THE PROBLEM OF DISTORTIONS IN RECONSTRUCTION ANALYSIS

François E. Cellier*
Dept. of Electr. & Comp. Engr.
University of Arizona
P.O.Box 210104
Tucson, AZ 85721-0104, U.S.A
Cellier@ECE.Arizona.Edu
http://www.ece.arizona.edu/~cellier/

Alvaro de Albornoz†
Departament d'ESAII
Universitat Politècnica de Catalunya
Diagonal 647, 2na. planta
Barcelona 08028, Spain
Albornoz@ESAII.upc.es

Abstract

Reconstruction Analysis (RA), as referred to in this article, denotes an implementation of a subset of algorithms from general *Reconstructability Analysis* [Cavallo and Klir, 1981; 1982]. SAPS–II [Cellier, 1991], a Matlab toolbox [MathWorks, 1992], implements a subset of algorithms from *General Systems Problem Solving (GSPS)* [Klir, 1985a], including the *Fuzzy Inductive Reasoning (FIR)* tools [de Albornoz, 1996], as well as the *Reconstruction Analysis (RA)* tools [de Albornoz, 1996]. This article discusses the problem of distortions in *Crisp Reconstruction Analysis (CRA)*, a problem that is inherent in the methodology, and that has, to our knowledge, never been reported. An algorithm is presented that corrects these distortions. Also, the article points out that *Fuzzy Reconstruction Analysis (FRA)* is distortion–free.

1 Introduction

The main idea behind Reconstruction Analysis is quite simple. Given an object in an n–dimensional space. It is always possible to make projections of that object onto lower–dimensional subspaces. If a number of these projections together preserve complete knowledge of the original object, then it should be possible to reconstruct the original object from these projections.

It is a legitimate and interesting question to ask, what is the smallest number of dimensions that are needed to preserve, in a suitable number of projections, total, or at least sufficiently complete, knowledge about the original object.

In *Reconstruction Analysis (RA)*, as implemented in SAPS–II, a special case of the above general problem is discussed. Given a set of observations of variables pertaining to a system, is it possible to decompose this set of n observations (variables) into smaller subsets, such that the accumulated knowledge that can be obtained about the system from analyzing these subsets separately is equal to the knowledge that can be obtained from analyzing the single set of all observations together?

Why is this question relevant? If n observations are available, why would anybody want to analyze subsets of observations separately? The answer to this question is related to the problem of efficiency in analyzing data. Analysis of data for the purpose of knowledge generalization is always equivalent to solving an optimization problem. It is usually much cheaper to solve k optimization problems in $n-r$ variables, than to solve one optimization problem in n variables.

One of these optimization problems is presented in [de Albornoz, 1996]. It is well known that human operators of industrial plants start making mistakes if they are asked to monitor too many variables at a time. This is referred to as the *human overload problem*. However, the same problem also applies to automated monitoring systems, if these systems have to reach decisions within limited time. In [de Albornoz, 1996], Reconstruction Analysis has been used to determine, in a large–scale system environment, what is the smallest number of variables that an automated fault monitoring system has to take into consideration at any point in time, in order to reach reasonable decisions about faults and their causes. In this Ph.D. dissertation, the RA methodology has been applied, in combination with the FIR methodology, to the problem of fault monitoring the behavior of a Boiling Water Nuclear Reactor (BWR).

2 Projections and Recombinations in CRA

Given a set of observations of five variables of a system. Let us assume that each individual observation is in either of the categories 'high' or 'low,' which, for simplicity, shall be denoted as '1' and '0,' respectively.

Since the system has five binary variables, it can be in any one of 32 *legal states*. Let us assume that only eight

*Supported by Ministerio de Educación y Ciencia, Dirección General de Investigación Científica y Técnica (DGICYT), under Grant SAB95-0021-521

†Supported by Consejo Nacional de Ciencia y Tecnología (CONACyT) México

of these legal states were ever observed. The *basic behavior* of the system refers to an alphanumeric enumeration of all observed states together with their observation frequencies:

$$b = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad p = \begin{bmatrix} 0.10 \\ 0.20 \\ 0.15 \\ 0.10 \\ 0.22 \\ 0.03 \\ 0.05 \\ 0.15 \end{bmatrix}$$

where b is the *behavior matrix*, and p is the *vector of relative observation frequencies*. Each row denotes one observed state. The p_i value of the i^{th} state can be interpreted as the probability of its occurrence. Each column of the b matrix denotes one observed variable.

