UNIVERSITAT POLITÈCNICA DE CATALUNYA
Programa de doctorat:
TECNOLOGIES AVANÇADES DE LA PRODUCCIÓ

Tesi doctoral

Qualitative Modelling of Complex Systems by Means of Fuzzy Inductive Reasoning. Variable Selection and Search Space Reduction.

Josep Maria Mirats Tur

Directors:

Rafael Huber
(Univ. Politècnica de Catalunya)

François E. Cellier
(Univ. of Arizona)

Barcelona, Octubre 2001

A Tothom, i a Ningú.
Especialment però, a la iaia Maria, peninsular, i a sa güela Maria, insular,
les qui només es preocupaven per la realitat diària però quedaven felices
per qualsevol explicació que els donés sobre la recerca feta.

Agraïments

Abans d'entrar en matèria desitjaria deixar escrit el meu sentiment d'agraïment cap a les següents persones:

Drs. Rafael Huber i François E. Cellier, de la Univ. Politècnica de Catalunya i la University of Arizona respectivament, directors de la tesi, sense la direcció, ajut, crítica i consell dels quals, la realització d'aquesta recerca no hagués arribat on ho ha fet.

Dr. S. Joe Qin, de la University of Arizona, per integrar l'autor en el seu grup de recerca la tardor del 1998. L'experiència guanyada a Austin, Texas, tan professional com personal és indescriptible.

Drs. Carlo Lauro, Roberta Siciliano, Rosanna Verde, de la Università degli Studi di Napoli, per integrar l'autor en el seu grup de recerca la tardor de l'any 1999. Part important de la tesi va néixer a Nàpols. El guanyat i el perdut a la ciutat Partenòpea no té cabuda en simples paraules.

Als companys dels grups de recerca a la Univ. of Austin i a la Univ. degli Studi di Napoli. En especial a Carlos Pfeiffer i Sergio Valle (Univ. of Austin) i a Carmela Cappelli i Claudio Conversano (Univ. degli Studi di Napoli).

Als companys de l'Institut de Robòtica i Informàtica Industrial, IRI, pel suport en la llarga espera.

A na Maria Cristina i en Josep Maria; a na Fefa Colomer; tots ells família.

A en Guillem, na Susanna i, darrerament, n'Aloma.

Als 1.500 fills que han fet arrels a les terres d'Eivissa.

L'últim agraïment és per a qui més ha sofert les crisis doctorals i per a qui, en contrapartida, més ha ajudat a superar-les, na Cristina Rius.

La recerca duta a terme en aquesta tesi ha estat parcialment finançada per la Comisión Interministerial de Ciencia y Tecnología (CICYT) del Govern Espanyol en el marc del projecte "Seguridad de funcionamiento en sistemas dinámicos complejos" amb referència TAP96-0882.

En realidad, sólo existe la dirección que tomamos
M. Benedetti

Contents

1 CONTEXT, MOTIVATION AND OVERVIEW 1

2 FUZZY INDUCTIVE REASONING METHODOLOGY FOR COMPLEX SYSTEMS MODELLING 7

2.0 ABSTRACT 7
2.1 INTRODUCTION AND BACKGROUND 7
2.2 THE FIR METHODOLOGY 10
2.3 GENERAL SYSTEMS PROBLEM SOLVER 12
2.4 RECODING PROCESS 15
2.5 QUALITATIVE MODELLING ENGINE 18
2.5.1 New algorithm to obtain a qualitative model of a complex system *28*
2.6 QUALITATIVE SIMULATION ENGINE 31
2.6.1 Variable acceptability interval *35*
2.7 DEFUZZIFICATION MODULE 37
2.8 CONCLUSIONS 37

3 VARIABLE SELECTION 38

3.0 ABSTRACT 38
3.1 INTRODUCTION 38
3.2 PCA BASED METHODS 41
3.3 UNRECONSTRUCTED VARIANCE METHODOLOGY 45
3.3.1 Original method *45*
3.3.2 Reducing the FIR mask search space using the unreconstructed variance methodology *49*
3.3.2.1 Modelling NOx output using the unreconstructed variance method 50
3.3.2.2 Modelling NOx output using an alternative set of variables 51
3.4 CORRELATION BASED METHODS 52
3.4.1 Simple correlation matrix *53*
3.4.2 Multiple correlation coefficients *56*
3.4.3 Variable similarity measure derived from the state transition matrix *57*
3.5 CONCLUSIONS 63

4 SUBSYSTEM DETERMINATION 64

4.0 ABSTRACT 64
4.1 INTRODUCTION 64
4.2 RECONSTRUCTION ANALYSIS 67
4.2.1 The concepts of Reconstructability *68*
4.2.2 Structure systems in Reconstruction Analysis *69*
4.2.3 Tools available in Reconstruction Analysis *73*
4.2.4 Algorithms to generate reconstruction hypotheses *77*
4.2.5 Application of Reconstruction Analysis to a real system *83*
4.2.6 Application of Reconstruction Analysis in conjunction with a variable selection technique *89*
4.2.7 Understanding the different types of structures in FRA *93*
4.3 USING FIR TO FIND THE STRUCTURE OF A SYSTEM 97
4.4 CORRELATION MATRIX SINGULAR VALUE DECOMPOSITION 100
4.5 FINDING NON-LINEAR RELATIONS BETWEEN VARIABLES AND SUBSYSTEMS 103
4.6 CONCLUSIONS 108

5. METHODOLOGIES LEADING TO A DYNAMIC MODEL 113

5.0 ABSTRACT 113
5.1 INTRODUCTION 113
5.2 METHODOLOGIES THAT LEAD TO A STATIC MODEL 117
5.2.1 Methods based on linear relationship searching *118*
5.2.1.1 FIR models excluding temporal relations 119

5.2.1.2 Ordinary Least Squares method ... 120
5.2.1.3 Principal Components Regression model ... 123
5.2.1.4 Partial Least Squares regression method ... 125
5.2.1.5 Methods based on cluster analysis ... 129
5.2.1.6 Using subsets of variables for static FIR predictions ... 130
5.2.2 Methods based on linear and non-linear relationship search ... *135*
5.2.2.1 Subsystem decomposition algorithm ... 135
5.3 METHODOLOGIES LEADING TO A DYNAMIC MODEL ... 142
5.3.1 Subsystem decomposition method extended with time ... *142*
5.3.2 Delay estimation using energy information ... *150*
5.4 CONCLUSIONS ... 161

6 APPLICATION OF THE PROPOSED METHODOLOGIES TO AN INDUSTRIAL SYSTEM: A GAS TURBINE FOR ELECTRIC POWER GENERATION. ... 164

6.0 ABSTRACT ... 164
6.1 INTRODUCTION ... 164
6.2 TURBINE PRINCIPLES ... 167
6.3 FIR MODEL OF A GENERAL ELECTRIC MARK 5 FRAME 6 GAS TURBINE ... 170
6.3.1 System description ... *171*
6.3.2 Turbine FIR model from its subsystem decomposition ... *175*
6.3.2.1 Static system decomposition ... 176
6.3.2.2 Extended system decomposition including time ... 180
6.3.3 Dynamic FIR model of the gas turbine using energy information ... *188*
6.4 CONCLUSIONS ... 203

7 FINAL CONSIDERATIONS ... 204

7.1 CONTRIBUTIONS ... 209
7.2 FUTURE WORK ... 214
7.3 CONCLUDING REMARKS ... 215

Appendices ... 216

I CASE STUDIES ... 216

I.1 WATER TANK ... 216
I.2 STEAM GENERATOR (BOILER) ... 219
I.2.1 FIR models excluding temporal relations ... *223*
I.3 GARBAGE INCINERATOR SYSTEM ... 223
I.4 GAS TURBINE FOR ELECTRIC POWER GENERATION ... 225

II ASSESSMENT OF RESULTS ... 235

II.1 APPLICATION OF THE ALGORITHM DISCUSSED IN SECTION 2.4.1 TO THE GARBAGE INCINERATOR SYSTEM ... 230
II.1.1 Reducing the mask search space using qualitative data ... *230*
II.1.2 Qualitative simulation results ... *241*
II.2 APPLICATION OF THE ACCEPTABILITY INTERVAL TO FAULT DETECTION IN A COMMERCIAL AIRCRAFT. 243
II.2.1 Previous research and the aircraft model ... *244*
II.2.2 Smooth changes in the aircraft parameters. ... *246*
II.2.3 Crisp detection using smoothed data. ... *247*
II.2.4 Detection with envelopes ... *249*
II.2.4.1 Sudden change detection ... 250
II.2.4.2 Smooth change detection ... 252
II.2.5 Discussion ... *254*

8. REFERENCES AND BIBLIOGRAPHY ... 255

List of figures

Figure 2-1 FIR methodology four main modules. 11
Figure 2-2 The role of GSPS. 13
Figure 2-3 Klir's epistemological levels of systems. 14
Figure 2-4 Example of recoding a quantitative temperature value. 17
Figure 2-5 Fuzzification process example. 18
Figure 2-6 Obtaining static relations by use of a mask. 21
Figure 2-7 Fuzzy forecasting process 31
Figure 2-8 Re-normalisation of a Gaussian to compute a pseudo-regeneration value 32
Figure 2-9 Interval of variable acceptability. 36

Figure 3-1 Unreconstructed variance as the summation of $\hat{u}_i$ and $\tilde{u}_i$. 48
Figure 3-2 Prediction given by a PCA model using input variables 1, 2, 4, 5, 6, 7, and 8. 51
Figure 3-3 NOx output predicted using a PCA model built from variables 2, 3, and 8. 52
Figure 3-4 The $x \cdot \log_2(x)$ function when $x \varepsilon [0,1]$. 60

Figure 4-1 A model with two possible subsystems. 69
Figure 4-2 Seven-variable system with a model consisting of three submodels. 71
Figure 4-3 Seven-variable system with a model consisting of four submodels. 72
Figure 4-4 Reconstruction evaluation process. 73
Figure 4-5 Topological structure obtained by means of a FRA based algorithm. 96
Figure 4-6 Second substructure found for the incinerator system using FIR. 98
Figure 4-7 Possible structure for the incinerator system found with FIR. 99
Figure 4-8 Incinerator system: projections onto the first and second principal axes. 101

Figure 5-1 Expressing a 5-variable dynamic system by means of a 15-variable static one. 114
Figure 5-2 The main lines of thought along the dissertation. 116
Figure 5-3 Boiler OLS prediction using all input variables. 121
Figure 5-4 Boiler OLS predictions using variables 1, 2, 4, 5, and 7 (top), and variables 2, 3, and 8 (bottom). 123
Figure 5-5 Boiler PCR model using all input variables retaining 4 LVs. 124
Figure 5-6 Prediction of the boiler output using a PCR model with variables 2, 4, 5, 6, 7, and 8 (top); and 2, 3, and 8 (bottom). 125
Figure 5-7 PLS model using 4 LVs and all physical variables. 127
Figure 5-8 Prediction of the boiler output using a PLS model with physical variables 2, 4, 5, 6, and 7 (top); and 2, 3, and 8 (bottom). 128
Figure 5-9 Loss of quality for models of complexity 3 (left) and 4 (right), when each of the incinerator system variables is modelled from other variables with which they are related through one or more subsystems. 139
Figure 5-10 Loss of quality for models of complexity 5 (left) and 6 (right), when each of the incinerator system variables is modelled from other variables with which they are related through one or more subsystems. 139
Figure 5-11 Loss of quality for models of complexity 3 (left) and 4 (right), when each of the incinerator system variables is modelled from the complementary sets of variables to those used in Figure 5-9. 140
Figure 5-12 Loss of quality for models of complexity 5 (left) and 6 (right), when each of the incinerator system variables is modelled from the complementary sets of variables to those used in Figure 5-10. 140
Figure 5-13 Comparative loss of quality, when using the two different presented sets of the incinerator variables for models of complexities 3 and 4. 141
Figure 5-14 Garbage incinerator output variable simulated with the complexity 5 model derived from S2. 142
Figure 5-15 Garbage incinerator output variable simulated with the complexity 5 system derived from subsystem S4. 147
Figure 5-16 Flowchart summarizing the subgroup decomposition method. 149
Figure 5-17 Garbage incinerator output variable simulated with the highest quality complexity 5 model when the candidate mask given to FIR is derived from the energy method. 159

