad. Sci.*, **79**, pp. 2554–2558.

[**Hung, J.Y., W. Gao, and J.C. Hung (1993)**], "Variable Structure Control: A Survey," in *IEEE Trans. Industrial Electronics*, **40**(1), pp. 2–22.

[**Itkis, U. (1976)**], *Control Systems of Variable Structure*, Keter Publishing House Jerusalem Ltd., Jerusalem, Israel.

[**Isermann, Rolf (1981)**], "Fault Detection Methods for the Supervision of Technical Processes," in *Process Automation*, **20** Oldenbourg, Munich, pp. 36–44.

[**Isermann, Rolf (1984)**], "Process Fault Detection Based on Modeling and Estimation Methods. A Survey," in *Automatica*, **20**(4) Pergamon Press, U.K., pp. 387–403.

[**Isermann, Rolf (1989)**], "Process Fault Diagnosis Based on Dynamic Models and Parameter Estimation Methods," in (R. Patton, P. Frank, R. Clark eds.) *Fault Diagnosis in Dynamic Systems, Theory and Applications*, Prentice Hall, U.K., pp. 253–291.

[**Iwasaki, Y. and H. Simon (1986)**], "Causality in Device Behavior," in *Artificial Intelligence*, **29**(1).

[**Iwasaki, Y. (1991)**], "Causal Ordering Analysis," in *Qualitative Simulation Modeling and Analysis,* (Fishwick & Luker, Eds.), Springer–Verlag, USA.

[**Jones, H.L. (1973)**], *Failure Detection in Linear Systems*, Ph.D. Thesis, MIT, Cambridge, MA., USA.

**[Jones, J.G. and M.J. Corbin (1989)]**, "Band–Limiting Filter Approach to Fault Detection," in (R. Patton, P. Frank, R. Clark eds.) *Fault Diagnosis in Dynamic Systems, Theory and Applications*, Prentice Hall, U.K., pp. 189–251.

**[Jouse, W.C. (1989)]**, "Self–Teaching Neural Network Learns Difficult Reactor Control Problem," in *Trans. of The American Nuclear Society*, **60**, pp 191.

**[Jouse, W.C. and J.G. Williams (1991)]**, "Neural Networks Controllers: Analogs of Cognitive Structure," *AI'91 – Frontiers in Innovative Computing for the Nuclear Industry*, Jackson, Wyo. USA.

**[Jouse, W.C. (1992)]**, *Self–Training Neural Networks for Risk Reduction in Nuclear Power Operations*, Ph.D. Dissertation, University of Illinois, USA.

**[Kandel, Abraham. (1986)]**, *Fuzzy Mathematical Techniques with Applications*, Addison Wesley, USA.

**[Kandel, Abraham. (Ed.) (1992)]**, *Fuzzy Expert Systems*, CRC Press, USA.

**[Kerr, T.H. (1987)]**, "Decentralized Filtering and Redundancy Management for Multisensor Navigation," in *IEEE Trans. on Aerospace and Electronic Systems*, **23**, pp. 83–119.

**[Khanna, T. (1990)]**, *Foundations of Neural Networks*, Addison–Wesley Publishing Co., USA.

**[Kitamura, M. (1980)]**, "Detection of Sensor Failures in Nuclear Plants Using Analytivc Redundancy," in *Trans. of the American Nuclear Society*, **34**, pp. 581–583.

**[Kitamura, M. (1989)]**, "Fault Detection in Nuclear Reactors with the Aid of Parametric Modeling Methods," in (R. Patton, P. Frank, R. Clark eds.) *Fault Diagnosis in Dynamic Systems, Theory and Applications*, Prentice Hall, U.K., pp. 311–360.

**[Klir, G.J. (1969)]**, *An Approach to General Systems Theory*, Van Nostrand Reinhold, N.Y., USA.

**[Klir, G.J. (1985a)]**, *Architecture of Systems Problem Solving*, Plenum Press, N.Y., USA.

**[Klir, G.J. (1985b)]**, "Reconstructability Analysis: Aims, Results, Open Problems," *Systems Res.*, **2**(2).

**[Klir, G.J and T.A. Folger (1988)]**, *Fuzzy Sets, Uncertainty and Information*, Prentice Hall, Englewood Cliffs, N.J. USA.

**[Klir, G.J. (1989)]**, "Inductive Systems Modelling: An Overview." in *Modelling and Simulation Methodology: Knowledge Systems' Paradigms*, (M. S. Elzas, T. I. Ören, and B. P. Zeigler eds.), Elsevier Science Publishers B.V. (North–Holland), Amsterdam, The Netherlands, pp. 55–75.