Figure 1 shows a hypothesized decomposition of this 5–variable system, M, into two 4–variable systems, 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 each of the submodels is decomposed further, but their internal structure is not shown.

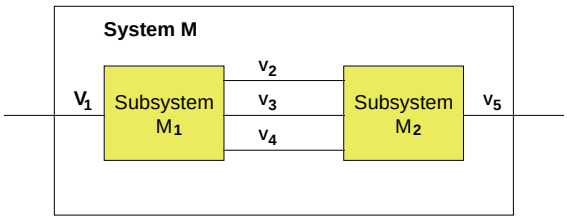

Figure 1: System M with subsystems M_1 and M_2.

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:

$$cs1 \;=\; [\,1\;2\;3\;4\;0\;2\;3\;4\;5\,]$$

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].

Let us assume that the basic behavior presented earlier represents the behavior of system M. It is possible to extract the behavior of the submodels M_1 and M_2 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 spaces of the variables v_1, v_2, v_3, and v_4 (submodel M_1), and v_2, v_3, v_4, and v_5 (submodel M_2). The composite structure of model M has been defined as cs_1. The composite structure of submodel M_1 is cs_2:

$$cs2 \;=\; [\,1\;2\;3\;4\,]$$

The projection to extract the submodel M_1 behavior is accomplished using the SAPS–II function:

$$[b2, p2] \;=\; \text{extract}\,(b, p, cs2)$$

The result of this operation will be the four–variable behavior matrix b_2, and its corresponding probability vector p_2:

$$b2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \qquad p2 = \begin{bmatrix} 0.10 \\ 0.20 \\ 0.15 \\ 0.10 \\ 0.22 \\ 0.03 \\ 0.05 \\ 0.15 \end{bmatrix}$$

The projection onto the subspace of submodel M_2 is made similarly by defining first its composite structure as:

$$cs3 \;=\; [\,2\;3\;4\;5\,]$$

and then by extracting the behavior of submodel M_2:

$$[b3, p3] \;=\; \text{extract}\,(b, p, cs3)$$

The result of this operation is the four–variable behavior matrix b_3 and its corresponding probability vector p_3:

$$b3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \qquad p3 = \begin{bmatrix} 0.03 \\ 0.10 \\ 0.20 \\ 0.20 \\ 0.25 \\ 0.22 \end{bmatrix}$$

This result is more interesting since the number of states and the probability values are different from those of the original behavior $[b, p]$ and the previously extracted behavior $[b_2, p_2]$ that correspond to model M and submodel

M_1, respectively. The behavioral states of submodel M_2 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.

The two projections can now be recombined, by reconstructing a model M_r from the two submodels M_1 and M_2. M_r, just like M, is an object in the five–dimensional space spanned by the five variables v_1, v_2, v_3, v_4, and v_5.

In SAPS–II, the reconstruction is accomplished using the function:

$$[br, pr] \;=\; \text{combine}\,(b2, p2, b3, p3, cs1)$$

The result of this operation will be:

$$br = \left[\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{array}\right] \qquad pr = \left[\begin{array}{c} 0.0231 \\ 0.0769 \\ 0.2000 \\ 0.1500 \\ 0.1000 \\ 0.2200 \\ 0.0069 \\ 0.0231 \\ 0.0500 \\ 0.1500 \end{array}\right]$$