Figure 5-18 Garbage incinerator output variable simulated using a complexity-5 model, the inputs of which are those four variables that show the highest frequencies among all the complexity-5 masks. The candidate mask given to FIR is derived from the energy method 160
Figure 5-19 Flowchart of the energy-based method. 160

Figure 6-1 The Hero's aeolipile. 165
Figure 6-2 Steam turbine to operate machinery. 165
Figure 6-3 The basic cycle of a gas turbine. 167
Figure 6-4 Simplified schematic of a gas turbine with a load generator for electric power generation. 168
Figure 6-5 Trajectory of the considered gas turbine output (7200 data points). 170
Figure 6-6 Schematic of the General Electric MARK 5 Frame 6 turbine. 171
Figure 6-7 Control loops for the fuel gas input pipe. 172
Figure 6-8 Projection onto the first and second principal axes. 178
Figure 6-9 Projection onto the first and third principal axes. 178
Figure 6-10 Projection onto the first and second principal axes when time is considered. 182
Figure 6-11 Projection onto the first and third principal axes when time is included. 183
Figure 6-12 Projection onto the first and fourth principal axes when time is included. 184
Figure 6-13 Gas turbine output simulation results using a complexity-5 FIR model derived from subsystem S2 taking into account the time variable. 186
Figure 6-14 Gas turbine output simulation results using a complexity-4 FIR model derived from subsystem S2 taking into account the time variable. 187
Figure 6-15 Gas turbine output simulation results using a complexity 5 FIR model derived from subsystem S6 having into account the time variable. 187
Figure 6-16 Real gas turbine output, 1080 points (15% of the available data). 199
Figure 6-17 Real (continuous line) and simulated (dotted line) gas turbine output variable. 200
Figure 6-18 Real (continuous line) and simulated (dotted line) trajectory between data points 450 to 580. 201
Figure 6-19 Real (continuous line) and simulated (dotted line) data when using the best complexity-4 model. 202
Figure 6-20 On the left, data points from 450 to 520 for the real and simulated data sets when using a complexity-4 model showing details of the real output peak simulation. On the right, data points from 240 to 320 showing an erroneous peak resulting present in the simulation only. 202

Figure I-1 Plant scheme for the water tank system 216
Figure I-2 Real input and output signals of the system 217
Figure I-3 Real and simulated (dotted) output signals when using the quantitative model 217
Figure I-4 Percent of relative error in the quantitative simulation 218
Figure I-5 Real and simulated output when using the qualitative model 219
Figure I-6 Relative error (percent) of the qualitative simulation 219
Figure I-7 Schematic of the boiler process 220
Figure I-8 Original and predicted validation set for the output using FIR 222
Figure I-9 Incineration process scheme 224

Figure II-1 Real and simulated NOx. Last 500 points.   Figure II-2 First 100 NOx FIR simulated points. 242
Figure II-3 Real and simulated NOx. Last 500 points.   Figure II-4 First 100 NOx FIR simulated points. 242
Figure II-5 Quality versus depth of the mask 243
Figure II-6 Input and output variables of the aircraft model 244
Figure II-7 Drag (D): trajectory with a sudden change in the parameters 246
Figure II-8 Drag (D): trajectory with a smooth change in the parameters. 247
Figure II-9 Fault detection scheme 248
Figure II-10 Variable D: real system values and forecast acceptability envelope 250
Figure II-11 Envelopes with smooth change in variable L. 252
Figure II-12 Envelopes with smooth change in variable D. 252
Figure II-13 Envelopes with smooth change in variable GA. 253

List of tables

Table 3-I Unreconstructed variance table for the boiler example. 49
Table 3-II Total unreconstructed variance for each number of principal components. 50
Table 3-III. Groups of linearly correlated variables 54
Table 3-IV Input data correlation matrix for the garbage incinerator system. 55
Table 3-V. Variation coefficient for the garbage incinerator input variables. 55
Table 3-VI Retained and discarded variables for the boiler process when multiple correlation coefficients are used as variable selection method. 57
Table 3-VII Input data similarity matrix for the garbage incinerator system. 62
Table 3-VIII Subgroups of similar variables. 63

Table 4-I Incinerator subsystem decomposition obtained with FRA using e_{max} = 0.01. 92
Table 4-II Incinerator subsystem decomposition obtained with FRA using e_{max} = 0.015. 92
Table 4-III Remaining correlation matrix for the garbage incinerator system after performing a variable selection. 101
Table 4-IV Subsystems resulting from the projection onto the first and second principal axes. 102
Table 4-V Obtained static subsystems for the garbage incinerator system from different principal axes projections. 102
Table 4-VI Obtained static subsystem decomposition for the garbage incinerator system only taking into account linear relations. 103
Table 4-VII Non-linear correlations between non-linear transformed variables of the incinerator and the previously found subsets. 106
Table 4-VIII Final subsystem decomposition for the garbage incinerator system including linear as well as non-linear static relations among variables. 106
Table 4-IX Capturing of binary relations by the three proposed structure identification algorithms. 111

Table 5-I Boiler FIR models without temporal relations. 119
Table 5-II Regression coefficients using OLS with the boiler system. 121
Table 5-III. Regression coefficients for the boiler variables 1, 2, 4, 5, and 7 (left column) and for variables 2, 3, and 8 (right column). 122
Table 5-IV Regression coefficients for all variables when using PCR with the boiler system. 124
Table 5-V. Boiler regression coefficients for variables 2, 4, 5, 6, 7, and 8 (left column); and variables 2, 3, and 8 (right column). 125
Table 5-VI Regression coefficients for all variables when using PLS with the boiler system. 127
Table 5-VII. Boiler regression coefficients for a PLS model for variables 2,4,5,6, and 7 (left column); and variables 2,3, and 8 (right column). 128
Table 5-VIII Boiler variable selection achieved using cluster analysis. 130
Table 5-IX Boiler FIR Dynamical models obtained from different reduced sets of variables. 131
Table 5-X Loss of prediction quality due to selection of variable subsets. 134
Table 5-XI Model search space reduction attained with each of the methods applied to the boiler system. 134
Table 5-XII Garbage incinerator static subsystem decomposition. 136
Table 5-XIII Non-linear correlations between not chosen variables of the incinerator and the five previously identified subsets. 137
Table 5-XIV Final subsystem decomposition for the garbage incinerator system including linear as well as non-linear relations between variables. 138
Table 5-XV. Number of FIR models to compute considering a full candidate matrix. 141
Table 5-XVI Reduction of computation effort in the garbage incinerator system, when using the proposed subsystem decomposition method. 142
Table 5-XVII Garbage incinerator system selected variables, when including time in the subsystem decomposition method. 143
Table 5-XVIII Formed subsystems for the garbage incinerator when only linear relations are considered and time is included. 144
Table 5-XX Final subsystem decomposition when time is included for the garbage incinerator system. 145
Table 5-XXI Delays to consider between inputs and output variable for the garbage incinerator system. 156
Table 5-XXII Results obtained when determining the most important delays in energy terms. 158
Table 5-XXIII Reduction on the computational cost for the garbage incinerator system. 158

Table 6-I Variables of the gas fuel system control module. 173
Table 6-II Turbine variables used to compute the FIR qualitative model. 174
Table 6-III Turbine variables chosen after the correlation analysis of the data has been performed 177
Table 6-IV Subsystems formed from the projection of the first versus the second principal axes. 177
Table 6-V Subsystems formed from the first versus second principal axes projection. 179
Table 6-VI Final formed subsystems for the gas turbine system. 179
Table 6-VII Turbine variables chosen after the correlation analysis of the data including time 181
Table 6-VIII Subsystems formed from the first versus second principal axes projection when time is considered. 182
Table 6-IX Subsystems formed from the first versus third principal axes projection when time is included in the analysis. 183
Table 6-X Subsystems formed from the first versus fourth principal axes projection when time is included in the analysis. 183
Table 6-XI Considered subsystems formed from linear relations between variables when time is included in the gas turbine analysis. 184
Table 6-XII Final susbsystem decomposition when time is included in the gas turbine system. 185
Table 6-XIII Delays with maximum energy related to the output variable. 189
Table 6-XV Number of models to compute for a candidate mask containing 414 and 1984 '-1' elements respectively. 190
Table 6-XVI Estimation of the computing time needed for candidate masks with 414 and 1984 '-1' elements, respectively. 191
Table 6-XVII Number of models to be computed for each allowed complexity with a 84 '-1' element candidate mask. 192
Table 6-XVIII Results from the analysis of all the computed masks, when a depth-2 candidate mask with 84 '-1' elements is proposed to FIR. Enhanced in bold are the m-inputs to the next depth-3 candidate mask. 193
Table 6-XX Results from the analysis of all computed masks when a depth-3 candidate mask with 51 '-1' elements is proposed to FIR. Enhanced in bold are the m-inputs to the next depth-4 candidate mask. 194
Table 6-XXI Results from the analysis of all computed masks when a depth-4 candidate mask with 61 '-1' elements is proposed to FIR. Enhanced in bold are the m-inputs to the next depth-5 candidate mask. 195
Table 6-XXII Results from the analysis of the computed masks when a depth-5 candidate mask with 54 '-1' elements is proposed to FIR. Enhanced in bold are the m-inputs to the next depth-6 candidate mask. 197
Table 6-XXIII Results from the analysis of the computed masks when a depth-6 candidate mask with 47 '-1' elements is proposed to FIR. Enhanced in bold are the m-inputs to the next depth-7 candidate mask. 198
Table 6-XXV Total computation required to obtain a depth-6 gas turbine model using the energy-based method described in this research work. 200

Table I-I Variables of the boiler system 220
Table I-II FIR models without temporal relations 223
Table I-III Garbage incinerator system variables. 224

Table II-I Static model search results 231
Table II-II Good masks obtained by a depth-2 model search. 232
Table II-III Good masks obtained by a depth-3 model search. 233
Table II-IV Good masks obtained by a depth-4 model search. 234
Table II-V Good masks obtained by a depth-5 model search. 235
Table II-VI Good masks obtained by a depth-6 model search. 236
Table II-VII Good masks obtained by a depth-7 model search. 237
Table II-VIII Good masks obtained by a depth-8 model search. 238
Table II-IX Good masks obtained by a depth-9 model search. 239
Table II-X Good masks obtained by a depth-10 model search. 240
Table II-XI Mask depth 11. Simulation results. 241
Table II-XII Computation alleviation achieved when using the proposed algorithm 241
Table II-XIII Alarm vectors using the detection approach proposed in [6,9] with smooth change 249
Table II-XV Alarm vector obtained when performing fault detection with the envelopes approach in a sudden parameter change situation. 251
Table II-XVI Comparing envelopes and crisp detection results for a smooth change situation 253

1 Context, motivation and overview

Modelling and simulating the output or outputs of a system from its inputs has always been an important task within control engineering. It is of interest to be able to predict, if possible on-line, the future behaviour of any one or any subset of the system variables (outputs) given their own current and past behaviour as well as the current and past behaviour of any set of additional auxiliary variables (inputs).

If the prediction of a variable trajectory is available then the system can be optimally controlled. Intelligent controllers frequently operate with look-ahead data in order to compensate for system delays and/or improve their performance. For example, the controllers that regulate the water distribution system of a city may, on the one hand, work with predicted values of water flows, because the water incurs a delay from the time it is released at the reservoir until it arrives at the city where it is to be used; and on the other hand, they may work with predictions of water needs at the time when the water that is currently being released will arrive at the city. Hence tools and techniques for predicting future values of observed trajectory behaviour constitute important elements of intelligent control architectures.