**[Klir, G.J. (1991)]**, "Aspects of Uncertainty in Qualitative Systems Modeling," in *Qualitative Simulation Modeling and Analysis*, (Fishwick & Luker, Eds.), Springer–Verlag, USA.

**[Klir, G.J and B. Yuan (1995)]**, *Fuzzy Sets, and Fuzzy Logic. Theory and Applications*, Prentice Hall, Englewood Cliffs, N.J. USA.

**[Kohonen, K. (1984)]** *Self Organization and Associative Memory*, Springer–Verlag, Berlin, Germany.

**[Konstantinov, K.B. and T. Yoshida (1993)]**, "A Method for On–Line Reasoning About the Time–Profiles of Process Variables," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 93–98.

**[Koshijima, I., and K. Niida (1992)]**, "Neural Network Approach to Fault Detection Under Unsteady State Operation," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 174–179.

**[Kosko, B. (1992)]**, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, USA.

**[Krijgsman, A. and R. Jager (1992)]**, "DICE: A Real–Time Toolbox," in *Proc. of the IFAC/IFIP/IMACS Simposyoum on Artificial Intelligence in Real–Time Control*, Delft, The Netherlands.

**[Krijgsman, Ardjan (1993)]**, *Artificial Intelligence in Real–Time Control*, Ph.D. Dissertation, Delft University of Technology, The Netherlands.

**[Krippendorff, K. (1981)]**, "An Algorithm for Identifying Structural Models of Multivariable Data," in *International Journal of General Systems*, **7**(1), pp. 63–79.

**[Kröse B.J.A. and P. Van der Smagt (1991)]**, *An Introduction to Neural Networks*, Faculty of Mathematics and Computer Science, University of Amsterdam.

**[Kuipers, B. (1982)]**, "Getting the Envisionment Right," in *Proc. of the National Conference on Artificial Intelligence (AAAI-82)*, Aug. 18–20.

**[Kuipers, B. (1984)]**, "Commonsense Reasoning About Causality: Deriving Behavior From Structure," in *Artificial Intelligence*, **24**.

**[Kuipers, B. (1986)]**, "Qualitative Simulation," in *Artificial Intelligence*, **29**(3), pp. 289–338.

**[Kuipers, B. (1989a)]**, "Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge," in *Automatica*, **25**(4), Pergamon Press, UK.

**[Kuipers, B. (1989b)]**, "Qualitative Reasoning with Causal Models in Diagnosis of Complex Systems," in *Artificial Intelligence, Simulation and Modeling*, (L.A. Widman, K.A. Loparo, and N. Nielsen, Eds.), John Wiley & Sons, New York, pp. 257–274.

**[Kumamaru, K., S. Sagara, and H. Yanagida (1984)]**, *Fault Detection of Dynamical Based on a Model Discriminating Approach*, Report UPTEC 84123R, Institute of Technology, Uppsala University, Uppsala, Sweden.

**[Kumamaru, K., S. Sagara, and T. Söderström (1989)]**, "Some Statistical Methods for FAult Diagnosis for Dynamical Systems," in (R. Patton, P. Frank, R. Clark eds.) *Fault Diagnosis in Dynamic Systems, Theory and Applications*, Prentice Hall, U.K., pp. 439–475.

**[Labarrère, M. (1987)]**, "Aircraft Sensor Failure Detection by Analytic Redundancy," in (M.G. Singh ed.) *Systems and Control Encyclopedia*, Vol. 1, Pergamon Press, Oxford, pp. 246–251.

[Laffey, T.J, W.A. Perkins, and T.A. Nguyen (1986)] "Reasoning About Fault Diagnosis with LES," in *IEEE Expert*, pp. 13–20.

[Laffey, T.J, P.A. Cox, J.L. Schmidt, S.M. Kao, and J.Y. Read (1988)], "Real–Time Knowledge–Based Systems," in *Artificial Intelligence Magazine*, 6(2) pp. 115–139.

[Lamarsh, John R. (1983)], *Introduction to Nuclear Reactor Engineering*, Addison–Wesley, Massachusetts, U.S.A.

[Larsson, J.E. (1991)], "Model–Based Fault Diagnosis Using Multilevel Flow Modeling," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 331–336.

[Law A., and D. Kelton (1990)], *Simulation Modeling and Analysis*, 2nd Edition, McGraw–Hill, New York, 1990.

