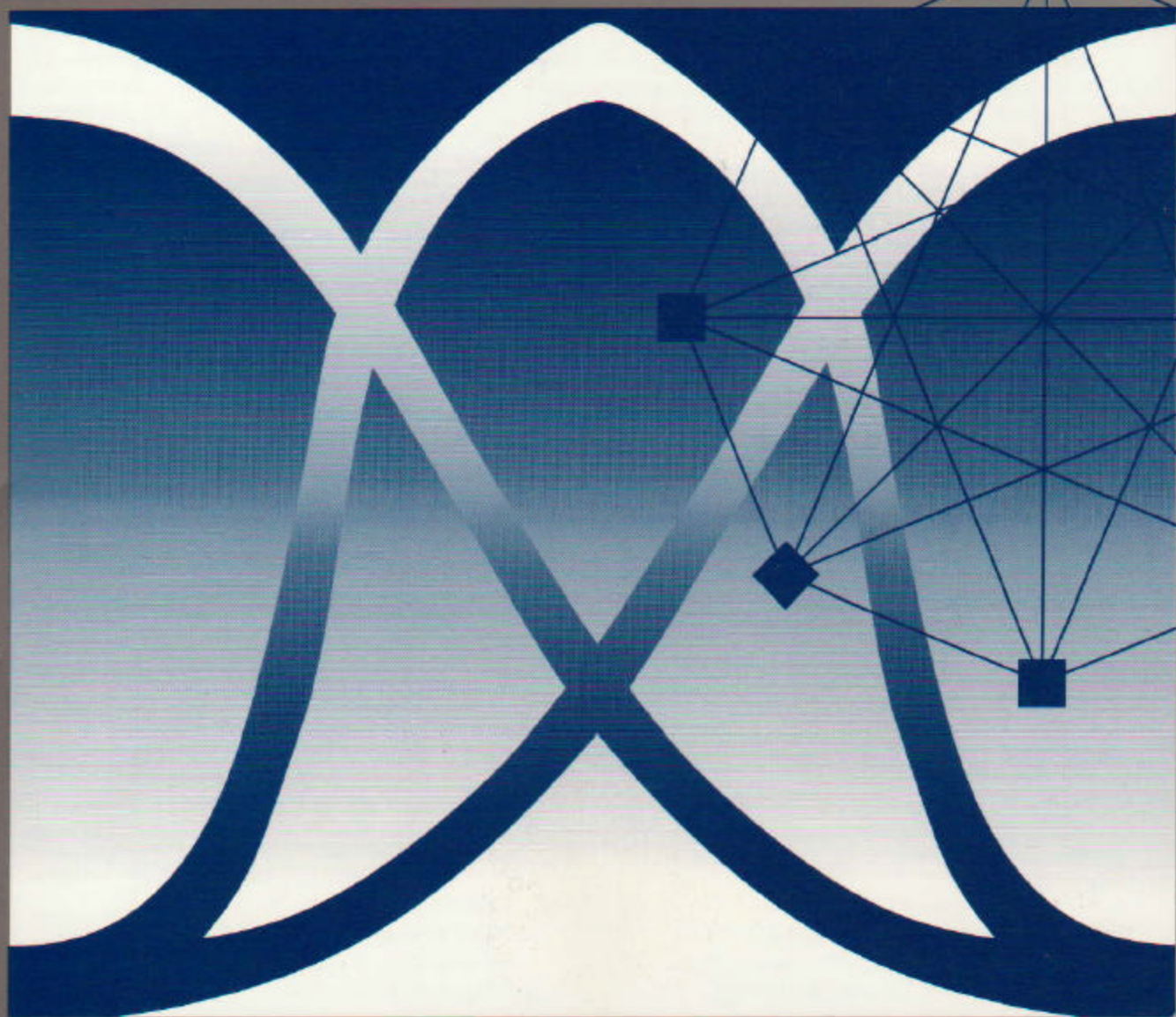


VOLUME 3 ISSUE 1 1995

# JOURNAL OF INTELLIGENT & FUZZY SYSTEMS

APPLICATIONS IN ENGINEERING  
AND TECHNOLOGY



 A Wiley-Interscience Publication  
**John Wiley & Sons, Inc.**  
New York • Chichester • Brisbane • Toronto • Singapore

SPECIAL ISSUE ON  
FUZZY LOGIC AND NEURAL NETWORKS:  
THEORY AND APPLICATIONS

MOHAMMAD JAMSHIDI & TIMOTHY J. ROSS, EDITORS

# JOURNAL OF INTELLIGENT & FUZZY SYSTEMS

---

## FUZZY LOGIC AND NEURAL NETWORKS: THEORY AND APPLICATIONS *A SPECIAL ISSUE*

### Guest Editorial

- Fei-Yue Wang, Z. Jason Geng, and Mo Jamshidi* 1
- Representation and Analysis of Uncertainty Using Fuzzy Petri Nets**  
*Tiehua Cao and Arthur C. Sanderson* 3
- Hierarchical Fuzzy Control of Robotic Manipulators**  
*Kishan Kumar Kumbla and Mo Jamshidi* 21
- Autonomous Robotic Mining Excavation Using Fuzzy Logic and Neural Networks**  
*Paul J. A. Lever, Fei-Yue Wang, Deqian D. Chen, and Xiaobo Shi* 31
- Design and Implementation of Fuzzy Logic Controllers for a Heat Exchanger in a Water-for-Injection System**  
*Constantine N. Manikopoulos, MengChu Zhou, and Sujit S. Nerurkar* 43
- Derivation of Fuzzy Rules for Model-Free Tuning of PID Controllers**  
*John S. Baras and Nital S. Patel* 59
- Inductive Reasoning Supports the Design of Fuzzy Controllers**  
*François E. Cellier and Francisco Mugica* 71
- Fuzzy CMAC Neural Networks**  
*Z. Jason Geng* 87
- Announcements** 103
- Diskette Submission Instructions** 1

**François E. Cellier**  
Department of Electrical and  
Computer Engineering  
The University of Arizona  
Tucson, AZ 85721

**Francisco Mugica**  
Departament d'ESAI  
Universitat Politècnica de Catalunya  
Diagonal 647, 2da. planta  
Barcelona 08028, Spain

## INDUCTIVE REASONING SUPPORTS THE DESIGN OF FUZZY CONTROLLERS

### ABSTRACT

*In this article, a new systematic design methodology for fuzzy controllers is presented. For any desired plant output, it is possible to find off-line the optimal plant input that will produce a plant output that is as close as possible to the desired one. However, this constitutes an open-loop design. In this article, a new methodology is introduced that allows computing a signal on-line that is close to the optimal plant input as a function of system inputs and plant outputs. To this end, an inductive reasoning model is created that estimates the optimal plant input from given system inputs and plant outputs. The inductive reasoning model can be interpreted and realized as a fuzzy controller. Thereby, a large portion of the controller is realized through feedback, and the previous open-loop design is converted to an equivalent and more robust closed-loop design. The inductive reasoning technique is described in detail in the first part of this article. An example is shown in the second part of the article to demonstrate the validity of the chosen approach. © 1995 John Wiley and Sons, Inc.*

### INTRODUCTION

Fuzzy controllers have become quite popular over the past years, particularly in Japan. At least four reasons can be mentioned that make fuzzy controllers potentially attractive:

1. *Price:* A fuzzy controller can be realized cheaply. Special chips have been designed that can be used to implement fuzzy controllers for a large variety of different industrial processes.
2. *Flexibility:* A fuzzy controller can be designed with very little knowledge of the plant it is supposed to control. Conse-

quently, the same fuzzy controller can, by adjusting its performance parameters, be used to control different types of processes. Only the classical PID controller can compete with the fuzzy controller in flexibility.