Fuzzy Inductive Reasoning (FIR) [Cellier, 1991], a modelling and simulation methodology capable of generating a qualitative input-output model of a system from real-valued trajectories of its physical variables, often offers excellent features for dealing with the aforementioned modelling and simulating problems. Whereas *deductive* modelling approaches function well in the case of systems, the internal workings of which are well-understood such as electronic circuits, *inductive* modelling approaches should be used whenever the internal equations of the system to be modelled are either unavailable, or contain parameters that cannot be accurately estimated. FIR has been successfully employed to modelling such diverse systems as the central nervous control of the human heart [Nebot *et al.*, 1997], the growth patterns of shrimp populations in semi-intensive shrimp farming [Carjaval and Nebot, 1997], the water demand in the City of Barcelona [López, 1999], and the *NOx* emission level in a steam boiler [Mirats *et al.*, 2000].

The functioning basis of the *Fuzzy Inductive Reasoning* methodology is to qualitatively learn the behaviour of a system from its past real data. That is, thanks to experience gained in the past with similar unrelated events, a pattern can be recognised, i.e., the behaviour of the system can be qualitatively learnt. This is an interesting feature when dealing with ill-defined systems, because it is possible that an accurate description of the system (i.e., a quantitative differential equation based model) is not available, and only data trajectories of the process are at the disposal of the modeller. Most ill-defined systems are unfortunately large-scale systems. It is precisely because, in these systems, output variables depend on so many different inputs that these systems are ill defined.

As explained in Chapter 2, the modelling engine of FIR determines a so-called *optimal mask* that indicates, which variables best explain any given output, and how much time delay these variables should have relative to the chosen output. Unfortunately, any algorithm that can find the optimal mask is necessarily of exponential complexity, i.e., the number of masks to be visited during the search for the optimal mask grows exponentially with the number of available input variables and with the allowed depth of the mask. This makes the FIR methodology, in its actual implementation, impractical for those cases in which it would be most useful, i.e., large-scale systems. Hence the modelling of large-scale systems with FIR could not be done in practice because of the aforementioned computational complexity problem. This had been the primary limitation of the practical use of the FIR methodology.

To this end, the thesis discusses whether sub-optimal search algorithms or methods of pre-simplifying the large-scale system are most suitable for dealing effectively and efficiently with large-scale systems in the quest of deriving qualitative FIR models for them. In other words, the mask search space of the FIR modelling engine must be reduced if one wants to compute a model of a large-scale system in an affordable amount of time.

The declared task of this thesis shall be to determine the most adequate subset of variables (the most promising set of potential inputs of the FIR qualitative model[1]), as judged from previous observations of their comportment, to be used for forecasting the behaviour of any one of the measurement variables. The (large-scale) system from which these observations are taken may be arbitrarily non-linear, and its internal structure may be unknown. It is also reasonable to assume that the system is exerted through unknown inputs, i.e., disturbances that may have significant effects on the system's comportment. Hence the observation variables can be considered together as a single high-dimensional multivariate time-series. Given any variable among the observed set of variables, the task is to determine the smallest set of other relevant variables needed to make optimal predictions of the future behaviour of the selected output variable; that is, to determine the smallest possible set of variables that together contain all, or at least almost all, the information characterising the comportment of the chosen output variable as available through the overall set of measurement variables. If the system is time-variant, it may furthermore happen that the optimal set of input variables depends on the mode in which the system is currently operating.

Hence, in order to reduce the mask search space of FIR, basically two lines of thought are given in the present dissertation. The first one is to simplify the mask search space of FIR, trying to directly simplify the candidate mask that is proposed to FIR. This can be done either directly, by reducing the number of input variables to the FIR model, or indirectly, using sub-optimal mask search algorithms. Both approaches lead to sub-optimal FIR models.

In order to obtain a decent prediction of the output, it may not be necessary to make use of *all* potential inputs of a system. Different input variables often contain redundant information. Moreover, some of the input information may indirectly be captured in the past history of the output. For example, when predicting the water demand of Barcelona, it may suffice to propose a model that predicts tomorrow's water demand on the basis of today's water demand and the water demand six days ago. This autoregressive model does not list a single input variable directly; yet, the dependence on the day of the week is indirectly captured by including the (meanwhile known) water

[1] Later on, in Chapter 2, those input variables shall be called mask inputs (*m-inputs*). An *m-input* is an input variable of the FIR qualitative model. It may be a measured variable as well as delayed version of it. Here, the *m-input* notation has not been used for the clarity of the reader.

demand six days ago, whereas including today's water demand in the model indirectly captures the dependence on the weather.

For linear systems, variable selection techniques have been previously developed, and are widely discussed in the open literature. For example, subsets of variables can be determined using correlation analysis: good candidate input variables need to show a strong correlation with the selected output, yet a weak correlation among each other. Also, techniques have been developed, such as principal component analysis, that furthermore enable the engineer or scientist to find linear combinations of variables that best characterise the selected output.

The case of arbitrarily non-linear systems is far more complicated. It has been shown that, even in quite simple systems, correlation analysis may provide answers that are far from optimal. Other methods, based on information theory or the energy of the signals, need to be used to determine the relationship between variables. Two such techniques that have shown promise in this context are *Fuzzy Inductive Reasoning (FIR)* and *Reconstruction Analysis (RA).*

Either with variable pre-selection or using a sub-optimal mask search algorithm, a sub-optimal FIR model will be obtained, since not all possible models are being considered, i.e., the mask search space is reduced. Such techniques are often justifiable, since different sub-optimal masks may serve almost equally well as the truly optimal mask because of the redundancy contained in the data. Hence, finding the very best of all possible masks may not be a critical need. It may suffice to find a sub-optimal mask, as long as it has a quality that is not much lower than that of the optimal mask. Sub-optimal mask search strategies have the goal of searching for masks of acceptable quality, while keeping the search space sufficiently small, such that the sub-optimal search algorithm is of polynomial rather than exponential complexity.

Sub-optimal mask search strategies were studied before in [Nebot and Jerez, 1997; Jerez and Nebot, 1997; Van Welden, 1999]. These research studies analyse several variants of hill-climbing algorithms, which are of polynomial complexity, but often result in a sub-optimal mask of significantly inferior quality. One of these studies also proposed the use of genetic algorithms, which, in some cases, may work amazingly well. Unfortunately, genetic algorithms cannot be guaranteed to converge in polynomial time, and often also come up with masks of unacceptably inferior quality. A statistical approach based on cross-correlation functions is discussed in [de Albornoz, 1996]. It converges in polynomial time, but only looks at linear relationships between variables, and therefore often finds a sub-optimal mask of highly inferior quality.

In this dissertation, two new sub-optimal mask search algorithms are proposed. The first method is another variant of a hill-climbing technique. It converges more slowly than those algorithms described in [Nebot and Jerez, 1997]; yet, it is more likely to result in a high-quality mask while still converging in polynomial time. The second method is a new variant of a statistical approach that is based on spectral coherence functions. It also converges in polynomial time; yet, contrary to the technique described in [de Albornoz, 1996], it avoids the pitfall on relying on linear relationships only. Thus, it also is more likely to find a high-quality mask.

No sub-optimal mask search strategy can be guaranteed to work always and for every example. Therefore, it is important to make available several techniques that can be used in parallel. If all of these techniques find a mask of similar quality, it is likely that these sub-optimal masks are indeed good masks with quality values not far different from that of the optimal mask.

The second line of research in this dissertation is to obtain a decomposition of the system into subsystems. This would allow obtaining a model of the system from its subsystems, which in turn reduces the computational time needed for the overall effort. Given a k-variable system, the cost of computing a unique k-variable model is much higher than computing a set of p models of $j_p < k$ variables. The task of decomposing a model into subsystems is done sequentially. First, linear static relations between variables are searched in order to form groups of variables that are linearly related. Afterwards, with the variables that do not form part of any of the found groups, non-linear relations are searched. Finally, time will be added to the process in order to cope with the information contained in the time dependence between variables.

Therefore, with these complementary lines of work, two complete methodologies can be proposed, each of which enables the construction of qualitative models of complex systems. The former, based in simplifying the number of potential inputs to the FIR models[2], is an energy-based method, capable of detecting the variables at given delays that are more closely related to the considered output of the system. The latter proposes a decomposition of the overall system into subsystems using a methodology that is based on the methods described in Section 3.4.1 of Chapter 3 and Sections 4.4 and 4.5 of Chapter 4, and completed with the inclusion of time, as explained subsequently in Chapter 5.

Both methodologies can be used in a fairly automated fashion as tools made available within the FIR toolbox. Yet, the thesis treats their component algorithms as individual tools that can be combined flexibly within a system analysis. No methodology can be found that is capable of dealing equally well with all systems. Either of the two methodologies has been found on occasion to outperform the other by leaps and bounds. In some test cases, one of the techniques did not work at all, whereas the other performed splendidly. Often, small variations in how the individual algorithms are combined may greatly enhance the efficiency of the overall method. Therefore, it is well worthwhile to discuss the component algorithms in detail, explain their rationale, and illustrate their use. This is therefore the route that was taken in this dissertation.

The bulk of the research is presented along the chapters of the thesis. A detailed description of the case studies used in the dissertation as well as a discussion of some of the research results obtained has been structured into appendices, in order not to interrupt the flow of thought along the chapters in undue ways. Because of the diverse methodologies used throughout the dissertation, the state of the art of previous research results has not been compiled into a single chapter, but rather dispersed throughout the thesis, where it is being discussed as the different techniques are introduced. The author feels that this decision aids the clarity of the presentation. The contents of the dissertation, hence, may be summarised as follows:

[2] The potential inputs to the FIR models shall be referred to, later on, as the number of '-1' elements existing in the candidate mask proposed to the modelling engine of FIR.

Chapter 2 contains a complete description of the FIR methodology as well as the state of the ongoing research on FIR, making the dissertation self-contained. Section 2.4.1 deals with a new sub-optimal mask search algorithm that is based on proposing to FIR successive mask candidate matrices with increasing depth, enabling the user to find sub-optimal masks (FIR qualitative models) with a significantly reduced computational effort in comparison with an exhaustive search. It has been found that this sub-optimal search algorithm, a variant of the well-known hill-climbing methods, usually finds masks of excellent quality that perform in qualitative simulations almost as well as the truly optimal mask. Section 2.5 tackles the prediction of future output values (qualitative simulation) with FIR. A modified prediction formula is proposed that eliminates some of the known problems with the formulae that were used in the past, and the new concept of variable acceptability, useful for fault detection purposes, is introduced.

In Chapter 3, the terms ‘static’ and ‘dynamic’ are put into the context of the performed research. Different procedures are applied to perform variable selection, at this point only taking into account linear static relations between variables. The method of the unreconstructed variance for the best reconstruction, described in [Dunia and Qin, 1998] in a different context, is one of the methods presented for the purpose of selecting, which input variables to use to model a given system output. This is the task of section 3.3. Other linear statistical methods, based on principal components analysis (PCA) and/or correlation analysis, are also applied so as to obtain a representative subset of variables of the system under study, hence allowing a first reduction in the FIR model search space. In section 3.4.3, the FIR entropy index, H_r, used to compute the quality of a FIR model, is defined as a measure of similarity (alternative correlation index) between two variables.

Chapter 4 deals with methodologies that lead to a subset (subsystem) determination, in which the variables that form these subsets are maximally related among each other. The Reconstruction Analysis methodology, made popular by Klir, is presented as an informational approach to the problem of decomposing a large-scale system into subsystems. Section 4.3 offers a FIR-based algorithm for decomposing systems into subsystems. In Section 4.4, an alternate method based on singular value decomposition is investigated in order to form subsets of linearly related variables. Section 4.5 then introduces a procedure to study the possible non-linear relation between any of the previously formed subsets of variables and a selected output variable. The algorithms offered in this chapter allow the modeller to decompose a large-scale system into subsystems. This permits to split a complex problem into *n* simpler problems that are computationally easier to deal with. Up to this point, non-linear relations are added to the analysis; yet, the inclusion of time has been left for a subsequent step.