[Lee, In–Beum., M. Kim, J. Jung, and K.S. Chang (1992)], "Rule–Based Expert System for Diagnosis of Energy Distribution," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 239–243.

[Leonard, J.A. and M.A. Kramer (1992)], "A Decomposition Approach to Solving Large–Scale Fault Diagnosis Problems with Modulkar Neural Networks," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 180–185.

[Li, D. and F.E. Cellier (1990)], "Fuzzy Measures in Inductive Reasoning," in *Proceedings of the 1990 Winter Simulation Conference*, New Orleans, LA, pp. 527–538.

[Linford, R.B. (1973)], *Analytical Methods of Plant Transient Evaluations for the G.E. BWR*, General Electric, NEDO 1082.

[Lippman, P. (1987)], "An Introduction to Neural Networks," in *IEEE ASSP Magazine*, 3(4), pp. 4–22.

[Ljung, Lennart (1987)], *System Identification: Theory for the User*, Prentice Hall Information and System Sciences Series, USA.

[López de Mántaras, R. (1986)], "Técnicas de Representación del Razonamiento Aproximado," in *Inteligencia Artificial: Sistemas Expertos*, Alianza Editorial, España, pp. 69–110.

[López de Mántaras, R. (1989)], "Sistemas Expertos, Limitaciones y Perspectivas," in *Mundo Electrónico*, No. 200, España, pp. 237–243.

[López de Mántaras, R. (1990)], *Approximate Reasoning Models*, Ellis Horwood Artificial Intelligence Series, USA.

[López de Mántaras, R. (1991)], "A Distance–Based Attribute Selection Measure for Decision Tree Induction," in *Machine Learning*, 6(1).

[López de Mántaras, R. and L. Godó (1993)], "From Fuzzy Logic to Fuzzy Truth–Valued Logic: A Survey," in *Proc. of the IEEE Intl. Conference on Fuzzy Systems*, San Francisco, CA, USA, IEEE Press, pp. 750–755.

[Lou, X.C., A.S. Willsky, and G.L. Verghese (1986)], "Optimally Robust Redundancy Relations for Failure Detection in Uncertain Systems," in *Automatica*, 22, pp. 333–344.

[Luo, R.C. and M.G. Kay (1989)], "Multisensor Integration and Fusion in Intelligent Systems," in *IEEE Trans. on Systems, Man and Cybernetics*, 19(5).

[Lund, E.J., J.G. Balchen and B.A. Foss (1992)], "Monitoring Processes with Structurally Different Operating Regimes," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 126–131.

[Maiers, J. and Y.S. Sherif (1985)], "Applications of Fuzzy Set Theory," in *IEEE Transactions on Systems, Man and Cybernetics*, SMC-15, pp. 175–186.

[Mathworks Inc. (1992a)], *MATLAB: High Performance Numeric Computation and Visualization Software — Reference Guide*, Natick, Mass., USA.

[Mathworks Inc. (1992b)] *The Student Edition of MATLAB for MS-DOS or Macintosh Computers*, Prentice-Hall, Englewood Cliffs, N.J., USA.

[Massoumnia, M.A. (1986)], *A Geometric Approach to Failure Detection and Identification in Linear Systems*, Ph.D. Thesis, MIT, Cambridge, MA., USA.

[Mehra, R.K. and I. Peshon (1971)], "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems," in *Automatica*, 7, pp. 637–640.

[Meserole, J.S. (1981)], *Detection Filters for Fault Tolerant Control of Turbofan Engines*, Ph.D. Thesis, MIT, Cambridge, MA., USA.

[MGA. Mitchell & Gauthier Assoc. (1986)], *ACSL: Advanced Continuous Simulation Language — User Guide and Reference Manual*, Concord, Mass., USA.

[MGA, Mitchel & Gauthier Assoc. (1991)] *ACSL: Advanced Continuous Simulation Language — Reference Manual*, Edition 10.0, Concord, Mass., USA.

[Miller W.T., R.S. Sutton, and P.J. Werbos (Eds.), (1990)], *Neural Networks for Control*, The MIT Press, Cambridge, U.K.

[Minsky, M., and S. Papert (1969)], *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, MA, USA.

[Mizumoto, M. and H. Zimmermann, (1982)], "Comparison of Fuzzy Reasoning Methods," in *Fuzzy Sets and Systems*, 8, pp. 253–283.