3. *Robustness:* Contrary to the optimal state feedback controller, which is very sensitive to parameter variations, a fuzzy controller can deal much more reliably with a plant whose parameters are time varying. While a human aircraft pilot is unable to compute an optimal flight path in his or her head by solving a matrix Riccati equation, he or she is able to control the aircraft successfully and reliably in situations where any one of today's autopilots would fail miserably. When an anomaly has occurred, the first thing that the human pilot will do is switch off the autopilot. However, under normal circumstances, the autopilot can fly the aircraft more softly and more economically (in terms of kerosene consumption) than a human pilot could do. In some sense, optimality can be traded for robustness. The same holds true for fuzzy controllers. A fuzzy controller can never compete with an optimal controller in terms of efficiency, but it can be built to be considerably more robust than any optimal controller.
4. *Adaptability:* Because a fuzzy controller requires less knowledge of the environment it operates in, it is easier to make it adapt itself to a changing situation than any optimal controller.

Except for point 1 above, which is purely econ-

omically founded, all the other points are closely related to one another. They all deal with questions of efficiency versus flexibility, of specialization versus generality.

Obviously, a fuzzy controller is not a cure for all diseases. Although fuzzy controllers are potentially more adaptable than state feedback controllers, this does not mean that a fuzzy controller will automatically adapt itself to arbitrarily changing operating conditions. While fuzzy controllers are usually fairly robust when dealing with nonlinear plant behavior or unmodeled plant dynamics (much more so than state feedback controllers), they are known to be rather sensitive to changes in the operating point. Moreover, fuzzy controllers are still more of an art than of a science. In particular, there is no general technique currently available that would allow the design of a fuzzy controller that is guaranteed to be stable under all feasible operating conditions. This is in contrast to the state feedback controllers, whose stability properties can be analytically determined, at least when applied to linear systems. Techniques have also been developed to analyze the stability properties of some types of controllers when applied to some classes of nonlinear plants. It is this lack of mathematical capabilities to assess the stability and convergence properties of fuzzy controllers that makes many good classical control engineers skeptical of the potentials of this technology.

However, it is uncontested that fuzzy controllers have indeed been successfully applied to many practical problems that were previously either not solvable at all or solvable only with considerably more expensive controllers, and it is this success story that makes the technology attractive to the more practically oriented vintage among the control engineers.

If humans are ever going to colonize other planets, they will have to rely on an army of fairly autonomously operating robots that will be needed to prepare these other planets for human arrival. These robots are not manufacturing robots. It is not essential that they produce as much merchandise per time unit as possible. It is much more important that they are *robust*, i.e., can operate on their own without running into any sort of trouble, that they are *adaptive*, i.e., can reliably accommodate to a changing environment, and that they are *flexible*, i.e., can be used for various different tasks. Fuzzy control may

provide a partial answer to some of these demands.

Fuzzy controllers are essentially rule-based controllers whereby continuous variables are discretized (recoded) into classes. A recoded fuzzy variable preserves its quantitative information in a fuzzy membership function that it carries along with its class value. Operations performed on fuzzy variables are performed separately on their class values (using finite state automata techniques, so-called "rules") and on their fuzzy membership functions (using fuzzy logic).

The design of any controller is an optimization problem. It may be that the optimization problem can be solved analytically (such as in the case of state feedback controllers for linear plants), and in this case, the optimization problem may no longer be directly visible to the user. However, when a controller is to be built for an arbitrarily nonlinear and/or partially unknown plant, the tuning of the controller becomes a formidable task. The more complex the controller, the more tuning parameters it will offer, and the more difficult it will be to find the optimal setting of these parameters. In general, the control engineer must search a continuous  $k$ -dimensional search space ( $k$  being the number of controller parameters) for the optimal setting.

The purpose of recoding both the input and output variables of the system into fuzzy variables is to reduce the effort necessary to find the optimal controller parameters. The search is performed only in the much reduced discrete search space of the (discrete) class values of these parameters. If necessary, even an exhaustive search of this search space may be affordable, which will guarantee that the global optimum is found (at least in terms of the discrete search space). The fuzzy membership functions are then used to interpolate in an efficient fashion between neighboring class values, i.e., to deduce quasi-optimal continuous values of the controller parameters. The task of the fuzzy inferencing algorithm (Mugica and Cellier, 1993) is to ensure that the interpolation between neighboring discrete solution points is smooth and preserves, on the way, as much information about the system to be modeled as possible.

The rules and fuzzy membership functions employed in a fuzzy controller are usually determined heuristically, i.e., they are manually coded on the basis of an intuitive understanding

of the functioning of the underlying process to be controlled.

A systematic design of the rules and/or their accompanying fuzzy membership functions has been attempted in the past. For example, a type of *genetic algorithm* (Goldberg, 1989) has been successfully employed to optimize the behavior of a fuzzy controller used in an autonomous spacecraft rendezvous maneuver (Karr et al., 1989). More recently, a neural network of the associative memory type was employed to initially train (off-line) and then adapt (on-line) the parameters of a fuzzy controller for an inverted pendulum (Kosko, 1992).

This article presents a new systematic design of fuzzy controllers that can be used to control any plant for which the inverse dynamics problem can be solved. The methodology employed in the design is centered around *fuzzy inductive reasoning*, (Li and Cellier, 1990; Cellier, 1991a) a technique geared at the qualitative simulation of dynamical continuous-time processes (Cellier et al., 1992).

Fuzzy inductive reasoning is accomplished using SAPS-II (Cellier and Yandell, 1987), a software that evolved from the General System Problem Solving (GSPS) framework (Uyttenhove, 1979; Klir 1985, 1989). SAPS-II is implemented as a (FORTRAN-coded) function library of CTRL-C (Systems Control Technology, 1985). A subset of the SAPS-II modules, namely the recoding, forecasting, and regeneration modules, have also been made available as an application library of ACSL (Mitchell & Gauthier Assoc., 1986), which is the software used in the mixed quantitative and qualitative simulation runs.

The underlying *inverse dynamics problem* is a well-known control problem that has been studied extensively, particularly in the context of robot control. Given the desired path of the end-effector (the result of solving the path planning problem), find the optimal position of each joint (inverse kinematics problem), then find the optimal forces and torques in each joint (inverse dynamics problem) (Fu et al., 1987). It is not the purpose of this article to reiterate on inverse dynamics. An example for which the solution of the inverse dynamics problem is trivial has been selected to show how to apply the fuzzy inductive reasoning methodology to the design of fuzzy controllers.