Chapter 5 deals with the analysis of methodologies to model a large-scale system using the modelling engine of the FIR methodology. First, a summary of the results obtained with the different statistical methodologies discussed in Chapter 3, as well as the use of other statistical methods, based on regression coefficients for the purpose of selecting input variables, is given. To this end, the sets of proposed inputs obtained by the different variable selection algorithms are offered to the modelling engine inside *FIR*, and the predictions obtained by *FIR* when using the so obtained qualitative models are compared to each other. Up to this point, only linear static relations have been taken into account. Then, non-linear relations are added and a method for obtaining a decomposition into subsystems is proposed using, in order, Sections 3.4.1 of Chapter 3, and Sections 4.4 and 4.5 of Chapter 4. Yet, those sections have presented the subsystem decomposition method studying only static relations between variables. No dynamic information was used at all. The subsystem decomposition method should be extended including dynamic information of the system. The inclusion of time in order to obtain a dynamic decomposition of the system into subsystems is provided in Section 5.3.1. Finally, in Section 5.3.2, a method based on the energy of the available trajectories, that includes both non-linear relations as well as dynamic information, is developed in order to propose a unique sparse candidate mask to FIR for the purpose of simplifying its model search space.

Hence two modelling methodologies are proposed in Chapter 5 that enable the user to obtain a model of a large-scale system in an affordable amount of time. The former proposes a decomposition of the complex system into subsystems. The latter proposes a unique sparse candidate matrix to FIR, in which the number of potential inputs has been reduced studying the cross-energy among the input and output variables of the system. Both methods are derived for the case of MISO systems, but they are also extendable to MIMO systems, as the different analysis steps can simply be repeated for each of the selected output variables separately.

Chapter 6 presents an industrial application of the different methods advocated in the previous chapters. The analysed system is a gas turbine for electric power generation comprising hundreds of variables.

Throughout the dissertation, practical examples are used to illustrate the different algorithms explained. Appendix I contains a detailed description of each one of the examples used in the thesis. Finally in Appendix II, two interesting applications of the new mask search algorithm and the concept of variable acceptability advocated in Chapter 2 are presented.

With the research presented in this thesis, the FIR modelling capabilities have been extended with capabilities for modelling large-scale systems within a reasonable time.

2 Fuzzy Inductive Reasoning Methodology for complex systems modelling

2.0 Abstract

This chapter is devoted to a description of the FIR modelling and forecasting methodology, and the state of the ongoing research on FIR. Of particular relevance to this dissertation is Section 2.5 that deals with the qualitative modelling engine of FIR, i.e., the computation of so-called optimal masks. Subsection 2.5.1 introduces a new algorithm to determine a set of sub-optimal masks that are characterised by a small quality reduction compared to the optimal mask, yet can be found with a much reduced computational effort. Section 2.6 deals with the qualitative simulation engine of FIR, i.e., the algorithms used by FIR for prediction of future output values. Although such algorithms had been previously available as part of the FIR toolbox, Section 2.6 presents an important modification of these algorithms that leads to a significant improvement in prediction accuracy. The improved formulae are based on the new concept of variable acceptability that is presented in Section 2.6.1.

2.1 Introduction and background

Why are qualitative modelling techniques important? Why should not all systems be modelled quantitatively? There are at least three answers to these questions. From a *short-term perspective*, a quantitative (e.g. differential equation based) model of a complex (large-scale) system may not currently be available. To construct such a quantitative model, though theoretically feasible, may not be economical, may be too slow, or may require experimentation with the real system that may not be acceptable. The size of a quantitative model grows with the complexity of the system to be modelled. This is not necessarily true for qualitative, i.e., observation based, models. A highly complex system, such as the human heart, may be modelled successfully using qualitative modelling techniques [Nebot *et al.*, 1998], because facets of interest of the model can be isolated more easily using qualitative modelling techniques. A useful qualitative model of a complex system is not necessarily a large-scale model. Simplifications in quantitative models invariably lead to inaccuracies in the predictions made because of the non-negligible side effects of the un-modelled system dynamics. Observation based models do not necessarily share this fate. Even simple qualitative models, i.e., models comprising few variables, can be used to make highly accurate predictions of a complex system, because the complexity may be hidden in the experience-data base rather than in the model itself.

From a *medium-term perspective*, the complexity of an accurate quantitative model, even if it can be made available, may be undesirable. Model simplifications may be needed in order to be able to discover and isolate particular behavioural patterns, i.e., to recognise the forest for the trees. Qualitative models lend themselves more easily to reductionism than quantitative models. In the line of the examples given in [Cellier, 1991a], a person knows that if he opens his fingers while holding a cup of coffee in his hands, this cup will fall to the ground and break into a million pieces (apart from spreading the coffee all around). It is not necessary or even useful to solve a set of differential equations to come up with this result. But, how can one arrive at the conclusion that a fragile object will fall to the ground and break if one lets go of this object? Surely it is thanks to experience gained in the past with similar unrelated events; a pattern has been recognised, i.e., the behaviour of the system has been qualitatively learnt. This is the functioning basis of the *Fuzzy Inductive Reasoning (FIR)* methodology, to qualitatively learn the behaviour of a system from past observations.

From a *long-term perspective*, high autonomy systems of the future, such as robots roaming around on planet Mars, will require the ability to make sense of their environment, i.e., to make on the fly models of the environment and their interactions with it, models that can be used to make accurate predictions of future behaviour, i.e., that are able to answer all sorts of what-if questions. Qualitative modelling methodologies lend themselves much more easily than their quantitative counterparts to automated modelling.

Hence the study of qualitative modelling and simulation methodologies is important, as it is useful to both science and engineering. *Fuzzy Inductive Reasoning (FIR)* is one among several feasible approaches to qualitative modelling and simulation. It is a methodology that has been studied extensively within our research group at the Universitat Politècnica de Catalunya in recent years, and it is also at the focus point of this dissertation.

This thesis in particular deals with the problem of qualitatively modelling observation-rich large-scale systems, i.e., systems that contain many variables for which observations are available. A qualitative model, which by its very nature is highly reductionistic, i.e., relies on few input variables only, needs to decide, which, among the many available observations, to use as input variables to make predictions about any chosen output.

It is ironic that, while qualitative modelling techniques would be most useful when dealing with such large-scale systems (short-term perspective), FIR up to now has been least capable of successfully coping with large-scale systems. The reason for this discrepancy has to do with a computational complexity issue. Finding the optimal mask, i.e., the best qualitative FIR model, is an *np*-complete problem, i.e., any algorithm that determines the optimal mask must invariably be of exponential computational complexity.

Several previous attempts were made to tackle this problem. [Nebot and Jerez, 1997] proposed a number of different hill-climbing algorithms for determining sub-optimal masks. While these algorithms are indeed of polynomial complexity, they have a tendency of generating masks of inferior quality. The faster the algorithm converges, the less likely it is that a mask of acceptable quality will be found. This dissertation introduces a new class of hill-climbing algorithms that is much more likely to generate high-quality masks. [Jerez and Nebot, 1997] proposed the use of genetic algorithms to determine sub-optimal masks. Whereas genetic algorithms can be tuned more easily than hill-climbing techniques to avoid the problem of homing in on side valleys, these algorithms are not of polynomial complexity. Their convergence speed depends heavily on the terrain encountered in their parameter space. They converge fast on simple problems that can be solved easily by hill-climbing approaches as well, but converge very slowly in those situations where classical hill-climbing algorithms don't work well. Finally, [van Welden, 1999] compared tree classifiers to FIR in their ability to qualitatively model and simulate complex systems. The primary advantage of tree classifiers is that the qualitative modelling algorithms employed by tree classifier techniques are of polynomial complexity, i.e., tree classifiers lend themselves more easily to coping with large-scale systems. The primary disadvantage of tree classifiers is that their models are characterised by poor resolution, i.e., the predictions obtained by the qualitative simulation algorithms associated with tree classifier techniques are of low quality. Whereas it is possible to use tree classifiers as a variable selection technique for FIR, the resulting sets of variables, although found quickly, may not be the most suitable set of variables for a FIR simulation. These and related issues are dealt with explicitly and in much detail in the present thesis.

Fuzzy Inductive Reasoning is a modelling and simulation methodology capable of generating a qualitative input-output model of a system from observations of the trajectory behaviour of its (usually real-valued) variables. The methodology originates with *Inductive Reasoning (IR)*, an inductive modelling technique designed by Klir as part of his *General Systems Problem Solver (GSPS)* framework [Klir, 1985]. A first implementation of IR was made available by Uyttenhove as part of his Ph.D. dissertation [Uyttenhove, 1979]. This implementation was called *Systems Approach Problem Solver (SAPS).* This first implementation of SAPS was not flexible enough due to limitations of the computer science tools available in those days. An improved version of the original SAPS program, re-implemented as a CTRL-C function library, was developed by [Cellier and Yandell, 1987]. This new implementation was called SAPS-II. With the new CTRL-C version of the tool, all the matrix manipulation capabilities necessary to capture and manipulate the GSPS data structures effectively and efficiently were provided.

Then, in the late eighties, fuzzy measures were incorporated into the GSPS framework [Klir and Folger, 1988; Klir 1989]. In parallel, the SAPS-II platform was extended by [Li and Cellier, 1990] to offer fuzzy reasoning capabilities. Accordingly, the enhanced methodology is now called *Fuzzy Inductive Reasoning (FIR)* [Cellier *et al.,* 1992].

Subsequently, a number of different authors have used FIR to qualitatively model and simulate different kinds of systems while constantly improving the methodology. In [Cellier and Mugica, 1992], the FIR methodology was used to design fuzzy controllers. In [Vesanterä and Cellier, 1989; de Albornoz and Cellier, 1993a; Mirats and Huber, 1999], it was applied to fault monitoring in a simulated aircraft model. FIR has been very successful in dealing with biomedical systems [Nebot *et al.*, 1993; Cueva *et al.*, 1997]. In [López *et al.*, 1996; Cellier *et al.*, 1996], FIR is applied to the problem of modelling and predicting time series. In [Uhrmacher *et al.*, 1997], FIR is used in the context of ecological systems. In [Moorthy et al., 1998], it is used to predict the U.S food demand in the 20th century.

Of course, the publications only speak to the successful uses of the methodology, not to the failures. Mugica tried to qualitatively model a 6 degree-of-freedom robot using FIR. He worked for six months on that model, before he finally gave up. He was lacking an appropriate technology for extracting subsets of variables on the fly for any particular manoeuvre at hand. The previously mentioned aircraft model was successful in qualitatively simulating a Boeing 747 aircraft, but the model only dealt with a single facet of the flight, namely high altitude horizontal flight; i.e., the model was able to deal with small signal behaviour around a trimmed flight only. The model did not deal at all with the much more interesting problem of landing the aircraft. The complexity of the system to be modelled had been artificially reduced by limiting the behavioural patterns that were to be captured by the model. Finally, de Albornoz successfully dealt with monitoring faults in a nuclear reactor. The quantitative reactor model contained close to 500 variables. The research was able to generate a hierarchical qualitative FIR model of the reactor able to recognise a number of so-called transients. Yet, the variables used by these FIR models weren't chosen by FIR. They were essentially hand picked. Their selection was based on years of experience with dealing with nuclear reactors. De Albornoz was lacking a methodology that would have allowed him to apply FIR to a 500-variable system directly.

Research on the FIR methodology at the University of Arizona (Tucson, U.S.A.), the Universitat Politècnica de Catalunya (Barcelona, Spain), and later at the Universiteit Gent (Belgium), has lead to a number of Ph.D. dissertations relating to the FIR methodology. [Nebot, 1994] offered the first comprehensive description of the FIR methodology. The dissertation specifically introduced treatment of missing values into the methodology and discussed its application to biomedical systems. [Pan, 1994] provided a neural network architecture for implementing the FIR methodology in real time. [Mugica, 1995] discussed the synthesis of fuzzy controllers on the basis of FIR models of inverse plant dynamics. In [de Albornoz, 1996], FIR was applied to fault monitoring in large-scale industrial plants. FIR was used to model and predict time series in [López, 1999], where new confidence measures of similarity and proximity were provided. In [Van Welden, 1999], the methodology is complemented with tree classification procedures. Finally the present dissertation extends the use of FIR to previously unmanageable large-scale systems.