[Montgomery, R.C. and A.K. Caglayan (1974)], "A self–Reorganizing Digital Flifght Control System for Aircraft," in *Proc. of the AIAA 12th Aerospace Sciences Meeting, Washington D.C., USA.*

[Montgomery, R.C. and J.P. Williams (1989)], *"Analytic Redundancy Management for Systems with Appreciable Structural Dynamics," in (R. Patton, P. Frank, R. Clark eds.)* Fault Diagnosis in Dynamic Systems, Theory and Applications, *Prentice Hall, U.K., pp. 361–386.*

**[Motaabbed, A. (1992)]**, A Knowledge Acquisition Scheme for Fault Diagnosis in Complex Manufacturing Processes,, *M.S. Thesis, (F.E. Cellier, Adv.), Dept. of Electrical & Computer Engineering, University of Arizona, Tucson, AZ., USA.*

**[Mugica, F. (1995)]** Diseño Sistemático de Controladores Difusos Usando Razonamiento Inductivo, *Ph.D. dissertation (F.E. Cellier and R.M. Huber adv.), Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Spain.*

**[Mugica, F. and F.E. Cellier (1993)]**, *"A New Fuzzy Inferencing Method for Inductive Reasoning," in* Proc. of the Sixth Intl. Symposium on Artificial Intelligence, and Intelligent Systems in Industry and Business, *Monterrey, México, pp. 372–379.*

**[Mugica, F., and F.E. Cellier (1994)]**, *"Automated Synthesis of a Fuzzy Controller for a Cargo Ship Steering by Means of Qualitative Simulation," in* Proc. of the SCS European Simulation Multiconference ESM'94, *Barcelona, Spain, pp. 523–528.*

**[Mugica, F., and F.E. Cellier (1995)]**, *"Inductive Reasoning Supports the Design of Fuzzy Controllers," in* Journal of Intelligent and Fuzzy Systems, **3***(1), John Wiley & Sons, pp. 523–528.*

**[Nebot, A. (1994)]** Qualitative Modeling and Simulation of Biomedical Systems Using Fuzzy Inductive Reasoning, *Ph.D. dissertation (F.E. Cellier and R.M. Huber adv.), Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Spain.*

**[Nebot, A., F.E. Cellier, and D.D. Linkens (1993)]**, *"Controlling an Anaesthetic Agent by Means of Fuzzy Inductive Reasoning," in* Proc. IMACS Workshop on Qualitative Reasoning and Decision Technologies QUARDET'93, *Barcelona, Spain, pp. 345–356.*

**[Nebot, A., S. Medina, and F.E. Cellier (1994)]**, *"The Causality Horizon: Limitations to Predictability of Behavior Using Fuzzy Inductive Reasoning," in* Proc. of the ESM–ICQFN'94, *Barcelona, Spain, pp. 492–496.*

**[Nebot, A., and F.E. Cellier (1994a)]**, *"Preconditioning of Measurement Data for the Elimination of Patient–Specific Behavior in Qualitative Modeling of Medical Systems," in* Proc. of CISS'94 First Joint Conference of International Simulation Societies, *Zurich, Switzerland, August 22–25, pp. 584–588.*

**[Nebot, A., and F.E. Cellier (1994b)]**, *"Dealing with Incomplete Data Records in Qualitative Modeling and Simulation of Biomedical Systems," in* Proc. of CISS'94 First Joint Conference of International Simulation Societies, *Zurich, Switzerland, August 22–25, pp. 605–610.*

**[Newell, A. and H. Simon (1972)]**, Human Problem Solving, *Prentice Hall, USA.*

**[NNguyen, D., and B. Widrow (1989)]**, *"The Truck Backer–Upper: An Example of Self–Learning in Neural Networks," in* Proc. of the Intl. Joint Connf. on Neural Networks IJCNN'89, *vol. IIpp. 357–363.*

**[Niccoli, L.G., N.P. Wilburn, R.W. Colley, F. Alexandro, R.N. Clark, and S. Bouzerdoum (1985)]**, *"Detection of Instrument or Component Failures in a Nuclear Power Plant by Luenberger Observers," in* Proc. Of the Intl. Topical Meeting on Computer Applications for Nuclear Plant Operation and Control, *Pasco, WA. USA.*

**[NSAC–EPRI (1980)]**, *"Fundamental Safety Parameter Set for Boiling Water Reactors,"* *in* Report NSAC–21, TSA 80–367, *Nuclear Safety Analysis Center, Electric Power Research Institute, California, U.S.A.*

**[Onken, R. and N. Stuckenberg (1979)]**, *"Failure Detection in Signal Processing and Sensing in Flight Control Systems," in* Proc. of the IEEE Conference on Decision and Control, *San Diego, CA., pp 449–454.*