It is important to remark that, although the

example chosen is very simple and linear, neither the complexity nor the linearity of the chosen application are important to the success of applying the proposed inductive reasoning methodology to qualitatively modeling the dynamics of a system or controller. Very complex and highly nonlinear systems have already been successfully modeled using the proposed methodology (Cellier et al., 1992; Albornoz and Cellier, 1993a, 1993b).

This is a proof-of-concept article. The application chosen to demonstrate the systematic fuzzy controller design methodology is simple and generic. It is simple enough to be described in full, yet complex enough to prove the practicality of the approach. More extensive applications, such as fuzzy control applied to cargo ship steering, fuzzy control of a double inverted pendulum, and fuzzy control of a large robot arm are currently under development by the authors of this article to study the true merits as well as limitations of the chosen approach.

## FUZZY INDUCTIVE REASONING

### FUZZY RECODING

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 97°F contains more information than the value "hot." Fuzzy recoding avoids this problem.

Figure 1 shows the fuzzy recoding of a variable called "systolic blood pressure." For example, a quantitative systolic blood pressure of 135.0 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 triple. Any systolic blood pressure with a quantitative value between 100.0 and 150.0 will be recoded into the qualitative value "normal." The fuzzy membership function denotes the value of the bell-shaped curve shown in Figure 1, always a value between 0.5 and 1.0. It was decided to use bell-shaped fuzzy membership functions rather than the more commonly used triangular ones. This membership function can be easily calculated using the equation

$$Memb_i = \exp(-\tau_i \cdot (x - \mu_i)^2), \quad (1)$$

where  $x$  is the continuous variable to be recoded,

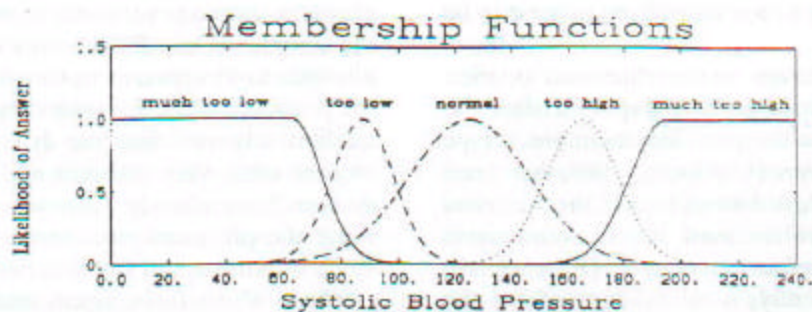


Figure 1. Fuzzy recoding.

$\mu_i$  is the algebraic mean between two neighboring landmarks, and  $\tau_i$  is determined such that the membership function,  $Memb_i$ , degrades to a value of 0.5 at the landmarks. Contrary to other fuzzy approaches, the tails of the membership functions ( $Memb_i < 0.5$ ) are ignored in the method described in this article. 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. Obviously, the qualitative triple contains the same information as the original quantitative variable. The quantitative value can be regenerated accurately, i.e., without any loss of information, from the qualitative triple.

At this point, the question can be raised of how many discrete levels should be selected for each state variable, and where the borderlines (landmarks) that separate two neighboring regions from each other are to be drawn.

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

$$n_{rec} \geq 5 \cdot n_{leg} = 5 \cdot \prod_i k_i, \quad (2)$$

where  $n_{rec}$  denotes the total number of recordings, i.e., the total number of observed states,  $n_{leg}$  denotes the total number of distinct legal states,  $i$  is an index that loops over all variables in the state, and  $k_i$  denotes the number of levels that the  $i^{\text{th}}$  variable can assume. The number of variables is usually given, and the number of

recordings is frequently predetermined. In such a case, the optimum number of levels,  $n_{lev}$ , of all variables can be found from the following equation:

$$n_{lev} = \text{ROUND}(n_{var} \sqrt{n_{rec}/5}), \quad (3)$$

assuming that all variables are classified into the same number of levels. 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 controllers usually require between 7 and 13 classes for each variable (Majers and Sherif, 1985; Aliev et al., 1992; Wu et al., 1992). 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 3 or 5, a number confirmed by several practical applications (Vesanterä and Cellier, 1989; Cellier, 1991c; Albornoz and Cellier, 1993a, 1993b).

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. The most effective way is based on the idea that the expressiveness (or information contents) of the model will be maximized if each level is observed equally often. 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  $n_{lev}$  segments of equal length, and the landmarks are chosen anywhere between the extreme values of neighboring segments, e.g., using the arithmetic mean values of neighboring observed data points in different segments.

#### FUZZY OPTIMAL MASKS

By now, the quantitative trajectory behavior has been recoded into a qualitative episodic behavior. In SAPS-II, the episodic behavior is stored in a *raw data matrix*. Each column of the raw data matrix represents one of the observed variables, and each row of the raw data matrix represents one time point, i.e., one recording of all variables or one recorded state. The values of the raw data matrix are in the set of legal levels that the variables can assume, that means, they are all positive integers, usually in the range from "1" to "5" (SAPS-II uses integers in place of symbolic values to represent qualitative levels).

**Masks as Qualitative Models.** How does the episodic behavior support the identification of a model of a given system for the purpose of forecasting the future behavior for any given input stream?

A continuous trajectory behavior has been recorded and is available for modeling. It is further assumed that the inputs into the real system and the outputs that can be measured are known. The trajectory behavior thus can 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 and three outputs:

$$\begin{array}{l}
 \text{time} \\
 0.0 \\
 \delta t \\
 2 \cdot \delta t \\
 3 \cdot \delta t \\
 \vdots \\
 (n_{rec} - 1) \cdot \delta t
 \end{array}
 \begin{pmatrix}
 u_1 & u_2 & y_1 & y_2 & y_3 \\
 \dots & \dots & \dots & \dots & \dots \\
 \dots & \dots & \dots & \dots & \dots \\
 \dots & \dots & \dots & \dots & \dots \\
 \dots & \dots & \dots & \dots & \dots \\
 \vdots & \vdots & \vdots & \vdots & \vdots \\
 \dots & \dots & \dots & \dots & \dots
 \end{pmatrix}
 \quad (4)$$

To avoid possible ambiguities, it is defined that the terms "inputs" and "outputs," when used in this article 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 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

$$\begin{aligned}
 y_1(t) &= \tilde{f}(y_3(t - 2\delta t), u_2(t - \delta t), \\
 & y_1(t - \delta t), u_1(t)).
 \end{aligned}
 \quad (5)$$

Eq. (5) can be represented as follows:

$$\begin{array}{l}
 \Delta x \\
 t - 2\delta t \\
 t - \delta t \\
 t
 \end{array}
 \begin{pmatrix}
 u_1 & u_2 & y_1 & y_2 & y_3 \\
 0 & 0 & 0 & 0 & -1 \\
 0 & -2 & -3 & 0 & 0 \\
 -4 & 0 & +1 & 0 & 0
 \end{pmatrix}
 \quad (6)$$

The negative elements in this matrix are referred to as *m-inputs*. These *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,  $y_3(t - 2\delta t)$ , whereas the second *m-input* refers to the input variable  $u_2$  one sampling interval in the past,  $u_2(t - \delta t)$ , etc.