Recently, the SAPS-II platform has been re-programmed in C as a Matlab toolbox using dynamic storage allocation. This newest version of the code offers faster algorithms and removes the previous limits on matrix sizes. The only limits on system size are the virtual memory of the computer and the patience of the user.

2.2 The FIR methodology

FIR offers four main modules and many auxiliary routines. Two of the main modules are computational engines, a *Qualitative Modelling Engine*, and a *Qualitative Simulation Engine*; the other two are data filters, a *Fuzzification Module*, and a *Defuzzification Module*. Figure 2-1 provides a flow diagram of these main modules. FIR operates on observations of input/output behaviour of multi-input single-output (MISO) systems, and consequently, a MIMO system has to be modelled by several parallel FIR models, one for each output. In order to reason qualitatively about these observed behaviours, real-valued trajectory behaviour needs to be fuzzified, i.e., mapped into a set of fuzzy classes. In FIR, the process of fuzzification is called *recoding*. In this process, real-valued data are mapped into qualitative triples, consisting of a class value (representing a coarse discretization of the original real-valued variable), a fuzzy membership value (denoting the level of confidence in the chosen class), and a side value (telling whether the quantitative value lies to the left, to the right or at the centre of the membership function peak). Section 2.4 offers an accurate insight of the recoding process.

The goal of the Qualitative Modelling Engine (QME) is to determine the best behaviour system from a given data system and a predetermined output variable. The data system is initially provided to the QME in the form of three matrices of equal dimensions: a qualitative class value matrix, a real-valued fuzzy membership matrix, and a ternary side value matrix. Then the QME tries to find relationships among the class values that are as deterministic as possible, trying to discover behavioural patterns among the observations using the information stored in the class value matrix. Such a qualitative relationship, encoded in the form of a matrix, is called a *mask* in the context of FIR. The optimal mask is found by a process of exhaustive search in the discrete search space of the class values. Details about how FIR finds a qualitative model are handed in Section 2.5.

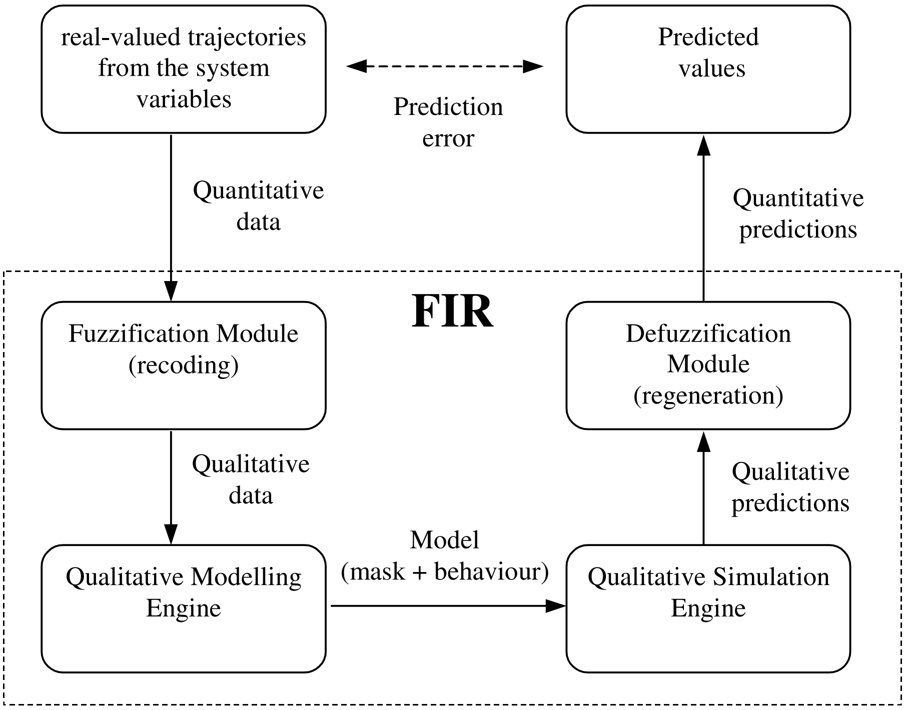

real-valued trajectories from the system variables
Predicted values
Prediction error
Quantitative data
Quantitative predictions
FIR
Fuzzification Module (recoding)
Defuzzification Module (regeneration)
Qualitative data
Qualitative predictions
Model (mask + behaviour)
Qualitative Modelling Engine
Qualitative Simulation Engine

Figure 2-1 FIR methodology four main modules.

The task of the Qualitative Simulation Engine (QSE), given a FIR qualitative model described by means of a mask and three behaviour matrices (constituting a set of fuzzy rules) for every output variable to be modelled, is to forecast a value for each of the chosen output variables. Fuzzy simulation extrapolates the output variable across time using the interpolated values of previous occurrences of similar behavioural patterns. Section 2.6 accurately explains the process of qualitative simulation.

The Defuzzification Module, also called fuzzy regeneration, performs the reverse operation of the Fuzzification Module, converting qualitative triples back to real-valued data. The side value makes it possible to perform the defuzzification of qualitative into quantitative values unambiguously and without information loss. Section 2.7 briefly describes this last filter Module of the FIR methodology.

New contributions to the methodology are also described in this chapter. The first one deals with the computation of a FIR model for a complex, large-scale, system. The algorithm exclusively uses internal FIR tools, so no external help is needed. It is based on reducing the number of models to be visited, thereby obtaining a sub-optimal model. The algorithm is a variant of the well-known hill climbing approaches. Yet it dramatically reduces the probability that the optimisation gets side-tracked, leading to a mask of vastly inferior quality. In all cases that were tested, the new algorithm outperformed the previously available hill climbing methods by leaps and bounds. The algorithm is explained in detail in subsection 2.5.1, and then, in Appendix II.1, it is applied to modelling a garbage incinerator system, the details of which are provided in Appendix I.3. In Section 2.6, where the QSE is explained, the new concept of variable acceptability is introduced, a concept that is subsequently (Appendix II.2) applied to fault detection in an aircraft simulated model. Also in this section, a modification of the formula used for making predictions is proposed.

To better illustrate the FIR methodology, the dissertation makes use of various smaller applications that are easy to describe and to model. For example, a system composed of a water tank, a pump, and a flow sensor with second order dynamics will be used to introduce the modification made in the prediction formula in Section 2.6. This is the kind of systems that can be found, for example, in a simple irrigation system. Appendix I.1 offers a brief explanation of the water tank system as well as a short discussion of how its qualitative model has been obtained.

2.3 General Systems Problem Solver

Fuzzy Inductive Reasoning, FIR, is a modelling and simulation methodology capable of generating a qualitative input-output model of a system from real-valued trajectories of its observed variables. It originates from an inductive modelling technique designed by Klir as part of his *General Systems Problem Solver* (*GSPS*) framework [Cavallo and Klir, 1978; Klir, 1985] that, in turn, is a discipline of *systems science*.

Areas such as cybernetics, information theory, control theory, and artificial intelligence, the origin and major development of which are strongly related to the advances of computer technology, can be viewed as parts of a larger field of inquiry, usually referred to as systems science. These areas deal with system problems in which the informational, relational or structural aspects predominate rather than the physical characteristics of the entities that form the system.

The domain of inquiry of systems science concerns *systems*. The term *system* stands for a set of things and a set of relations between them. Systems can be different depending on the set of considered things (chemical, electrical, social, etc.) or depending on the relations between the components of the system. The kinds of relations characterise various levels of knowledge regarding the phenomena under consideration, i.e., different epistemological levels that will be later explained in more detail.

Knowledge about systems must be gained in order to deal with them. Different systems require different experiments so as to acquire data from them. Then, a methodology is necessary to manage the gathered information. Systems science provides methodological tools for studying relational properties of various classes of systems problems to users in various disciplines and problem areas.

Systems Problem Solving is the part of system science that operationally describes system problems to develop a methodology for solving them. Given a specific system problem, its characteristics can be extracted and, by means of abstraction, translated into a general problem. Then, the problem can be solved using the GSPS framework. A solution of the general system is obtained that can be interpreted to obtain a solution to the specific problem. Figure 2-2 illustrates this process.

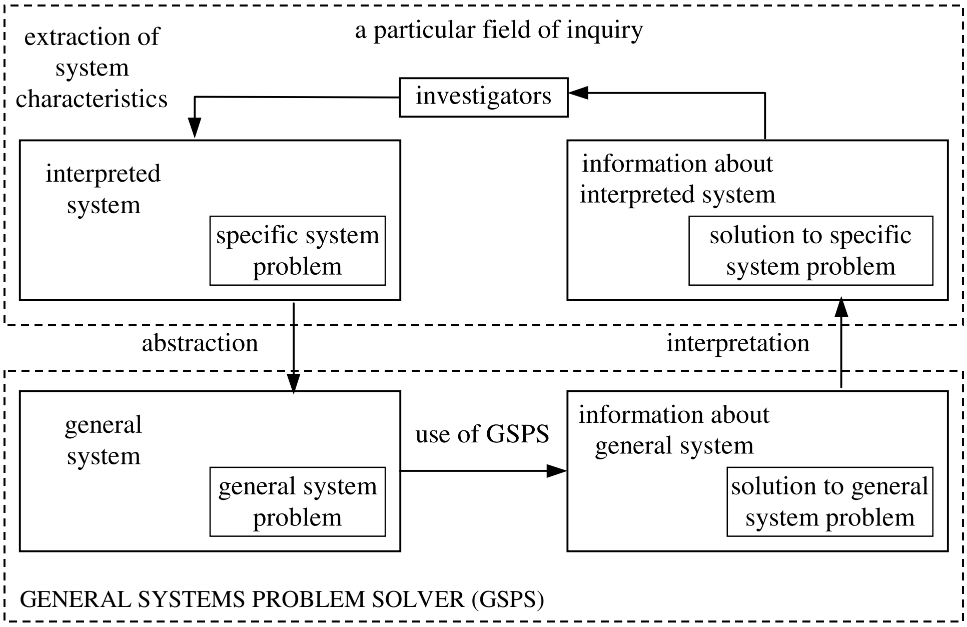

extraction of system characteristics
a particular field of inquiry
investigators
interpreted system
specific system problem
information about interpreted system
solution to specific system problem
abstraction
interpretation
general system
general system problem
use of GSPS
information about general system
solution to general system problem
GENERAL SYSTEMS PROBLEM SOLVER (GSPS)

Figure 2-2 The role of GSPS.

The structure of GSPS is a hierarchy of *epistemological levels of systems* in which different levels of knowledge are represented. This hierarchy is derived from simple ancient notions: there exists an environment where an interaction between an observer (investigator) and the observed object (investigated system) holds. There are basically five epistemological levels: source, data, generative, structure, and meta systems. In Klir's notation they are labelled from level zero to level four, respectively. Figure 2-3 offers a simplified view of the epistemological levels.

At the lowest level of the epistemological ladder, denoted as level 0 or source system, "a system is what is distinguished as a system by the investigator," quoting Klir's words. At this level, the system is a potential source of empirical data. The attributes of the system that the investigator wants to observe are expressed by means of a set of variables. Those variables can be partitioned into basic and support variables (for instance, time or space). Basic variables can be further divided into input, output, and control variables. So the input variable states can be seen as the conditions that affect output variables. At this point, the investigator knows if the variables are discrete, continuous, crisp or fuzzy. As the ladder of epistemological levels is climbed up, more and more knowledge regarding the variables of the associated source system is gained. A higher-level system embraces all knowledge corresponding to the lower levels, plus additional knowledge not available at the lower levels. Therefore, the source system is included in all the higher levels.