**[Ören, T.I., B.P. Zeigler, M.S. Elzas, Eds.(1982)]**, Simulation and Model–Based Methodologies: An Integrated View, *NATO ASI Series F: Computer and System Sciences, Vol. 10, Springer–Verlag, Hiedelberg, Berlin, Germany.*

**[Pan, J. (1986)]**, *"Qualitative Reasoning with Deep–Level Mechanism Models for Diagnose of Failures," in* Proc. of the CAIA'86, *Denver, Co., USA, pp. 295–301.*

**[Patil, R., P. Szolovits, and W. Schwartz (1981)]**, *"Causal Understanding of Patient Illness in Medical Diagnosis," in* Proc. of Intl. Joint Conf. on A.I. IJCAI'81, *Vancouver, B.C. Canada, pp. 893–899.*

**[Patton, R.J. and S.M. Kangethe (1989)]**, *"Robust Fault Diagnosis Using Eigenstructure Assignement of Observers," in (R. Patton, P. Frank, R. Clark eds.)* Fault Diagnosis in Dynamic Systems, Theory and Applications, *Prentice Hall, U.K., pp. 99–154.*

**[Patton, J.R., S.W. Willcox, and J.S. Winter (1987)]**, *"A Parameter Insensitive Technique for Aircraft Sensor Fault Analysis," in* AIAA Joint Conf. on Guidance, Control and Dynamics, *pp. 359–367.*

**[Patton, J.R., P. Frank, and J. Clark (1989)]**, Fault Diagnosis in Dynamic Systems. Theory and Applications, *Prentice Hall, U.K.*

**[Patton, R.J., J. Chen, and J.H.P. Millar (1992)]**, *"Robust Fault Detection for a Nuclear Reactor System: A Feasibility Study," in* Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry, *Oxford Pergamon Press, Vol. 8, pp. 120–125.*

**[Pau, L.F. (1973)]**, Diagnostc Statistique, *2nd. ed., ENS de l'Aéronautique, Toulousse, 83 pp.*

**[Pau, L.F. (1974)]**, *"Diagnosis of Equipment Failures by Pattern Recognition," in* IEEE Transaction on Reliability, **23***(3), pp. 202–208.*

**[Pau, L.F. (1981)]**, Failure Diagnosis and Performance Monitoring, *Marcel Dekker, New York and Basel.*

**[Pau, L. (1989)]**, *"Knowledge Representation Approaches in Sensor Fusion," in* Automatica, **25***(2).*

**[Pedrycz, W. (1993)]**, Fuzzy Control and Fuzzy Systems, *Second Extended Edition, Research Studies Press Ltd., Jhon Wiley & Sons Inc., Taunton, Somerset, UK.*

**[Perkins, W.A., and A. Austin (1990)]**, *"Adding Temporal Reasoning to Expert–System–Building–Environments," in* IEEE Expert, *February.*

**[Plaza, E. (1992)]**, *"Tendencias en Inteligencia Artificial: Hacia la Cuarta Década," in* Nuevas Tendencias en Inteligencia Artificial, *Universidad de Deusto, España, pp. 379–415.*

**[Puccia, C.J. and R. Levins, (1985)]**, Qualitative Modeling of Complex Systems, *Harvard University Press. USA.*

**[Quian, D.Q. (1992)]**, *"Representation and Use of Imprecise Temporal Knowledge in Dynamic Systems," in* Fuzzy Sets and Systems, **50**, *pp. 59–77.*

**[Ramos, J.C. and A. de Albornoz (1988)]**, *"A One Dimensional Model for the Laguna Verde Boiling Water Reactor," in* Proc. of the 1988 SCS Conference on Power Plant Simulation, *San Diego, CA., USA.*

**[Ramos, J.C. (1991)]**, Simulación en Tiempo Real de un Reactor Nuclear de Agua Hirviente, *Tesis de Maestría, Esc. Sup. de Física y Matemáticas, Inst. Politécnico Nacional, México D.F.*

**[Rasmussen, Jens (1985)]**, *"The Role of Hierarchical Knowledge Representation in Decisionmaking and System Management," in* IEEE Trans. on Systems, Man, and Cybernetics, *SMC* **15***(2), pp. 234–243*

**[REAKT (1990)]**, *"REAKT: Environment and Methodology for Real–Time Knowledge–Based Systems," ESPRIT II 4651 Report, Universitat Politècnica de Valencia, Spain.*