In inductive reasoning, such a representation is called a *mask*. A mask denotes a dynamic relationship among qualitative variables. A mask has the same number of columns as the episodic behavior to which it should be applied and it has a certain number of rows. The number of rows of the mask matrix is called the *depth* of the mask.

The mask can be used to flatten a dynamic relationship out into a static relationship. It can be shifted over the raw data matrix, the selected *m-inputs* and *m-output* can be extracted from the raw data, and they can be written next to

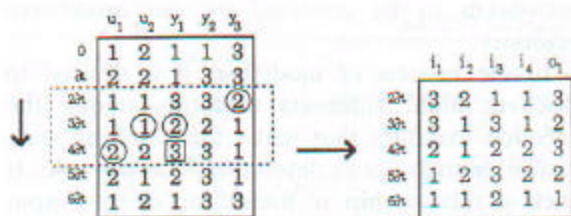


Figure 2. Flattening dynamic relationships through masking.

each other in one row of the so-called *input/output matrix*. Figure 2 illustrates this process. The position of the negative (input) elements of the mask matrix can be interpreted as round holes in a mask made from cardboard, whereas the single positive (output) element of the mask matrix can be viewed as a square hole. The zero elements are covered up by the nontransparent cardboard. The mask (this is where the name of the matrix originally came from) is then shifted downward along the recoded raw data matrix, and the numbers underneath the holes are read out and written into a single row of the input/output matrix. After the mask has been applied to the entire raw data, the formerly dynamic episodic behavior has become static, i.e., the relationships are now contained within single rows.

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.

It has not been discussed yet how the time distance between two logged entries of the trajectory behavior,  $\delta t$ , is chosen in practice. In a combined quantitative/qualitative simulation (mixed simulation),  $\delta t$  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 the object of intensive research. In general, experience has shown that the mask should cover the largest time constant that has to be captured in the model.

If the trajectory behavior stems from measurement data, a Bode diagram of the system to be modeled should be made. This enables us to determine the eigenfrequencies of the system, and, in particular, the smallest and largest eigen-

frequencies. 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:

$$t_{settling} = \frac{2\pi}{\omega_{low}}; \quad t_{fast} = \frac{2\pi}{\omega_{high}}. \quad (7)$$

The mask depth should be chosen as an integer approximation of the ratio between the largest and smallest time constants to be captured in the model plus one:

$$depth = \text{INT}\left(\frac{t_{settling}}{t_{fast}}\right) + 1, \quad (8)$$

but this ratio should not be much larger than 3 or 4. Otherwise, the inductive reasoner will not work very well, because the computing effort grows factorially with the size of the mask. Multiple frequency resolution in inductive reasoning is still an area of open research.

If the chosen mask depth is 3, the mask spans a time interval of  $2\delta t$ , thus:

$$\delta t = \frac{t_{settling}}{2}. \quad (9)$$

**Finding the Optimal Mask.** How is a mask found that, within the framework of all allowable masks, represents the most deterministic state transition matrix? This mask will optimize the predictiveness of the model. In SAPS-II, 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 a potential  $m$ -input, a  $(+1)$  element where the mask has its  $m$ -output, and  $(0)$  elements to denote forbidden connections. Thus, a good mask candidate matrix for the previously introduced five-variable example might be:

$$\begin{array}{c}
 t \setminus x \\
 t - 2\delta t \\
 t - \delta t \\
 t
 \end{array}
 \begin{pmatrix}
 u_1 & u_2 & y_1 & y_2 & y_3 \\
 -1 & -1 & -1 & -1 & -1 \\
 -1 & -1 & -1 & -1 & -1 \\
 -1 & -1 & +1 & 0 & 0
 \end{pmatrix} \quad (10)$$

In SAPS-II, the FOPTMASK routine determines



the optimal mask from a raw data matrix, the fuzzy memberships of the variables, a mask candidate matrix, and a parameter that limits the maximum tolerated mask complexity, i.e., the largest number of nonzero elements that the mask may contain. FOPTMASK searches through all legal masks of complexity two, i.e., all masks with one  $m$ -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. A good value for the maximum complexity is usually five or six.

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 its forecasting power.

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

$$H_i = \sum_{o \in \mathcal{O}} p(o|i) \cdot \log_2 p(o|i), \quad (11)$$

where  $p(o|i)$  is the conditional probability of a certain output state  $o$  to occur, given that the input state  $i$  has already occurred. The term probability is meant in a statistical rather than in a true probabilistic sense. It denotes the quotient of the observed frequency of a particular state divided by the highest possible frequency of that state.

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

$$H_m = - \sum_{i \in \mathcal{I}} p(i) \cdot H_i, \quad (12)$$

where  $p(i)$  is the probability of that input state to occur. 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

$$H_r = 1.0 - \frac{H_m}{H_{max}}. \quad (13)$$

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

The fuzzy membership associated with the value of a qualitative variable is a *measure of confidence*. In the computation of the input/output matrix a confidence value can be assigned to each row. The confidence of a row of the input/output matrix is the joint membership of all the variables associated with that row (Li and Cellier, 1990).

The joint membership of  $i$  membership functions is defined as the smallest individual membership:

$$\begin{aligned} Memb_{\text{joint}} &= \bigcap_{v_i} Memb_i \\ &= \inf_{v_i} (Memb_i) \stackrel{\text{def}}{=} \min_{v_i} (Memb_i). \end{aligned} \quad (14)$$

The confidence vector indicates how much confidence can be expressed in the individual rows of the input/output matrix.

The *basic behavior* of the input/output model can now be computed. It is defined as an ordered set of all observed distinct states together with a measure of confidence of each state. Rather than counting the observation frequencies (as would be done in the case of a probabilistic measure), the individual confidences of each observed state are accumulated. If a state has been observed more than once, more and more confidence can be expressed in it. Thus, the individual confidences of each observation of a given state are simply added together.

To be able to still use the Shannon entropy, which is a *probabilistic* measure of information content, in the computation of the fuzzy optimal mask, the accumulated confidences must first be converted back to values that can be interpreted as conditional probabilities. To this end, the confidences of all states containing the same input state are added together, and the confidence of each of them is then divided by this sum. The resulting normalized confidences can be interpreted as conditional probabilities.

Application of the Shannon entropy to a confidence measure is a somewhat questionable undertaking on theoretical grounds, because 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, (Shafer, 1976; Klir, 1989), 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.

One problem still remains. This 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. Because 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. This obviously leads to a totally deterministic state transition matrix, and  $H_r$  assumes a value of 1.0. Yet the predictiveness of the model will be dismal, because in all likelihood the next predicted state has never before been observed, and that means the end of forecasting. Therefore, this consideration must be included in the overall quality measure.