The probability of a reconstructed state x consisting of the substates x_1 and x_2 is computed as the conditional probability of x_2 given x_1 multiplied by the probability of x_1:

$$p(x) = p(x_1, x_2) = p(x_2|x_1) \cdot p(x_1)$$

For example, the state $x =< 0,0,0,0,0 >$ consists of the substates $x_1 =< 0,0,0,0 >$ of submodel M_1, and the substate $x_2 =< 0,0,0,0 >$ of submodel M_2. The first two states of M_2 are compatible with the first state of M_1. Their probabilities were 0.03 and 0.1. Thus:

$$p(x_1) = 0.1$$

is the probability of the first state of M_1, and:

$$p(x_2|x_1) = \frac{0.03}{0.03 + 0.1} = 0.23077$$

is the conditional probability of state x_2 given state x_1, and therefore:

$$p(x) = 0.023077$$

is the value of the first element of vector p_r. The probabilities of the reconstructed model M_r add up to 1.0 as they should.

The reconstruction is *distortion–free*, because its projection onto the same four–dimensional subspaces:

$$\begin{aligned} [br2, pr2] &= \text{extract}\,(br, pr, cs2) \\ [br3, pr3] &= \text{extract}\,(br, pr, cs3) \end{aligned}$$

are identical to the projections of the original model M:

$$\begin{aligned} br_2 &= b_2 \\ pr_2 &= p_2 \\ br_3 &= b_3 \\ pr_3 &= p_3 \end{aligned}$$

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. This process is illustrated in Figure 2.

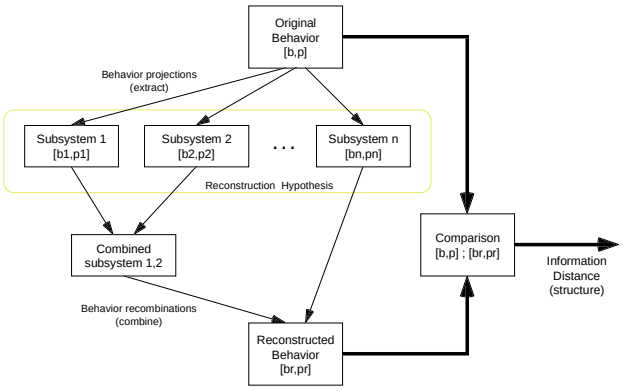

Figure 2: The Reconstruction Process.

For example, let us apply the structure cs_4:

$$cs4 \;=\; [\;2\;4\;0\;3\;5\;0\;4\;5\;0\;1\;2\;3\;]$$

to the same basic behavior $[b, p]$ previously used in this paper. The projections for the substructures in cs_4, called cs_{4a}, cs_{4b}, cs_{4c}, and cs_{4d} can be obtained by specifying:

$$\begin{aligned} cs4a &= [\;2\;4\;] \\ [b4a, p4a] &= \text{extract}\,(b, p, cs4a) \\ cs4b &= [\;3\;5\;] \\ [b4b, p5b] &= \text{extract}\,(b, p, cs4b) \\ cs4c &= [\;4\;5\;] \\ [b4c, p4c] &= \text{extract}\,(b, p, cs4c) \\ cs4d &= [\;1\;2\;3\;] \\ [b4d, p4d] &= \text{extract}\,(b, p, cs4d) \end{aligned}$$

The behavior recombinations must be done in pairs of two, i.e., combining behavior $[b_{4a}, p_{4a}]$ with $[b_{4b}, p_{4b}]$ in a subreconstruction called cs_{4e}:

$$\begin{aligned} cs4e &= [\ 3\ 5\ 0\ 2\ 4\ ] \\ [b4e, p4e] &= \text{combine}\,(b4b, p4b, b4a, p4a, cs4e) \end{aligned}$$

then behavior $[b_{4c}, p_{4c}]$ with $[b_{4e}, p_{4e}]$ in a subreconstruction called cs_{4f}:

$$\begin{aligned} cs4f &= [\ 4\ 5\ 0\ 2\ 3\ 4\ 5\ ] \\ [b4f, p4f] &= \text{combine}\,(b4c, p4c, b4e, p4e, cs4f) \end{aligned}$$

and finally, behavior $[b_{4d}, p_{4d}]$ with $[b_{4f}, p_{4f}]$ in the reconstruction called cs_{4g}:

$$\begin{aligned} cs4g &= [\ 1\ 2\ 3\ 0\ 2\ 3\ 4\ 5\ ] \\ [b4r, p4r] &= \text{combine}\,(b4d, p4d, b4f, p4f, cs4g) \end{aligned}$$