**[Rengaswamy, R. and V. Venkatasubramanian (1992)]**, *"An Integrated Framework for Process Monitoring, Diagnosis, and Control Using Knowledge–Based Systems and Neural Networks," in* Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry, *Oxford Pergamon Press, Vol. 8, pp. 325–330.*

**[Rich, E. and Knight, K. (1991)]**, Artificial Intelligence, *Second Edition, Mc Graw Hill Inc., USA.*

**[Rieger, C. and M. Grinberg (1977)]**, *"The Declarative Representation and Procedural Simulation of Causality in Physical Mechanisms," in* Proc. of the Fifth Intl. Joint Conference on A.I. (IJCAI-5), *Cambridge, MA, USA.*

**[Rojas–Guzman, C. and M.A. Kramer (1992)]**, *"Belief Networks for Knowledge Integration and Abductive Inference in Fault Diagnosis of Chemical Processes," in* Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry, *Oxford Pergamon Press, Vol. 8, pp. 14–19.*

**[Rotstein, G.E. and D.R. Lewin (1992)]**, *"Automatic Synthesis of Chemical Batch Plant Procedures. A Qualitative Process–Centered Approach," in* Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry, *Oxford Pergamon Press, Vol. 8, pp. 57–62.*

**[Sarjoughian, H. (1995)]**, An Approach to Inductive Modeling Based On Systems Theory and Non–monotonic Reasoning, *Ph.D. dissertation, University of Arizona, USA.*

**[Scarl, E.A., J.R. Jamieson, and C.I. Delaune (1985)]**, *"A Fault Detection and Isolation Method Applied to Liquid Oxygen Loading for the Space Shuttle," in* Proc. of Intl. Joint Conf. on I.A. IJCAI'85, *Los Angeles, CA, USA, pp. 414–416.*

**[Scarl, E.A., J.R. Jamieson, and E. New (1988)]**, *"Model–Based Reasoning for Diagnosis and Control," in* Proc. of the First Florida Artificial Intelligence Research Symposium (FLAIRS-88), *Orlando, FL, USA. Also presented at the* IFAC Workshop on Spacecraft Autonomy, *Pasadena, CA, USA.*

**[Scarl, E.A. (1991a)]**, *"Monitoring and Diagnosis: Stress from Weakened Environmental*

*Knowledge,"* Proc. of IEEE International Conference on Robotics and Automation, *Sacramento, CA, USA.*

**[Scarl, E.A. (1991b)]**, *"Iteration Against Default States for Qualitative Monitoring and Diagnosis," in* Fifth Intl. Workshop on Qualitative Reasoning (QR-91), *Austin, TX, USA. Selected for republication in* IEEE Expert.

**[Scarl, E.A. (1993)]**, *"Hierarchical Model Abstraction for Diagnosis," in* AIAA Computing in Aerospace, **9**, *San Diego, CA, USA.*

**[Scheines, R. P. Spirtes, and C. Glymour (1991)]**, *"A Qualitative Approach to Causal Modeling," in* Qualitative Simulation Modeling and Analysis, *(Fishwick & Luker, Eds.), Springer–Verlag, USA.*

**[Schenker, B. and M. Agarwal (1992)]**, *"Feedback Neural Nets for Supervision of Dynamic Processes," in* Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry, *Oxford Pergamon Press, Vol. 8, pp. 168–173.*

**[SCT, Systems Control Technology, (1985)]**, CTRL–C: A Language for the Computer-Aided Design of Multivariable Control Systems, User's Guide, *Palo Alto, CA, USA.*

**[Shafer, G. (1976)]**, A Mathematical Theory of Evidence, *Princeton University Press, Princeton, N.J., USA.*

**[Shapiro, E.Y. and H.E. Decarli (1979)]**, *"Analytic Redundancy for Flight Control Sensors on the Lockheed L–1011 Aircraft," in* Proc. 18th IEEE CDC Conference, San Diego, CA., pp. 131–138.

**[Shortfile, E., (1976)]**, *Computer–Based Medical Consultations: MYCIN*, Elsevier.

**[Sierra, C. and R. Sanguesa (1992)]**, "Herramientas de Desarrollo de Sistemas Expertos," in *Nuevas Tendencias en Inteligencia Artificial*, Universidad de Deusto, España, pp. 111–128.

**[Sira–Ramírez H. (1990)]**, "Dynamical Discontinuous Feedback Control of Nonlinear Systems," in *IEEE Trans. Automatic Control*, **35**(12), pp. 1373–1378.