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 (Li and Cellier, 1990):

$$O_r = \frac{5 \cdot n_{5 \times} + 4 \cdot n_{4 \times} + 3 \cdot n_{3 \times} + 2 \cdot n_{2 \times} + n_{1 \times}}{5 \cdot n_{leg}} \quad (15)$$

where

- $n_{leg}$  = 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,  $O_r$ :

$$Q_m = H_r \cdot O_r \quad (16)$$

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

In SAPS-II, the FOPTMASK function returns the overall best mask found in the optimization; a row vector that contains the Shannon entropies of the best masks for every considered complexity,  $H_m$ ; another row vector containing the corresponding uncertainty reduction measures,  $H_r$ ; and yet another row vector listing the quality measures,  $Q_m$ , of these suboptimal masks. Finally, FOPTMASK also returns the *mask history matrix*, a matrix that consists of a horizontal concatenation of all suboptimal masks. One of these masks is the optimal mask, which, for reasons of convenience, is also returned separately.

#### FUZZY FORECASTING

Once the optimal mask has been determined, it can be applied to the given raw data matrix, resulting in a particular input/output matrix. Because the input/output matrix contains functional relationships within single rows, the rows of the input/output matrix can now 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. The mask is simply shifted further down beyond the end of the raw data matrix, the values of the  $m$ -inputs are read out from the mask, and the behavior matrix is used to determine the future value of the  $m$ -output, which can then be copied back into the raw data matrix. In fuzzy forecasting, it is essential that, together with the qualitative output, a fuzzy membership value and a side value are also forecast. Thus, fuzzy forecasting predicts an entire qualitative triple from which a quantitative variable can be regenerated whenever needed.

In fuzzy forecasting, 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 cheap approximation of the regenerated quantitative signal

$$d_i = 1 + side_i * (1 - Memb_i) \quad (17)$$

is computed for every element of the new input state, and the regenerated  $d_i$  values are stored in a vector. This reconstruction is then repeated for all previous recordings of the same input state. Finally, the  $\mathcal{L}_2$  norms of the difference between the  $d$  vector of the new input state and the  $d$  vectors of all previous recordings of the same input state are computed, and the previous recording with the smallest  $\mathcal{L}_2$  norm is identified. Its *output* and *side* values are then used as forecasts for the *output* and *side* values of the current state.

Forecasting of the new membership function is done a little differently. Here, the five previous recordings with the smallest  $\mathcal{L}_2$  norms 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 forecast for the fuzzy membership function of the current state.

Absolute weights are computed as follows:

$$w_{abs_k} = \frac{d_{max} - d_k}{d_{max}}, \quad (18)$$

where the index  $k$  loops over the five closest neighbors, and  $d_i \leq d_j$ ,  $i < j$ ;  $d_{max} = d_5$ . The absolute weights are numbers between 0.0 and 1.0. Using the sum of the five absolute weights,

$$s_w = \sum_{\forall k} w_{abs_k}, \quad (19)$$

it is possible to compute relative weights:

$$w_{rel_k} = \frac{w_{abs_k}}{s_w}, \quad (20)$$

Also the relative weights are numbers between 0.0 and 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:

$$Memb_{out_{new}} = \sum_{\forall k} w_{rel_k} \cdot Memb_{out_k}. \quad (21)$$

The fuzzy forecasting function will usually give a more accurate forecast than the probabilistic forecasting function. A comparative study of the most commonly used inferencing methods and the five-nearest-neighbors defuzzification method is presented in Mugica and Cellier (1993). This method allows us to retrieve pseudo-continuous output signals with a high quality using the REGENERATE function. This means also that a forecast of the continuous-time signals can be obtained (Cellier, 1991a). Notice that the REGENERATE function is the inverse process of the RECODE function.

## FUZZY CONTROLLER DESIGN

In the previous section of this article, the fundamentals behind fuzzy inductive reasoning were outlined. Now, these elements will be used in the design of a systematic methodology for the development of fuzzy controllers. Details of the advocated approach will be presented by means of a simple example.

The procedure for the design of fuzzy controllers consists of two main stages. In the first stage, the fuzzy controller parameters are identified, and in the second stage, the controller is integrated into the overall system.

### FIRST STAGE: FUZZY CONTROLLER MODELING

"Measurement data" are obtained from the experiment shown in Figure 3. The model consisting of the desired closed-loop system together with the inverse plant dynamics is simulated during a given period of time with a preselected sampling rate. The sampling rate is chosen in accordance with the rules described in the subsection "Fuzzy Optimal Masks." A binary random input,  $r$ , is applied to the model input. This type of input excites the system optimally well at all frequencies. At the output of the closed-loop

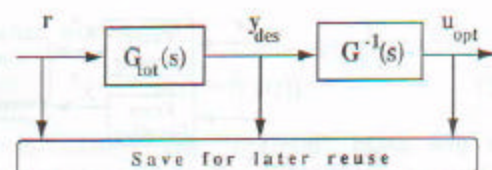


Figure 3. Data extraction.

transfer function,  $G_{tot}(s)$ , the desired output signal  $y_{des}$ , is measured. This signal is fed into the inverse transfer function,  $G^{-1}(s)$ , of the plant. As a result, the optimal control input,  $u_{opt}$ , is found. All three variables,  $r$ ,  $y_{des}$ , and  $u_{opt}$  are stored in a *measurement matrix*.

With the data obtained in this manner and using the inductive reasoning approach described in previous sections, it is possible to build a set of optimal masks that characterize the desired controller. This is accomplished by means of fuzzy recoding and fuzzy optimal mask synthesis (both discussed previously). Each of the masks found in this fashion represents a model of a fuzzy controller for the desired plant.

Once the best mask is selected, it is necessary to check its predictive power by means of fuzzy forecasting and fuzzy signal regeneration. At this point, it may happen that the selected mask exhibits unwanted behavior due to a poor selection of the sampling rate, or due to other inherent system problems such as instabilities or algebraic loops (direct coupling between input and output variables). In these cases, an analysis of the problem must be made to determine the possible causes. If the problem is caused by a bad selection of the sampling rate, it will be necessary to return to the previous step and redesign the fuzzy controller. If the problem is caused by the system structure, a search through the mask history, i.e., through the set of suboptimal masks, has to be performed to find a mask that exhibits good forecasting properties.

#### SECOND STAGE: FUZZY CONTROLLER INTEGRATION

The second stage in the fuzzy controller design process is the combination of this controller with the plant in a single system, the control system. Once again, this is accomplished by using the inductive reasoning technique described in the second section. The overall control system is shown in Figure 4.