LEVEL 4	META SYSTEMS Relations between relations below
LEVEL 3	STRUCTURE SYSTEMS Relation between models below
LEVEL 2	GENERATIVE SYSTEMS Models that generate data below
LEVEL 1	DATA SYSTEMS Observations described in language below
LEVEL 0	SOURCE SYSTEMS Data description language

Figure 2-3 Klir's epistemological levels of systems.

At level one, data systems are encountered. They include the previous level, the source system, and some additional knowledge, comprising the trajectories of all of its variables. Thus it is a source system supplied with data. The variable trajectories are obtained by observations from the system. Of course, the term *system* is fairly abstract. A system can be a physical entity, but this is not always the case. A *model* of a *system* satisfies all the characteristics of being itself a *system*, and therefore qualifies as a *system*. If the system is a model, then observations from the system are simulation results.

The next higher level comprises the generative systems, also named behaviour systems. Knowledge about the support-invariant (in our case, time-invariant) relationships existing among the considered variables is added at this level. These relationships can be used as the laws governing the underlying system, from which new states of the variables may be generated. The qualitative modelling and simulating engines of the FIR methodology are located at this level.

Level three contains structure systems. A structure system is defined as a set of generative systems that, in turn, can be viewed as subsystems of a global or general system. At this level, knowledge about the causal relations between those subsystems is added. Reconstruction analysis, a methodology that will be briefly explained later, lies at this epistemological level.

Finally at level four, there exist the meta-systems. They are composed of a set of structure systems, together with a set of meta-rules describing the relationships between those systems.

The methodology used in this dissertation, Fuzzy Inductive Reasoning, is based on the mentioned epistemological levels. Each one of its modules has a direct correspondence with the first three epistemological levels, so FIR is a level three GSPS-based tool.

2.4 Recoding process

The process of discretising continuous trajectories into discrete episodes is called recoding in the FIR context. The operation of recoding is performed by means of the first module of the qualitative modelling methodology, the so-called Fuzzification Module. It is the task of this module to convert quantitative data gathered from the system to their qualitative counterpart. In this process, each quantitative data point is mapped into a qualitative triple, containing class, fuzzy membership, and side values. Data are usually recoded into an odd number of classes using equal frequency partitioning to determine the landmarks between neighbouring classes. The fuzzy membership function is by default a bell-shaped Gaussian curve that assumes a maximum value of 1.0 at the centre between two landmarks, and a value of 0.5 at each of the landmarks. The side value describes whether the real data point lies to the left (side = -1), at the centre (side = 0), or to the right (side = 1) of the maximum of the membership function governing the chosen class.

Suppose a measured physical variable *V*, such as the temperature of a nuclear reactor, that varies, under normal operation conditions, between a lower and an upper bound, $[V_{min}, V_{max}]$. The range of possible variation of this variable has to be divided into different regions, called, in inductive reasoning, *levels* or *classes*. The values representing such regions could be either symbolic (for example, 'too low', 'normal', 'too high', denoting three different regions) or numeric (for example '1', '2', '3', denoting the same regions as above). In the current FIR implementation, SAPS-II, an integer numeric representation for the distinct classes has been adopted to simplify the task of implementing the tool using a numeric software package such as Matlab [Mathworks, 1997]. At this point, an obvious question arises: how many classes should be selected for each of the system variables?

From statistical considerations, when performing any type of cluster analysis, it is known that each legal discrete state should be recorded at least five times [Law and Kelton, 1990]. Here, the term *state* refers to a combination of legal classes of all the variables involved in the studied system. Thus, there exists a relation between the possible number of legal states and the number of data points that are required to base the modelling effort upon:

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

where n_{rec} denotes the number of observed states, n_{leg} denotes the total number of different legal states, i is an index that loops over all variables, and k_i represents the number of classes into which the i^{th} variable is recoded. If each of the system variables is to be recoded into the same number of classes the previous equation reduces to the expression:

$$n_{rec} \geq 5 \cdot \left(n_{lev}\right)^{n_{\mathrm{var}}}$$

where n_{var} is the number of variables and n_{lev} is the number of classes or levels each variable is recoded into. The number of variables is usually given by the application, and the number of recordings is frequently predetermined, because the modeller and the experimenter may be two different people. This is a normal situation when dealing with complex systems. In such a case, the optimum number of classes into which the variables are to be recoded can be computed as:

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

If the number of observed states is not predetermined, the best option is to consult with a human expert in order to determine a meaningful number of levels for a given variable.

Taking into account symmetry considerations, an odd number rather than an even number of levels may be preferred. Consider for example the three levels 'too low', 'normal' and 'too high'. By using an odd number of classes, the abnormal classes can be grouped symmetrically around the normal class.

The number of classes into which the variables are recoded determines the *expressiveness* and the *predictiveness* of the qualitative model. The expressiveness of a qualitative model is a measure of the information content that the model provides. The predictiveness of a qualitative model measures its forecasting power, i.e., the period of time over which the model can be used to predict the future behaviour of the system with an acceptable fidelity.

A compromise between predictiveness and expressiveness must be reached. If every variable is recoded into only one class, the qualitative model will exhibit a unique legal state. Under this condition, the model will be able to predict the future of the system indefinitely, but the predictions made are useless since they contain no information about the system whatsoever. This model would have an infinite predictiveness but zero expressiveness. On the other hand, if every variable is recoded into a thousand classes, the system will exhibit an enormous number of legal states. In this case, the resolution of the qualitative model is excellent because each state contains a large amount of useful information about the real system. Yet, its predictiveness will surely be very poor, unless an extremely large data-base of observations is available.

As confirmed by several practical applications [de Albornoz and Cellier; 1993a, 1993b; Cellier, 1991b; Mirats and Huber, 1999], using three or five classes is optimal for most purposes.

Once the number of classes has been decided, the *landmarks* that separate neighbouring regions must be chosen. The best option usually is to consult with a human expert. Sometimes this may not be possible, so another approach needs to be taken. If the observed data is limited, as it is usually the case, then it is preferable to maximise the expressiveness of the model. The expressiveness will be maximised if each level is observed equally often. Hence a way to find the landmarks between classes is to sort the observed trajectories into ascending order, cut the resulting vector into n_{lev} segments of equal length, identify each segment as one class, and choose the landmarks anywhere between the largest value of the class to the left and the smallest value of the class to the right.

To this point, the number of classes as well as the landmarks that separate those classes have been decided. Now, in order to convert a quantitative value into a qualitative triple, the membership function of each class has to be chosen. Among the different forms of membership functions that have been implemented in the SAPS platform, the bell-shaped or Gaussian membership function is the one most commonly used. This function can be mathematically expressed by the following equation

$$Memb_i = \exp\left(-\tau_i \cdot (V - \mu_i)^2\right)$$

where V is the continuous variable to be fuzzified, μ_i is the algebraic mean between two neighbouring landmarks, and τ_i is determined such that the membership function degrades to a value of 0.5 at both of these landmarks.

In Figure 2-4, the process of recoding a value of temperature is shown. In this example, the temperature has been discretised into five classes, 'cold', 'fresh', 'normal', 'warm', and 'hot'. A quantitative value of 23 degrees Centigrade is recoded into the qualitative class 'normal' with a fuzzy membership function value of 0.85, and a side function value of 'right', because the value is to the right of the maximum of the bell-shaped membership function associated with the 'normal' class. In the current implementation of the FIR methodology, classes are represented by integers rather than through linguistic values, and the side is denoted by the set {-1,0,1}.

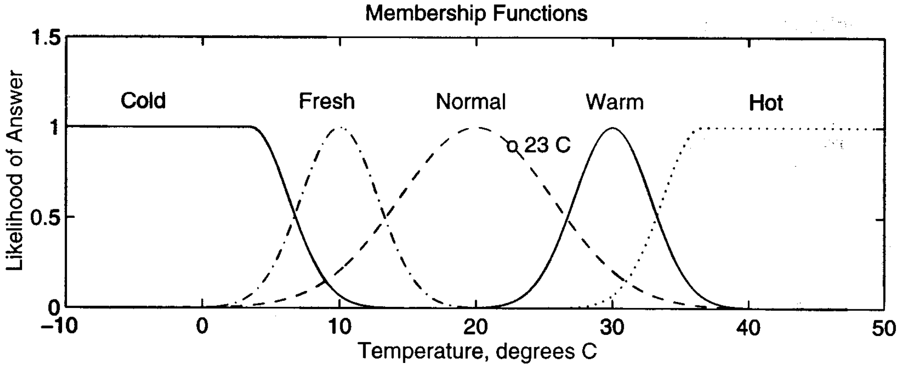
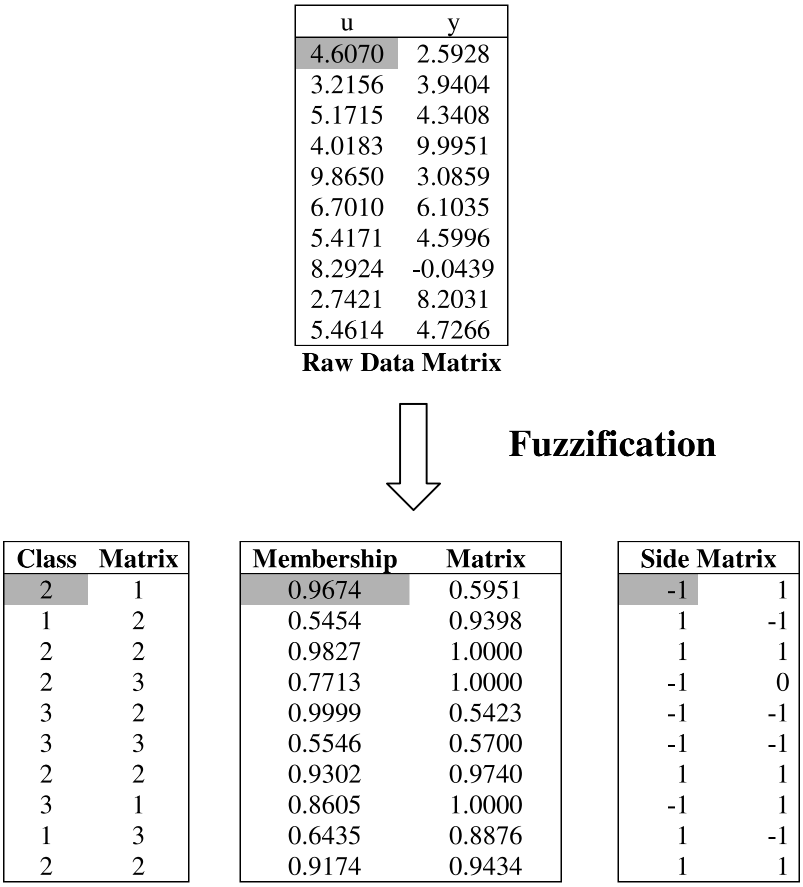


Figure 2-4 Example of recoding a quantitative temperature value.

Synthetic data are used to illustrate the fuzzification process from the raw (real-valued) data matrix to the three qualitative data matrices. The process shown in Figure 2-5 is valid for a one-input one-output system.



u	y
4.6070	2.5928
3.2156	3.9404
5.1715	4.3408
4.0183	9.9951
9.8650	3.0859
6.7010	6.1035
5.4171	4.5996
8.2924	-0.0439
2.7421	8.2031
5.4614	4.7266

Raw Data Matrix

Class	Matrix
2	1
1	2
2	2
2	3
3	2
3	3
2	2
3	1
1	3
2	2

Membership	Matrix
0.9674	0.5951
0.5454	0.9398
0.9827	1.0000
0.7713	1.0000
0.9999	0.5423
0.5546	0.5700
0.9302	0.9740
0.8605	1.0000
0.6435	0.8876
0.9174	0.9434

Side Matrix	
-1	1
1	-1
1	1
-1	0
-1	-1
-1	-1
1	1
-1	1
1	-1
1	1

Figure 2-5 Fuzzification process example.