The result of the overall reconstruction, i.e., the reconstructed system, is:

$$b4r = \begin{bmatrix} 0&0&0&0&0 \\ 0&0&0&0&1 \\ 0&0&0&1&1 \\ 0&0&1&0&1 \\ 0&0&1&1&1 \\ 0&1&0&0&0 \\ 0&1&0&0&1 \\ 0&1&1&0&1 \\ 1&0&0&0&0 \\ 1&0&0&0&1 \\ 1&0&0&1&1 \\ 1&0&1&0&1 \\ 1&0&1&1&1 \\ 1&1&0&0&0 \\ 1&1&0&0&1 \end{bmatrix} \qquad p4r = \begin{bmatrix} 0.1202 \\ 0.0930 \\ 0.0867 \\ 0.0776 \\ 0.0724 \\ 0.0564 \\ 0.0436 \\ 0.2200 \\ 0.0120 \\ 0.0093 \\ 0.0087 \\ 0.0259 \\ 0.0241 \\ 0.0846 \\ 0.0654 \end{bmatrix}$$

Unfortunately, although projections across one level are always distortion–free, this property is not preserved across several layers of reconstructions. For example, comparing the projection $[b_{4a}, p_{4a}]$ of the original model with the corresponding projection of the reconstructed model:

$$[b4ra, p4ra] = \text{extract}\,(b4r, p4r, cs4a)$$

the following results are obtained:

$$b4a = b4ra = \begin{bmatrix} 0&0 \\ 0&1 \\ 1&0 \end{bmatrix} \qquad p4a = \begin{bmatrix} 0.33 \\ 0.20 \\ 0.47 \end{bmatrix} \qquad p4ra = \begin{bmatrix} 0.338 \\ 0.192 \\ 0.470 \end{bmatrix}$$

i.e., although the behavior matrices are the same, the probability vectors are different.

A reconstruction that is not indistinguishable from the original when looking at its projections is said to be *distorted*.

3 The Information Distance

A distance function can be defined between the original and the reconstructed behavior, which is called the *Information Distance*. 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 interpreted 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. 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 of the difference between the original and the reconstructed probability vectors.

In SAPS–II, 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 reconstructions are not normally distortion–free, the distance will depend 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 different values of the distance function.

For example, the *structure* function applied to the basic behavior $[b, p]$ presented earlier in this paper together with the structure cs_4:

$$err4 = \text{structure}\,(b, p, cs4)$$

leads to a distance value of $err_4 = 0.3438$. However, if the substructures are rearranged as shown in cs_5:

$$cs5 = [1\ 2\ 3\ 0\ 3\ 5\ 0\ 2\ 4\ 0\ 4\ 5\ ]$$

and the *structure* function is applied to the same behavior $[b, p]$ and to the composite structure cs_5:

$$err5 = \text{structure}\,(b, p, cs5)$$

a distance value of $err_5 = 0.3557$ 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. Usually, the distortions obtained are fairly small.

4 Correction of Distortions

Let us obtain the reconstructed behavior of the previous two examples cs_4 and cs_5 supposing that both structures are applied to the same basic behavior $[b, p]$ that had been used throughout this article. The results are $[b_{4r}, p_{4r}]$ and $[b_{5r}, p_{5r}]$. Since $b_{4r} = b_{5r}$, they can be expressed by a single matrix:

$$b4r = b5r = \begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&1\\ 0&0&0&1&1\\ 0&0&1&0&1\\ 0&0&1&1&1\\ 0&1&0&0&0\\ 0&1&0&0&1\\ 0&1&1&0&1\\ 1&0&0&0&0\\ 1&0&0&0&1\\ 1&0&0&1&1\\ 1&0&1&0&1\\ 1&0&1&1&1\\ 1&1&0&0&0\\ 1&1&0&0&1 \end{bmatrix}$$