The crisp system input,  $r$ , is converted to a

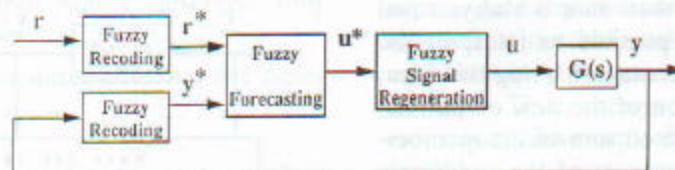


Figure 4. Fuzzy controller design.

fuzzy variable,  $r^*$ , by means of fuzzy recoding. Similarly, also the system output,  $y$ , is converted to a fuzzy variable,  $y^*$ . The fuzzy controller uses these two fuzzy variables to compute a fuzzy control input,  $u^*$ , by means of fuzzy forecasting. The fuzzy control input is then converted back to a crisp control signal,  $u$ , by means of fuzzy signal regeneration.

Finally, the results of the integrated fuzzy control system can be compared with those of the desired system.

#### A SIMPLE EXAMPLE

Given a linear SISO plant with the transfer function

$$G(s) = \frac{s^2 + 3 \cdot s + 7}{s^2 + 5 \cdot s + 10} \quad (22)$$

The plant was chosen as a proper but not strictly proper transfer function because, in this case, computation of the inverse dynamics is trivial. The goal is to design a fuzzy feedback controller around this plant such that the overall system behaves similarly to a linear system with the transfer function

$$G_{tot}(s) = \frac{1}{s + 1} \quad (23)$$

Obviously, this task can be accurately accomplished by the classical controller shown in Figure 5, where

$$G_{comp}(s) = \frac{s^2 + 5 \cdot s + 10}{s \cdot (s^2 + 3 \cdot s + 7)} \quad (24)$$

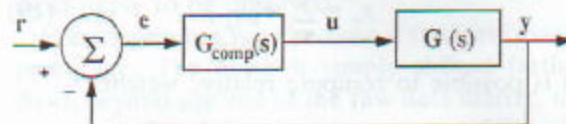


Figure 5. Classical controller design.

In this article, a fuzzy controller will be used instead. The control system with the fuzzy controller is shown in Figure 4.

#### BUILDING THE FUZZY INDUCTIVE MODEL

The design of the fuzzy controller starts by generating data from an appropriate experiment. The traits of this experiment have been described in the previous section. A subset of the collected data will be used to generate the fuzzy model of the controller; the rest of the data will be used for validation purposes. This simulation experiment has been coded in ACSL.

**Fuzzy Recoding of the Crisp Inputs and Outputs.** The raw data matrix is obtained from the measurement matrix by means of (off-line) fuzzy recoding. The first question to be addressed in the recoding process is the selection of the appropriate sampling rate. 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). Because the optimal mask should approximately cover the slowest time constant of the closed-loop system (1.0 sec), a mask depth of 3 would suggest the use of a communication interval of 0.5 sec, i.e., the measurement matrix (and the raw data matrix) should contain entries (rows) that are 0.5 sec apart.

Unfortunately, fuzzy inductive forecasting will predict only one value of  $u$  per sampling interval. Thus, the overall control system of Figure 4 will react like a sampled-data control system with a sampling rate of 0.5 sec. From a control system perspective, the variables should be sampled considerably faster, namely once every 0.05 sec.

The next step is to find the number of discrete levels into which each of these variables will be recoded. For a given example, it was decided that each of the three variables can be sufficiently well characterized by three levels. A discretization of the crisp variables in this manner implies that the number of legal states of the recoded system 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 6.5 sec, are 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 plant characteristics, considerably

more data is needed. It was decided to choose a total simulation time of 100 sec, with 90 sec being used for model identification and the last 10 sec being used for validation. This provides the optimal mask module with 1800 recordings used for model identification, while fuzzy forecasting is carried out over the final 200 steps.

**Fuzzy Optimal Mask of the Controller.** It was decided to choose the following mask candidate matrix:

$$\Delta x \begin{matrix} r & y & u \\ t-2\delta t & \begin{pmatrix} -1 & -1 & -1 \end{pmatrix} \\ t-19\delta t & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ \vdots & \begin{pmatrix} \vdots & \vdots & \vdots \end{pmatrix} \\ t-11\delta t & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t-10\delta t & \begin{pmatrix} -1 & -1 & -1 \end{pmatrix} \\ t-9\delta t & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ \vdots & \begin{pmatrix} \vdots & \vdots & \vdots \end{pmatrix} \\ t-\delta t & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t & \begin{pmatrix} -1 & -1 & +1 \end{pmatrix} \end{matrix} \quad (25)$$

of depth 21.

As mandated by control theory, the sampling interval  $\delta t$  is chosen to be 0.05 sec. Yet, as dictated by the inductive reasoning technique, the control input,  $u$ , at time  $t$  will depend on past values of  $r$ ,  $y$ , and  $u$  at times  $t-0.5$  and  $t-1.0$ .

The optimal mask found with this mask candidate matrix is

$$\Delta x \begin{matrix} r & y & u \\ t-20\delta t & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t-19\delta t & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ \vdots & \begin{pmatrix} \vdots & \vdots & \vdots \end{pmatrix} \\ t-11\delta t & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t-10\delta t & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t-9\delta t & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ \vdots & \begin{pmatrix} \vdots & \vdots & \vdots \end{pmatrix} \\ t-\delta t & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ t & \begin{pmatrix} 0 & -1 & +1 \end{pmatrix} \end{matrix} \quad (26)$$

In other words,

$$u(t) = \tilde{f}(y(t)). \quad (27)$$

Unfortunately, the "optimal" mask will not work in this example. Due to the direct coupling between the plant input,  $u$ , and the plant output,

$y$ , the optimal mask suggests that knowledge of the current value of the plant output,  $y$ , is sufficient to predict the optimal value of the plant input,  $u$ . In an open-loop situation, this is correct. If  $y(t)$  is given,  $u(t)$  can be estimated accurately with this optimal mask. However, this is a chicken-and-egg problem. If  $y(t)$  is given,  $u(t)$  can be computed, and once  $u(t)$  is known,  $y(t)$  can be computed also. There exists an algebraic loop between these two variables.

The fact that the plant was chosen as a proper but not strictly proper transfer function made the solution of the inverse dynamics problem easy, but, at the same time, made the fuzzy control problem considerably more difficult. The optimal mask algorithm optimizes the mask for open-loop. If the plant has low pass characteristics, the optimal mask will also work in a closed-loop setting. However, in the given example, some of the trivial masks (such as the above "optimal" mask) exhibit poor tracking behavior, while others show stability problems.