2.5 Qualitative Modelling Engine

At this point, each of the original real-valued variable trajectories has been recoded individually, i.e., using different landmarks and possibly a different number of classes, into a qualitative episodical behaviour, stored in the three qualitative data matrices, i.e., the class, membership, and side matrices. In those matrices (of the same dimensions as the original raw data matrix), each column represents one of the observed physical variables, and each row represents one time point, i.e., one recorded state. These three matrices form the qualitative data system that feeds the Qualitative Modelling Engine.

The goal of the Qualitative Modelling Engine (QME) is to find relationships among the class values that are as deterministic as possible, trying to discover behavioural patterns among the observations using the information stored in the class value matrix. Such a qualitative relationship is encoded in the form of yet another matrix, called a *mask* in the context of FIR. The optimal mask is found by a process of exhaustive search in the discrete search space of the class values that will be explained shortly.

How can a model of the given system be identified from the episodical behaviour for the purpose of predicting the future behaviour of any given variable?

In the previous section, a continuous trajectory behaviour has been recoded and is available for modelling. The trajectory behaviour comprises a set of measured inputs and outputs of the system that have been recorded. The trajectory behaviour can thus be formed by concatenating, from the right, the set of input trajectories u_i with the set of output trajectories y_i, as shown in the following example containing three inputs, u_1, u_2, and u_3, and two outputs y_1, and y_2.

$$\begin{array}{cc}
\textit{Time} & \begin{matrix} u_1 & u_2 & u_3 & y_1 & y_2 \end{matrix} \\
\begin{matrix} t-(n_{rec}-1)\,\delta t \\ \vdots \\ t-2\delta t \\ t-\delta t \\ t \end{matrix} & \begin{pmatrix} \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}
\end{array}$$

where n_{rec} is the number of records, and δt is the sampling interval.

In order to avoid possible ambiguity, it is defined that the terms 'input' and 'output', when used in this dissertation without further qualifier, shall always refer to the input and output physical variables of the system to be modelled by the qualitative reasoner.

To obtain a model it is desirable to find finite automata relations between the previously recoded variables that are as deterministic as possible. If this can be achieved for each of the output variables of the system, its future behaviour can be predicted by iterating through the state transition matrices. The more deterministic these matrices are, the higher is the probability that the future behaviour will be predicted correctly.

An example of a possible relation between qualitative variables is:

$$y_2(t) = f(u_3(t-2\delta t), u_1(t-\delta t), y_1(t-\delta t), y_2(t-\delta t), u_1(t))$$

where f denotes a qualitative relationship. It does not stand for any explicit formula relating the input and output arguments, but only represents a generic causality relationship, that in the case of the FIR methodology will be encoded in the form of a transition matrix in which the probable input-output patterns are stored.

The QME returns such a qualitative relationship encoded in the form of a matrix called a *mask* in the context of FIR. The mask is the qualitative model of the given system, and it represents the dynamic relationship among its qualitative variables. Masks have the same number of columns as the data system to which they belong, and a certain number of rows, called the *depth* of the mask, related to the number of sampling intervals that the mask covers. Negative elements represent inputs of the qualitative relationship (the so-called *m-inputs* in order to avoid ambiguity), whereas the single positive element represents the mask output (the so-called *m-output*). Notice that the m-inputs can be either inputs or outputs of the system, and they can represent different time instants. Zero elements of the mask denote irrelevant connections. The number of negative elements (corresponding to the m-inputs) is the so-called *mask complexity*. The mask corresponding to the previously introduced qualitative relationship is shown below.

$$\begin{array}{c|ccccc} t/x & u_1 & u_2 & u_3 & y_1 & y_2 \\ \hline t-2\delta t & 0 & 0 & -1 & 0 & 0 \\ t-\delta t & -2 & 0 & 0 & -3 & -4 \\ t & -5 & 0 & 0 & 0 & +1 \end{array}$$

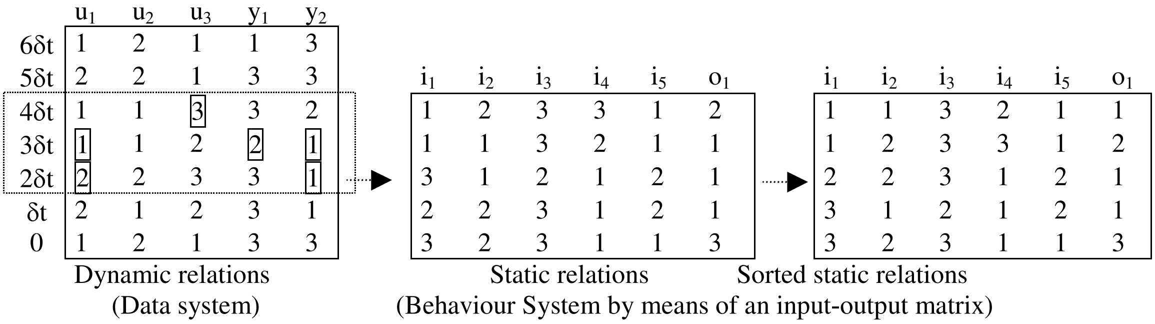
In the above example, there are five m-inputs and one m-output. The sequence in which they are enumerated is unimportant, usually the m-inputs are numbered left to right and top to bottom. The first m-input, let us call it i_1, corresponds to the input variable u_3 two sampling intervals back in time, $u_3(t\text{-}2\delta t)$; the second m-input, i_2, refers to the input variable u_1 one sampling interval in the past, $u_1(t\text{-}\delta t)$; the third m-input, i_3, refers to the output variable y_1 delayed one sampling interval, $y_1(t\text{-}\delta t)$; the fourth m-input, i_4, corresponds to the output variable y_2 one sampling interval in the past, $y_2(t\text{-}\delta t)$; the fifth m-input, i_5, corresponds to the input variable u_1 at the current sampling interval, $u_1(t)$; and, finally, the only m-output, o_1, corresponds to the output variable y_2 at the current sampling interval, $y_2(t)$. This is the variable to be modelled. The given mask has a depth of three, covering a time span of three sampling intervals of the system dynamics, and a complexity of five.

It still has not been discussed how the depth of the mask is chosen. Experience has shown that the following general rule is valid: the mask should cover the largest time constant of the system that is to be captured in the qualitative model. If the physical process that governs the underlying system is known, it is possible to have knowledge about the shortest and the largest time constants of this system, denoted t_s and t_l respectively. The former determines the sampling interval, δt, which cannot be larger than one half of the smallest time constant. The latter provides a minimum for the depth of the mask, i.e., the minimum required time that the qualitative model should cover of the physical system dynamics.

$$\delta t \leq t_s/2 \qquad (depth\text{-}1)\delta t \geq t_l$$

If the physical system is available for experimentation, a Bode diagram can be obtained, and the smallest and largest eigen-frequencies can be determined. From these frequencies, the largest and the shortest time constants can be computed, which, in turn, specify the depth of the mask and the sampling frequency. Yet, it is quite common, when dealing with large-scale systems, that the modeller does not have access to the system for active experimentation. All that may be available are already recorded variable trajectories. In this case, the depth of the mask needs to be determined in different ways. Ideally, the modeller would consult an expert of the studied system about how much time is needed to cover the dynamics of the system, but this is not always possible. Another solution would be to heuristically decide the depth of the mask, based on one's own modelling experience. A more systematic approach advocated in this thesis is to use the depth that maximises the quality of the resulting qualitative model. This approach is explained in subsection 2.5.1 where an alternative algorithm to find a FIR model is proposed.

At this point, the concept of a mask as used in the FIR methodology is clear. The mask can be employed to flatten dynamic relationships into static ones. To this end, the mask is shifted over the matrices that represent the recoded data system; in each mask position, the selected m-inputs and m-output can be extracted from the data system, and can be written next to each other in a static record. Every row of the obtained matrix represents a fuzzy rule. Static records can then be sorted alphabetically and stored in the so-called input-output matrix. Figure 2-6 illustrates this process.



	u1	u2	u3	y1	y2
6δt	1	2	1	1	3
5δt	2	2	1	3	3
4δt	1	1	$\boxed{3}$	3	2
3δt	$\boxed{1}$	1	2	$\boxed{2}$	$\boxed{1}$
2δt	$\boxed{2}$	2	3	3	$\boxed{1}$
δt	2	1	2	3	1
0	1	2	1	3	3

Dynamic relations
(Data system)

i1	i2	i3	i4	i5	o1
1	2	3	3	1	2
1	1	3	2	1	1
3	1	2	1	2	1
2	2	3	1	2	1
3	2	3	1	1	3

Static relations

i1	i2	i3	i4	i5	o1
1	1	3	2	1	1
1	2	3	3	1	2
2	2	3	1	2	1
3	1	2	1	2	1
3	2	3	1	1	3

Sorted static relations

(Behaviour System by means of an input-output matrix)

Figure 2-6 Obtaining static relations by use of a mask.

The obvious question now arises, how the qualitative model (i.e., the mask) of a given system is determined in the FIR methodology? It has been outlined before that the mask representing the most deterministic qualitative relation is to be found among all the possible qualitative models. Thus, a search in the space of potential models must be performed. To this end, two tools are still needed: an adequate search engine ensuring that the most deterministic qualitative relationship is found, and a way to compare different models quantitatively with respect to their merits.

In order to search the model space, FIR makes use of the concept of a *mask candidate matrix*. A mask candidate matrix is an ensemble of all the possible masks. It contains -1 elements in the positions of the potential m-inputs, 0 elements to denote forbidden connections, and a unique +1 element in the position of the m-output. FIR works with multi-input single-output, MISO, systems. If a MIMO system is to be modelled with the FIR methodology, one mask has to be computed for each of the system outputs. A good candidate mask to obtain a predictive model for the output variable y_2 of the previous five variables example might be

$$\begin{array}{c} \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} u_1 & u_2 & u_3 & y_1 & y_2 \end{array} \\ \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 0 & 1 \end{pmatrix} \end{array}$$
$$\textit{Candidate mask}$$

A similar mask candidate matrix can be written for the other output variable. Notice that a mask candidate matrix contains a unique m-output and that usually all connections are allowed except for those that include the other outputs of the system at time *t*. This is done to prevent possible algebraic loops of the type:

$$y_2(t) = f_1(y_1(t))$$
$$y_1(t) = f_2(y_2(t))$$

It has been said that all possible connections are usually permitted. Doing so is only feasible when dealing with low complexity systems, say, for instance, a ten variables system, because of the exponential complexity of the optimal mask search algorithm. This dissertation deals with application of the FIR modelling methodology to complex systems. One way of reducing the search space is to disallow some of the possible connections in the candidate mask matrix. For example, if it were a priori known that, in the above system, the m-output *$y_2(t)$* may not depend on the m-inputs *$u_1(t\text{-}2\delta t)$*, *$u_2(t\text{-}\delta t)$* and *$u_3(t)$*, additional zero elements could be introduced into the mask candidate matrix, thereby reducing the total number of models to be computed. The mask candidate matrix, in the supposed case, would have been

$$\begin{array}{c} \\ t-2\delta t \\ t-\delta t \\ t \end{array} \begin{array}{c} \begin{array}{ccccc} u_1 & u_2 & u_3 & y_1 & y_2 \end{array} \\ \begin{pmatrix} 0 & -1 & -1 & -1 & -1 \\ -1 & 0 & -1 & -1 & -1 \\ -1 & -1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

How this knowledge about the system to be modelled can be gained will be tackled in subsequent sections.

Once an appropriate mask candidate matrix has been formulated, it is provided to the FIR *mexfoptmask* function, which performs an exhaustive search among all the possible qualitative models expressed in the mask candidate matrix.

Each of the possible masks is compared to the others with respect to its potential merit. The index used to compare the masks is an entropy-based measure called the *quality of the mask* in the FIR context. The optimality of the mask is evaluated with respect to the maximisation of its predictive power.