$$p4r = \begin{bmatrix} 0.1202\\ 0.0930\\ 0.0867\\ 0.0776\\ 0.0724\\ 0.0564\\ 0.0436\\ 0.2200\\ 0.0120\\ 0.0093\\ 0.0087\\ 0.0259\\ 0.0241\\ 0.0846\\ 0.0654 \end{bmatrix} \qquad p5r = \begin{bmatrix} 0.1148\\ 0.0866\\ 0.0837\\ 0.0837\\ 0.0809\\ 0.0615\\ 0.0464\\ 0.1972\\ 0.0115\\ 0.0087\\ 0.0084\\ 0.0279\\ 0.0270\\ 0.0922\\ 0.0695 \end{bmatrix}$$

Seven additional states have been added in comparison with the original behavior $[b, p]$. The reconstructed states do not depend on the sequence of the reconstruction, only on the structure itself. However, the probability vectors depend slightly on the sequence in which the substructures were specified in the composite structure.

It may be useful to be able to guarantee a distortion–free reconstruction. Neither of the above reconstructions is distortion–free. In the above example, the reconstructed states are known. What is unknown are the correct values of the 15 probabilities associated with the reconstructed states. Let these probabilities be called q_1 to q_{15}.

What is known in addition is that, for a distortion–free reconstruction, 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 cs_4, 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 b_{4r}. It is also known 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, a value of 0.33 is obtained. Thus, it is known that the sum should be:

$$q_1 + q_2 + q_4 + q_9 + q_{10} + q_{12} \;=\; 0.33$$

One 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:

$$M \cdot q \;=\; y$$

where M and y are known, and q is unknown:

$$M = \begin{bmatrix} 1&1&0&1&0&0&0&0&1&1&0&1&0&0&0\\ 0&0&1&0&1&0&0&0&0&0&1&0&1&0&0\\ 0&0&0&0&0&1&1&1&0&0&0&0&0&1&1\\ 1&0&0&0&0&1&0&0&1&0&0&0&0&1&0\\ 0&1&1&0&0&0&1&0&0&1&1&0&0&0&1\\ 0&0&0&1&1&0&0&1&0&0&0&1&1&0&0\\ 1&0&0&0&0&1&0&0&1&0&0&0&0&1&0\\ 0&1&0&1&0&0&1&1&0&1&0&1&0&0&1\\ 0&0&1&0&1&0&0&0&0&0&1&0&1&0&0\\ 1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&1&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&1 \end{bmatrix}$$

$$y = \begin{bmatrix} 0.33\\ 0.20\\ 0.47\\ 0.28\\ 0.30\\ 0.42\\ 0.28\\ 0.52\\ 0.20\\ 0.30\\ 0.15\\ 0.10\\ 0.22\\ 0.03\\ 0.05\\ 0.15 \end{bmatrix}$$

Since the rank of M is nine, there are six degrees of freedom. Consequently, there are many ways for getting distortion–free reconstructions. What can be done is to find a distortion–free reconstruction, the probabilities of which are 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:

$$MM = [\ M\ ;\ \text{eye}(15)\ ]$$
$$yy = [\ y\ ;\ p4r\ ]$$

where *eye* is the Matlab function used to obtain the identity matrix, and p_{4r} is the vector of probabilities found using the recombination algorithm, and solve the overdetermined set of 31 equations in a least square sense:

$$q = MM \backslash yy$$

The remaining distortion can now be computed as:

$$err = MM * q - yy$$
$$err = \text{norm}(err(1:16), 'inf')$$

Clearly, the error is still not zero, i.e., the reconstruction is still not distortion–free. However, the so obtained q vector can be used as the new improved p_{4r} vector, and the same algorithm can be repeated iteratively:

```
while err > 1.0e-6,
  p4r = q;
  yy = [ y ; p4r ];
  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 p_{5r} instead of p_{4r}. The resulting probability vectors were:

$$p_{4r_{corr}} = \begin{bmatrix} 0.1215 \\ 0.0901 \\ 0.0884 \\ 0.0753 \\ 0.0747 \\ 0.0585 \\ 0.0415 \\ 0.2200 \\ 0.0133 \\ 0.0063 \\ 0.0104 \\ 0.0236 \\ 0.0264 \\ 0.0867 \\ 0.0633 \end{bmatrix} \qquad p_{5r_{corr}} = \begin{bmatrix} 0.1206 \\ 0.0898 \\ 0.0895 \\ 0.0751 \\ 0.0749 \\ 0.0589 \\ 0.0411 \\ 0.2200 \\ 0.0128 \\ 0.0074 \\ 0.0097 \\ 0.0242 \\ 0.0258 \\ 0.0877 \\ 0.0623 \end{bmatrix}$$

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* function of SAPS–II in the following way:

$$[brec, prec] = \text{reconstruct}\,(b, p, cs)$$

where $brec$ is the behavior of the reconstructed system and $prec$ its probability vector. However, since the correction is not always needed, and since it reduces the overall efficiency of the algorithm, the previously introduced *structure* command does not correct for distortions.

5 Fuzzy Reconstruction

Rather than interpreting the frequency vector p as a *probability vector* in the classical probabilistic sense, it can also be interpreted as a *confidence vector* in a possibilistic sense. The total confidences don't have to add up to 1.0, but it is acceptable if they do, i.e., confidence vectors cannot usually be interpreted as probability vectors without prior normalization, but probability values can always be re–interpreted as confidence values.

Let us make use of the same example, Model M with behavior $[b, p]$, as before, now interpreting the frequency vector p as the vector of confidences c:

$$b = \begin{bmatrix} 0&0&0&0&1 \\ 0&0&0&1&1 \\ 0&0&1&0&1 \\ 0&1&0&0&0 \\ 0&1&1&0&1 \\ 1&0&0&0&0 \\ 1&0&1&0&1 \\ 1&1&0&0&0 \end{bmatrix} \qquad c = \begin{bmatrix} 0.10 \\ 0.20 \\ 0.15 \\ 0.10 \\ 0.22 \\ 0.03 \\ 0.05 \\ 0.15 \end{bmatrix}$$

The functions *fextract*, *fcombine*, and *fstructure* are the SAPS–II fuzzy functions that correspond to the previously introduced crisp functions *extract*, *combine*, and *structure*. Hence projecting model M onto the four–dimensional space spanned by the variables v_1, v_2, v_3, and v_4, i.e., extracting the behavior of submodel M_1:

$$[b2, c2] = \text{fextract}\,(b, c, cs2)$$

the following submodel is obtained:

$$b2 = \begin{bmatrix} 0&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&1&1&0 \\ 1&0&0&0 \\ 1&0&1&0 \\ 1&1&0&0 \end{bmatrix} \qquad c2 = \begin{bmatrix} 0.10 \\ 0.20 \\ 0.15 \\ 0.10 \\ 0.22 \\ 0.03 \\ 0.05 \\ 0.15 \end{bmatrix}$$

Similarly, a projection of model M onto the subspace spanned by the variables v_2, v_3, v_4, and v_5, i.e., extracting the behavior of submodel M_2:

$$[b3, c3] = \text{fextract}\,(b, c, cs3)$$

results in the following submodel:

$$b3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \qquad c3 = \begin{bmatrix} 0.03 \\ 0.10 \\ 0.20 \\ 0.15 \\ 0.15 \\ 0.22 \end{bmatrix}$$

The behavior matrices are exactly the same as in the crisp case, but the confidence vectors are computed differently. In accordance with [Cavallo and Klir, 1982], the accumulated confidence value of a projected state is defined as the *largest* among all the confidence values associated with previous observations of that state. Contrary to the probabilistic case, the sum of confidence values of all projected states no longer necessarily adds up to 1.0.

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 confidence values. Recombining the projections of the fuzzy models M_1 and M_2:

$$[b4, c4] \;=\; \text{fcombine}\,(b2, c2, b3, c3, cs4)$$

the following result is obtained:

$$b4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad c4 = \begin{bmatrix} 0.03 \\ 0.10 \\ 0.20 \\ 0.15 \\ 0.10 \\ 0.22 \\ 0.03 \\ 0.03 \\ 0.05 \\ 0.15 \end{bmatrix}$$

However, and contrary to the crisp reconstructions, fuzzy reconstructions are always distortion–free even across multiple sequential reconstructions. Thus, no distortion correction algorithm is needed as was true for the crisp case.

In accordance with the suggestions of [Cavallo and Klir, 1982], a different distance measure was used to determine the fuzzy reconstruction error, namely the *Ambiguity Measure*.

The *ambiguity* of the basic behavior $[b, c]$ is computed as:

$$\begin{aligned} a &= 1.0 - \left(\frac{0.03}{8} + \frac{0.02}{7} + \frac{0.05}{6} + \frac{0.05}{4} + \frac{0.05}{2} + \frac{0.02}{1}\right) \\ &= 0.9276 \end{aligned}$$

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:

$$\begin{aligned} a_4 &= 1.0 - \left(\frac{0.03}{10} + \frac{0.02}{7} + \frac{0.05}{6} + \frac{0.05}{4} + \frac{0.05}{2} + \frac{0.02}{1}\right) \\ &= 0.9283 \end{aligned}$$

The distance function then is the difference between the ambiguities of the reconstructed and original behaviors:

$$err_4 \;=\; a4 - a \;=\; 0.00075$$

This same error can be obtained directly using the SAPS–II function *fstructure* that performs all the necessary projections and recombinations for the composite structure cs_4, and then computes the distance function error err_4:

$$err4 \;=\; \text{fstructure}\,(b, c, cs4)$$

In general, crisp and fuzzy reconstructions generate quite similar recommendations, i.e., structures that lead to a small reconstruction error with crisp reconstruction, usually also exhibit a small reconstruction error when the fuzzy reconstruction algorithms are applied, and vice–versa. However, fuzzy reconstruction is cheaper in its computation, and also, it offers always distortion–free reconstructions, which makes this approach more attractive than its crisp counterpart.

6 Conclusions

In this article, a new implementation of a subset of Klir's *Reconstructability Analysis* [Klir, 1985a] in the form of a Matlab toolbox called SAPS–II [Cellier and Yandell, 1987; Cellier, 1991] has been presented. Two different classes of reconstruction algorithms are implemented in SAPS–II, the original (crisp) algorithms proposed in [Klir, 1985a; 1985b; Cavallo and Klir, 1981], and the modified (fuzzy) algorithms proposed in [Cavallo and Klir, 1982]. It was demonstrated that crisp reconstruction is not distortion–free across several layers of extractions and recombinations, whereas the fuzzy reconstruction algorithms remain always distortion–free. A new algorithm for removing the distortion inherent in a crisp reconstruction was also presented.

7 Acknowledgments

The authors wish to express their gratitude toward Dr. Rafael Huber of the Instituto de Robótica y Informática Industrial of the Universitat Politécnica de Catalunya for his enthusiastic support of the research that resulted in this publication.

Bibliography

[**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. (1991)**], *Continuous System Modeling*, Springer–Verlag, New York, USA.

[**Cellier, F.E. and J. López (1995)**], "Causal Inductive Reasoning: A New Paradigm for Data–Driven Qualitative Simulation of Continuous–Time Dynamical Systems," in *Systems Analysis Modelling Simulation*, **18**(1-2), pp. 27–43.

[**Cellier, F.E. and D.W. Yandell (1987)**], "SAPS–II: A New Implementation of the Systems Approach Problem Solver," *International Journal of General Systems*, **13**(4), pp. 307–322.

[**de Albornoz, A. (1996)**], *Inductive Reasoning and Reconstruction Analysis: Two Complementary Tools for Qualitative Fault Monitoring of Large–Scale Systems*, Ph.D. Dissertation, Departament d'Enginyeria de Sistemes i Automàtica, Universitat Politècnica de Catalunya, Barcelona, Spain.

[**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).

[**MathWorks Inc. (1992)**], *Matlab: High Performance Numeric Computation and Visualization Sftware – Reference Guide, Natick, Mass, USA.*