In this case, it was necessary to search through the mask history, i.e., through the set of suboptimal masks. It was found that the second best mask of complexity four (containing four non-zero elements) exhibits both good tracking behavior and good stability behavior. The mask is as follows:

$$\begin{array}{l}
 t \Delta x \\
 t - 20\delta t \\
 t - 19\delta t \\
 \vdots \\
 t - 11\delta t \\
 t - 10\delta t \\
 t - 9\delta t \\
 \vdots \\
 t - \delta t \\
 t
 \end{array}
 \begin{pmatrix}
 r & y & u \\
 0 & 0 & -1 \\
 0 & 0 & 0 \\
 \vdots & \vdots & \vdots \\
 0 & 0 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 0 \\
 \vdots & \vdots & \vdots \\
 0 & 0 & 0 \\
 -2 & -3 & +1
 \end{pmatrix}
 \quad (28)$$

Thus,

$$u(t) = \tilde{f}(u(t-1.0), r(t), y(t)). \quad (29)$$

**Fuzzy Forecasting and Signal Regeneration.** As mentioned earlier, the first 1800 rows (90 sec) of the raw data matrix were used as past history data to compute the optimal mask. Fuzzy forecasting was used to predict new fuzzy triples of  $u$  for the last 200 rows (10 sec) of the raw data

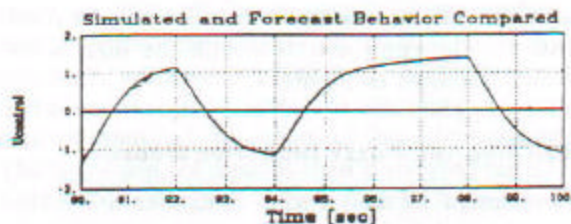


Figure 6. Simulated and forecast control input compared.

matrix. From the predicted fuzzy triples, crisp values were then regenerated.

Figure 6 compares the true "measured" values of  $u$  obtained from the original simulation (solid line), with the forecast and regenerated values obtained from fuzzy inductive reasoning (dashed line) in open-loop, i.e., the "measured" time trajectories  $r(t)$  and  $y(t)$  were optimally recoded into the fuzzy signals  $r^*(t)$  and  $y^*(t)$ . Fuzzy forecasting was then used to estimate the fuzzy signal  $u^*(t)$ . Fuzzy signal regeneration was used to reconstruct the crisp signal  $u(t)$ , which was then compared with the previously "measured" trajectory  $u_{opt}(t)$ . Figure 7 shows the configuration used in the experiment.

The results are encouraging. There is hardly any difference between the optimal trajectory,  $u_{opt}$ , and the output of the fuzzy controller,  $u$ , in open loop. Quite obviously, the optimal mask contains sufficient information to be used as a valid replacement of the true inverse dynamics. Notice that the fuzzy inductive reasoning model was constructed solely on the basis of measurement data.

#### INTEGRATION OF PLANT AND FUZZY CONTROLLER MODELS

At this point, the fuzzy controller can be inserted into the overall system as previously shown in Figure 4. The crisp plant input,  $r$ , is converted to a qualitative triple,  $r^*$ , using fuzzy recoding. Also the crisp plant output,  $y$ , is converted to a qualitative triple,  $y^*$ . From these two qualitative signals, a qualitative triple of the plant input  $u^*$  is computed by means of fuzzy forecasting. This qualitative signal is then converted back to a crisp signal,  $u$ , using fuzzy signal regeneration. The plant itself is described by means of a differential equation model. This experiment has been coded in ACSL/SAPS. The continuous plant dynamics are described by a conventional ACSL program. However, a subset of the func-

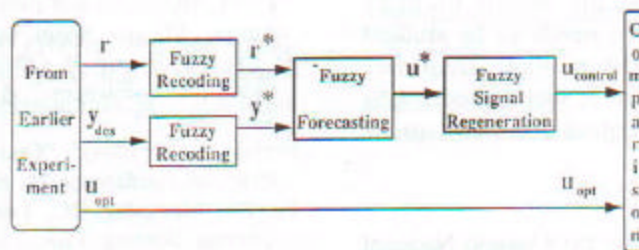


Figure 7. Forecasting of the fuzzy controller.

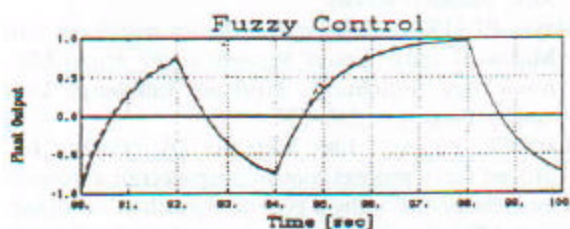


Figure 8. Validation of fuzzy controller.

of the functions contained in the SAPS-II library has been made available also as functions that can be called from within an ACSL program. This facility enables the user to perform mixed quantitative and qualitative simulation experiments (Cellier et al., 1992).

Forecasting was restricted to the last 200 sampling intervals, i.e., to the time span from 90.0 sec to 100.0 sec. Figure 8 compares the desired plant output,  $y_{des}(t)$ , from the purely quantitative simulation (solid line) with the output,  $y$ , of the model containing the fuzzy controller (dashed line).

As can be seen, the plant with the fuzzy controller behaves indeed almost exactly like  $1/(s+1)$ , as desired. The new design approach worked beautifully, although the direct input/output coupling in the plant made the design task considerably more difficult.

It has been shown that fuzzy inductive reasoning can indeed be used to support a systematic design of fuzzy controllers for systems with multiple controller inputs. If the plant contains multiple plant inputs (controller outputs), each controller output is computed separately by a different optimal mask.

## CONCLUSIONS

The example demonstrates the validity of the chosen approach. Control systems containing a

fuzzy controller designed using inductive reasoning are similar in effect to sampled-data control systems. *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 example recoded all three variables into fuzzy variables with the three classes "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 states, 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.

In spite of the nice results that were already obtained, the practicality of this methodology should be proven in a more profound way by applying the technique to more sophisticated examples. Fuzzy inductive reasoning has already been successfully applied to qualitative modeling of complex and highly nonlinear systems (Cellier et al., 1992; Alborno and Cellier, 1993a, 1993b), although not in the context of fuzzy controller design. Current research efforts are focused on the application of this approach to the design of fuzzy controllers for an inverted double pendulum, for a large industrial robot arm, and for steering a large tanker ship.

Some of the knowledge used in the construction of the mixed models, such as the selection of the most appropriate sampling rate, is still of a somewhat heuristic nature. These aspects need to be studied in more depth to guarantee that they can be fully automated, to incorporate them into the design methodology. Also, the applica-

tion of this technique to the design of fuzzy controllers for stiff systems needs to be studied in more depth due to the multiple frequency resolution problem inherent in such systems. It is not practical to simply request the mask to be made deeper and deeper.

This project was sponsored by the Consejo Nacional de Ciencia y Tecnología—México. Their support is gratefully acknowledged. The authors are furthermore thankful to Dr. Rafael Huber of the Institut de Cibernètica of the Universitat Politècnica de Catalunya for his continuous interest in and the support of this project.