**[Slotine, J.J.E. (1984)]**, "Sliding Controller Design for Nonlinear Systems," in *International Journal of Control*, **40**(2), pp. 421–434.

**[Stevens, L.S. and F.L. Lewis, (1992)]** Lewis, *Aircraft Control and Simulation*, John Wiley & Sons, USA.

**[Strömberg, J.E., U. Söderman, and J. Top (1993)]**, "Bond Graph Supported Estimation of Discretely Changing Parameters," in *Proc. ICBGM'93, Intl. Conference on Bond Graph Modeling*, San Diego, Calif., January 17–20, pp. 47–52.

**[Sudkamp, T. and R. Hammell (1994)]**, "Interpolation, Completion and Learning Fuzzy Rules," in *IEEE Trans. Systems, Man, Cybernetics*, **24**(2), pp. 332–342.

**[Surgenor, B.W. and P.J. Jofriet (1992)]**, "Thermal Fault Analysis and the Diagnostic Model Processor," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 20–25.

**[Tanaka, Shogo (1989)]**, "Diagnosability of Systems and Optimal Sensor Location," in (R. Patton, P. Frank, R. Clark eds.) *Fault Diagnosis in Dynamic Systems, Theory and*

*Applications*, Prentice Hall, U.K., pp. 155–188.

**[Terpstra, P.J., H.B. Verbruggen, M.W. Hoogland and R.A.E. Ficke (1992)]**, "A Real–Time, Fuzzy Deep–Knowledge Based Fault Diagnosis System for a Continuous Stirred Tank Reactor," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 26–31.

**[Travé–Massuyes, L., A. Missier, and N. Piera (1990)]**, "Qualitative Models for Automatic Control Process Supervision," in *11th IFAC World Congress*, Tallin, Lithuania, pp. 198–203.

**[Tzafestas, S.G., M. Singh, and G. Schmidt (1987)]**, *System Fault Diagnostics. Reliability and Related Knowledge–Based Approaches.* Vol. 1 & 2, Reidel Press, U.S.A.

**[Tzafestas, Spyros G. (1989)]**, "System Fault Diagnosis Using Knoeledge–Based Methodology," in (R. Patton, P. Frank, R. Clark eds.) *Fault Diagnosis in Dynamic Systems, Theory and Applications*, Prentice Hall, U.K., pp. 509–572.

**[Tzafestas, Spyros G., (1993)]**, *Knowledge–Based System Diagnosis, Supervision, and Control.* Plenum Press, New York, London.

**[Utkin, V.A. (1977)]**, "Variable Structure Systems Within Sliding Mode," in *IEEE Trans. Automatic Control*, **22**(2), pp. 212–222.

**[Utkin, V.A. (1984)]**, "Variable Structure Systems: Present and Future," in *Automation and Remote Control*, **44**(9), pp. 1105–1120.

**[Utkin, V.A. (1992)]**, *Sliding Modes in Control Optimization*, Springer–Verlag, Heidelberg, Germany.

**[Uyttenhove, H.J. (1978)]**, *Computer–Aided Systems Modeling: An Assemblage of Methodological Tools for Systems Problem Solving*, Ph.D. dissertation, (G.J. Klir, Adv.), School of Advancd Technology, University of New York, SUNY–Binghamton, USA.

**[Uyttenhove, H.J. (1981)]**, *SAPS (Systems Approach Problem Solver): An Introduction and Guide*, Computing and System Consultants, Binghampton, New York, USA.

**[VAchkov, Gancho (1992)]**, "Identification of Fuzzy Rule–Based System for Fault Diagnosis in Chemical Plants," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 315–318.

**[Vesanterä, P.J. and F.E. Cellier (1989)]**, "Building Intelligence into an Autopilot: Using Qualitative Simulation to Support Global Decision Making," in *Simulation*, **52**(3), pp. 111–121.

**[Vesanterä, P.J. (1988)]**, *Qualitative Flight Simulation: A Tool for Global Decision Making*, M.S. Thesis, (F.E. Cellier, Adv.), Dept. of Electrical and Computer Engineering, Univ. of Arizona, Tucson, AZ, USA.

**[Vinson, J.M. and L.H. Ungar (1992)]**, "Fault Detection and Diagnosis using Qualitative Modeling and Interpretation," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 81–86.

**[Vinson, J.M., S.D. Grantham, and L.H. Ungar (1992)]**, "Automatic Rebuilding of

Qualitative Models for Diagnosis," in *IEEE Expert*, **7**(4), pp. 23–30.