## REFERENCES

- Albornoz A de, Cellier FE (1993a): "Variable selection and sensor fusion in automated hierarchical fault monitoring of large scale systems." *Proc. QUARDET'93, Qualitative Reasoning and Decision Technologies*, Barcelona, Spain, pp. 722-734.
- Albornoz A de, Cellier FE (1993b): "Qualitative simulation applied to reason inductively about the behavior of a quantitatively simulated aircraft model." In *Proc. QUARDET'93, Qualitative Reasoning and Decision Technologies*, Barcelona, Spain, pp. 711-721.
- Aliev RA, Aliev FT, Babaev MD (1992): "The synthesis of a fuzzy coordinate-parametric automatic control system for an oil-refinery unit." *Fuzzy Sets and Systems*, **47**, 157-162.
- Babbie E (1989): *The Practice of Social Research*, 5th Ed., Belmont, CA: Wadsworth Publishing.
- Bobrow DG (ed.) (1985): *Qualitative Reasoning about Physical Systems*, Cambridge, MA: MIT Press.
- Cellier FE, Yandell DW (1987): "SAPS-II: A new implementation of the systems approach problem solver." *Int. J. of General Systems* **13**(4): 307-322.
- Cellier FE (1991a): *Continuous System Modeling*, New York: Springer-Verlag.
- Cellier FE (1991b): "Qualitative modeling and simulation—Promise or illusion." In *Proc. 1991 Winter Simulation Conference*, Phoenix, AZ, pp. 1086-1090.
- Cellier FE (1991c): "General system problem solving paradigm for qualitative modeling." In Fishwick PA, Luker PA (eds.): *Qualitative Simulation, Modeling, and Analysis*, New York: Springer-Verlag, pp. 51-71.
- Cellier FE et al. (1992): "Combined qualitative/quantitative simulation models of continuous-time processes using fuzzy inductive reasoning techniques." In *Proc. SICICA'92, IFAC Symposium on Intelligent Components and Instruments for Control Applications*, Málaga, Spain, pp. 589-593.
- de Kleer J, Brown JS (1984): "A qualitative physics based on confluences." *Artificial Intelligence* **24**: 7-83.
- Forbus KD (1984): "Qualitative process theory." *Artificial Intelligence* **24**: 85-168.
- Fu KS, Gonzalez RC, Lee CSG (1987): *Robotics: Control, Sensing, Vision, and Intelligence*, St. Louis: McGraw-Hill.
- Goldberg DE (1989): *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley.
- Hayes PJ (1979): "The naive physics manifesto." In Michie D (ed): *Expert Systems in the Micro-Electronic Age*, Edinburgh, Scotland: Edinburgh University Press, pp. 242-270.
- Karr CL, Freeman LM, Meredith DL (1989): "Improved fuzzy process control of spacecraft autonomous rendezvous using a genetic algorithm." *Proceedings SPIE Intelligent Control and Adaptive Systems Conference*, Philadelphia, PA, pp. 274-288.
- Klir GJ (1985): *Architecture of Systems Problem Solving*, New York: Plenum Press.
- Klir GJ (1989): "Inductive systems modeling: An overview." In Elzas MS, Ören TI, Zeigler BP (eds.): *Modeling and Simulation Methodology: Knowledge Systems' Paradigms*, Amsterdam: Elsevier Science Publishers B.V. (North-Holland), pp. 55-75.
- Kosko B (1992): *Neural Networks and Fuzzy Systems—A Dynamical Systems Approach to Machine Intelligence*, Englewood Cliffs, NJ: Prentice Hall.
- Kuipers B (1986): "Qualitative simulation." *Artificial Intelligence* **29**: 289-338.
- Law A, Kelton D (1990): *Simulation Modeling and Analysis*, 2nd Ed., New York: McGraw-Hill.
- Li D, Cellier FE (1990): "Fuzzy measures in inductive reasoning." *Proceedings 1990 Winter Simulation Conference*, New Orleans, LA, pp. 527-538.
- Maiers J, Sherif YS (1985): "Applications of fuzzy set theory." *IEEE Transactions on Systems, Man and Cybernetics* **SMC-15**: 175-186.
- Mitchell & Gauthier Assoc. (1986): *ACSL: Advanced Continuous Simulation Language—User Guide and Reference Manual*, Concord, MA.
- Mugica F, Cellier FE (1993): "A new method for the defuzzification for the fuzzy inductive reasoning." *Sixth International Symposium on Artificial Intelligence, Intelligence Systems in Industry and Business*, Monterrey, México.
- Systems Control Technology, Inc. (1985): *CTRL-C, A Language for the Computer-Aided Design of Multi-variable Control Systems, User's Guide*, Palo Alto, Ca.
- Shafer, G (1976): *A Mathematical Theory of Evidence*, Princeton, NJ: Princeton University Press.
- Uyttenhove HJ (1979): "SAPS—System approach



problem solver," Ph.D. dissertation, SUNY Binghamton, NY.

Vesanterä PJ, Cellier FE (1989): "Building intelligence into an autopilot—Using qualitative simulation to support global decision making." *Simulation* 52(3): 111–121.

Wu ZQ, Wang PZ, Teh HH (1992): "A rule self-regulating fuzzy controller." *Fuzzy Sets and Systems* 47: 13–21.

Received July 1994

Accepted January 1995

## ABSTRACT

This article reveals the relationship between the control of plant systems, composed of CMAC neural networks and fuzzy logic controllers. A neural network-based learning mechanism called the fuzzy CMAC neural network is established and analyzed. The structure, implementation of the program, features of the CMAC neural network, and the fuzzy logic controller. The proposed combination of fuzzy logic and neural networks can be applied to control systems. It is also possible to design the control system by using the insight of learning mechanism. Some more design examples are given to illustrate the fuzzy CMAC logic, which allows learning rules to be generated by experts using linguistic variables or concepts as the particular problem domain. We believe that the work herein proposed will provide a useful fuzzy CMAC for a wide range of applications. © 1995 John Wiley and Sons, Inc.

## INTRODUCTION

Individually, controller model adaptive controller (CMAC) neural networks and fuzzy logic controllers (FLCs) have received considerable attention in the past. Various methods based on either the CMAC or the FLC have been proposed to design self-organizing and/or self-learning control systems.

The CMAC is a computationally convenient learning network that performs a multi-resolution of a nonlinear function over a particular region of a design space. A CMAC network has the capability to learn efficiently an unknown non-

linear mapping by examples and to produce multiple outputs in response to multiple inputs in real time.

The FLCs do not require a complete analytical model of a dynamic system and provide knowledge-based heuristic controllers for ill-defined and complex systems. FLCs can be analytically validated; however, they are not primarily designed for, nor well suited to, learning.

As two separate control schemes, both CMAC neural networks and fuzzy logic controllers provide valuable approaches for intelligent control systems. This article attempts to bridge the connection between these two approaches and establish the relationship between these two different architectural viewpoints. Various features of both models will be compared and a fuzzy CMAC model that combines the distinctive features of both models will be developed. An approach to apply the fuzzy CMAC to active vibration control is discussed. Preliminary simulation results show very good learning and control performance.

The article is organized as follows: In the next section, the background of CMAC is briefly reviewed. The third section covers some basics of fuzzy set theory and their application in designing FLCs. A comparison of CMACs and FLCs is discussed in the fourth section. A fuzzy CMAC model is then proposed in the fifth section. A self-learning control system structure using fuzzy CMAC is formulated in the sixth section, and some simulation results on the application of fuzzy CMACs to active vibration control is presented.