**[Walker, Bruce K. (1989)]**, "Fault Detection Threshold Determination Using Markov Theory," in (R. Patton, P. Frank, R. Clark eds.) *Fault Diagnosis in Dynamic Systems, Theory and Applications*, Prentice Hall, U.K., pp. 477–508.

**[Wang, Q. (1991)]**, "The Development of a Diagnostic Expert System for Equipment Maintenance," in *Proceedings of the AMD Technical Conference*, San Francisco, CA, USA.

**[Wang, Q. and F.E. Cellier (1991)]**, "Time Windows Automated Abstraction of Continuous-Time Models into Discrete Events Models in High Autonomy Systems," in *International Journal of General Systems*, **19**(3), pp. 241–262.

**[Watanabe, K. and D.M. Himmelblau (1982)]**, "Instrument Fault Detection in Systems with Uncertainties," in *Intl. Journal of Systems Science*, **13**, pp. 137–158.

**[Waterman, D.A. (1986)]**, *A Guide to Expert Systems*, Addison–Wesley, USA.

**[Weiss, J.L., K.R. Pattipati, A.S. Willsky, J.S. Eterno, and J.T. Crawford (1985)]**, *Robust Detection/Isolation/accomodation for Sensor Failures*, NASA Contractor Report CR–17479, Alphatec Inc. Burlington , MA., USA.

**[Weld, D.S. and J. de Kleer, Eds. (1990)]**, *Readings in Qualitative Reasoning About Physical Systems*, Morgan Kaufman, San Mateo, CA, USA.

**[Wells, Gordon (1992)]**, "An Introduction to Neural Networks," in *Application of Artificial Intelligence in Process Control*, (L.Boullart, A. Krijgsman, and R.A. Vingerhoeds Eds.), Pergamon Press, UK.

**[Wells, W.R and C.W. de Silva (1977)]**, "Failure State Detection of Aircraft Engine Output Sensor," in *Proc. of the Joint Automatic Control Conference* San Francisco, CA., Vol. 2, pp. 1493–1497.

**[Whiteley, J.R. and J.F. Davis (1992)]**, "Qualitative Interpretation of Sensor Patterns Using a Similarity–Based Approach," in *Proceed. of the IFAC Symposium om On–Line Fault Detection and Supervision in the Chemical Process Industry*, Oxford Pergamon Press, Vol. 8, pp. 75–80.

**[Wilbers, D.N. and J.L. Speyer (1989)]**, "Detection Filters for Aircraft Sensor and Actuator Faults," in *Proc. of the ICCON'89 Intl. Conf. on Control and Applications*, Jerusalem, Israel, pp. 126–135.

**[Williams, M.D., J.D. Holland, and A.L. Stevens (1983)]**, "Human Reasoning About a Simple Physical System," in *Mental Models*, (Gentner & Stevens, Eds.), Lawrence Erlbaum Assoc.

**[Willsky, A.S. (1976 ]**, "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica*, **12**, pp. 601–611.

**[Willsky, A.S. (1980)]**, *Failure Detection in Dynamic Systems*, AGARD No. 109.

**[Wünnemberg, J. and P.M. Frank (1987)]**, "Sensor Fault Detection via Robust Observers," in *Proc. of the 1st. European Workshop on Fault Diagnostics, Reliability and Related Knowledge–Based Approaches*, Rhodes, Greece, Vol. 1, pp. 147–160.

**[Zadeh, L.A. (1965)]**, "Fuzzy Sets," in *Inf. Control*, **8**(3).

**[Zadeh, L.A. (1978)]**, "Fuzzy Sets as a Basis for a Theory of Possibility," in *Fuzzy Sets and Systems*, **1**(1).

**[Zeigler, B.P. (1986)]**, "Review of Architecture of Systems Problem Solving," in *International Journal of General Systems*, **13**(1).

**[Zeigler, B.P (1990)]**, "Endomorphic Modeling Concepts for High–Autonomy Systems Architectures," in *Applied Artificial Intelligence*, **6**, pp. 19–43.

**[Zhu, Yucai. and T. Backx (1993)]**, *Identification of Multivariable Industrial Processes for Simulation, Diagnosis, and Control.* Springer–Verlag, Berlin, New York.

**[Zwingelstein, G.C. and B.R. Upadhyaya (1979)]**, "Identification of Multivariate Models for Noise Analysis of Nuclear Plants," in *Proc. of the 5th IFAC Symposium Identification and System Parameter Estimation*, Darmstadt, Pergamon Press, pp. 175–181.