The Shannon entropy measure is used to determine the uncertainty associated with predicting a particular output given any legal input state [Shannon and Weaver, 1978]. This entropy measure relative to one input state *i*, is calculated from the equation:

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

where *p(o|i)* is the conditional probability of a certain m-output state *o* to occur, given an m-input state *i*. The overall entropy of the mask is then computed as the sum:

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

where *p(i)* is the probability of that m-input state to occur. These probabilities are usually not known and must therefore be statistically estimated. This is where the membership value of each recoded data point plays its role. The fuzzy membership associated with the value of a qualitative variable is a measure of confidence. It tells how much confidence there is about the correctness of the assigned class value. When computing the input-output matrix, a confidence value can be assigned to each row. This confidence value is computed as the *joint membership* of all the variables that are associated with the given row of the input-output matrix. Different fuzzy approaches vary in their interpretation of the joint membership value. In FIR, the joint membership of *i* membership functions is defined as the smallest individual membership [Li and Cellier, 1990]:

$$Memb_{jo\,\mathrm{int}} = \bigcap_{\forall i} Memb_i = \inf_{\forall i}\left(Memb_i\right) \overset{def}{=} \min_{\forall i}\left(Memb_i\right)$$

In contrast, some of the other available fuzzy approaches define the joint membership value as the product of the individual membership values.

Joining different memberships can be interpreted as an *intersection* operator, which, in the case of FIR, is a min-operator. Consequently, FIR defines the confidence of a particular observation as:

$$Conf_{obs} \overset{def}{=} \min_{\forall i}\left(Memb_i\right)$$

If the same input-output state is observed more than once, the confidence of the correctness of this state should grow. In FIR, the accumulated confidence of the i^{th} input-output state is defined as the sum of the confidences of its observations:

$$Conf_{accum} = \bigcup_{\forall i} Conf_i = \sup_{\forall i}\left(Conf_i\right) \overset{def}{=} \mathop{sum}_{\forall i}\left(Conf_i\right)$$

Again, different fuzzy techniques interpret the accumulated confidence differently. In some approaches, the accumulated confidence is defined as the largest of the individual confidences. The attentive reader may notice that the accumulated confidence, as defined in FIR, can no longer be interpreted as a probability, since values larger than 1.0 may result.

Accumulating confidences can be interpreted as a *union* operator, which, in the case of FIR, is a sum-operator. Consequently, FIR defines the confidence of an input-output state as:

$$Conf_{i-o\_state}\left(i\right) \overset{def}{=} Conf_{accum}$$

Among the different dialects of fuzzy logic, FIR is quite peculiar, as most of the classical fuzzy approaches use either the min/max pair of operators, or the prod/sum pair of operators. The min/sum pair is rather unusual. The reader may notice that the min-operator leads to a larger intersection than the prod-operator. Similarly, the sum-operator leads to a larger union than the max-operator. FIR therefore uses an optimistic approach to assessing confidence values. There is no deep theory behind this choice. All combinations were tried out when the methodology was first implemented, and the min/sum pair of operators gave consistently the best results.

With these confidence values, the conditional probabilities needed for the computation of the mask entropy can be estimated. The conditional probability of a certain m-output state *o* to occur, given an m-input state *i*, *p(o|i)*, may be estimated dividing the confidence of the given input-output state, denoted as $Conf_{i\text{-}o\_state}$, by the sum of the confidence values of all input-output states that share the same m-input state. Hence,

$$p(o \mid i) \approx \frac{Conf_{i-o\_state}(i,o)}{\sum_{o} Conf_{i-o\_state}(i,o)}$$

where *i* denotes an m-input state vector, and *o* denotes the m-output state.

In order to compute the overall mask entropy, H_m, the probability of an m-input state to occur, *p(i)*, needs to be estimated. In FIR, this probability is approximated by means of the relative observation frequency for that m-input state:

$$p(i) \approx \frac{\sum_{o} p(o \mid i)}{\sum_{\forall o \forall i} p(o \mid i)}$$

Once these probabilities have been estimated, a normalised overall entropy reduction index, H_r, may be defined as:

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

where H_{max} is the highest possible entropy, obtained when all the observed states have the same probability to occur. Zero entropy would be found for relationships that are totally deterministic.

The overall entropy reduction index, H_r, is a real number in the range [0,1], where higher values usually denote better predictive power. The best mask is defined as the one with the highest H_r value among the set of possible masks.

The use of Shannon entropy as a confidence measure is a questionable undertaking on theoretical grounds, since the Shannon entropy was derived only for probabilistic measures. Other scientists prefer using other performance indices [Klir, 1989; Shafer, 1976] rather that Shannon entropy, which have been derived in the context of the particular measure chosen. However, from a practical point of view, a large number of simulation experiments have shown that the Shannon entropy index works satisfactorily also in the context of fuzzy measures, when the confidences are re-normalised as conditional probabilities, as shown above.

Yet, while H_r indeed optimises the determinism of the input-output relationships, there is still a problem with using H_r as the sole measure of mask quality. The entropy index, H_r, depends on the complexity of the mask. As the mask complexity increases, the input-output relations become more and more deterministic. The reason for this observation is the following. With growing mask complexity, more and more possible m-input states exist. Since the total number of observations, n_{rec}, remains constant, the observation frequencies of the observed m-input states will become smaller and smaller, until the situation arises that each observed m-input state has been observed exactly once, whereas many legal m-input states have never been observed at all. At this point, the model becomes totally deterministic, and H_r assumes a value of 1. Yet, the predictiveness of such a model will be very poor, as already the next predicted state will probably never have been observed before, which invariably makes further predictions impossible.

One possible way to solve this problem is to introduce a weighting factor into the quality measure that accounts for the number of times that a state has previously been observed. It was mentioned earlier that, from a statistical point of view, every state should be observed at least five times [Law and Kelton, 1990]. Thus, the following *observation ratio*, O_r, can be introduced as an additional contributor to the overall quality measure:

$$O_r = \frac{5 \cdot n_{5x} + 4 \cdot n_{4x} + 3 \cdot n_{3x} + 2 \cdot n_{2x} + n_{1x}}{5 \cdot n_{leg}}$$

where:

n_{leg} = number of legal m-input states
n_{1x} = number of m-input states observed only once
n_{2x} = number of m-input states observed twice
n_{3x} = number of m-input states observed thrice
n_{4x} = number of m-input states observed four times
n_{5x} = number of m-input states observed five or more times

When every m-input state has been observed at least five times, O_r is equal to 1. Conversely, if no m-input states have been observed at all, i.e., no data are available, O_r is equal to 0.

Finally, the overall quality of a mask can be defined as the product of its uncertainty reduction measure, H_r, and its observation ratio, O_r:

$$Q_m = H_r \cdot O_r$$

In order to clarify this process and show how the probabilities and the intermediate matrices involved in the computation of the quality of a possible mask are obtained, the five variables example that has been utilised all throughout this section may serve for illustration.

Assume that the following qualitative data model, expressed by means of a multi-valued class matrix and a real-valued membership matrix, is obtained after the recoding process.

$$class = \begin{pmatrix} 1 & 1 & 3 & 1 & 2 \\ 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 & 2 \\ 2 & 1 & 2 & 3 & 2 \\ 2 & 1 & 1 & 2 & 3 \\ 2 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 3 & 1 \\ 2 & 2 & 1 & 1 & 3 \\ 2 & 3 & 1 & 3 & 1 \\ 2 & 2 & 3 & 2 & 1 \end{pmatrix} \qquad Membership = \begin{pmatrix} 1.0 & 0.9904 & 0.5 & 0.72 & 0.5 \\ 1.0 & 0.9905 & 0.9631 & 0.65 & 0.9709 \\ 1.0 & 0.9904 & 0.9761 & 0.55 & 0.9986 \\ 1.0 & 0.9904 & 0.9992 & 0.88 & 0.999 \\ 1.0 & 0.552 & 1.0 & 0.99 & 1.0 \\ 1.0 & 0.5798 & 0.6045 & 0.87 & 0.6711 \\ 1.0 & 0.5172 & 0.6123 & 0.622 & 0.5224 \\ 1.0 & 0.6186 & 0.5236 & 0.777 & 0.5255 \\ 1.0 & 0.5 & 0.5037 & 0.6555 & 0.5049 \\ 1.0 & 0.964 & 0.5033 & 0.567 & 0.5 \end{pmatrix}$$

In the above example, it is assumed that the first variable (recoded in the first column) was originally a binary variable, whereas the other four variables were initially real valued. Consequently, the first variable was mapped (recoded) into the discrete states ‘1’ and ‘2’ directly, and its associated membership values are 1.0 throughout.

Suppose that the mask to be analysed is

$$possible\,mask = \begin{pmatrix} -1 & 0 & 0 & 0 & -2 \\ 0 & -3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

When flattening this mask through the class and membership matrices, the following input-output matrix and confidence vector is obtained. The process followed is the one shown in Figure 2-6.

$$io = \begin{pmatrix} 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 3 \\ 2 & 2 & 1 & 1 \\ 2 & 3 & 3 & 1 \\ 2 & 1 & 1 & 3 \\ 2 & 1 & 2 & 1 \\ 2 & 3 & 3 & 1 \end{pmatrix} \qquad conf = \begin{pmatrix} 0.5 \\ 0.9709 \\ 0.9904 \\ 0.552 \\ 0.5224 \\ 0.5172 \\ 0.5049 \\ 0.5 \end{pmatrix}$$

The basic behaviour of the input-output model can now be computed. It is defined as the ordered set of all observed different states, together with a measure of accumulated confidence of each state.

$$beh = \begin{pmatrix} 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 3 \\ 2 & 1 & 1 & 3 \\ 2 & 1 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 3 & 3 & 1 \end{pmatrix} \qquad conf\_beh = \begin{pmatrix} 1.4709 \\ 0.9904 \\ 0.5172 \\ 0.5049 \\ 0.5520 \\ 1.0224 \end{pmatrix}$$

Now, from the cumulative confidence vector, that is no longer a probability, the so-called state transition matrix can be computed:

$$st = \begin{array}{c} in/out \\ '121' \\ '211' \\ '212' \\ '221' \\ '233' \end{array} \begin{array}{c} \begin{array}{ccc} '1' & '2' & '3' \end{array} \\ \begin{pmatrix} 0.0 & 1.4709 & 0.9904 \\ 0.0 & 0.0 & 0.5172 \\ 0.5049 & 0.0 & 0.0 \\ 0.5520 & 0.0 & 0.0 \\ 1.0224 & 0.0 & 0.0 \end{pmatrix} \end{array}$$

It shows, to the left of the matrix, the m-input state vectors observed, and at the top, the possible classes of the m-output; in the matrix itself, the confidence values of the corresponding input-output states are recorded. In the example at hand, the state transition matrix is highly deterministic due to the low number of observations considered.

The total input confidence is obtained by adding the individual confidences of all occurrences of the same m-input state:

$$tot\_iconf = \begin{pmatrix} 2.4613 \\ 0.5172 \\ 0.5049 \\ 0.5520 \\ 1.0224 \end{pmatrix}$$

In order to compute the quality of the considered mask, and hence, using the Shannon entropy index, the accumulated confidences must be converted back to values that can be interpreted as probabilities. For this reason the row sums of the state transition matrix are normalised to 1.0, and so it is the total m-input confidence vector:

$$st = \begin{array}{c} in/out \\ '121' \\ '211' \\ '212' \\ '221' \\ '233' \end{array} \begin{array}{c} \begin{array}{ccc} '1' & '2' & '3' \end{array} \\ \begin{pmatrix} 0.0 & 0.5976 & 0.4024 \\ 0.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \end{pmatrix} \end{array} \qquad tot\_iconf = \begin{pmatrix} 0.4866 \\ 0.1022 \\ 0.0998 \\ 0.1091 \\ 0.2022 \end{pmatrix}$$

With the information at hand it is now possible to compute the overall entropy of the mask, H_m, and, afterwards, the entropy reduction index, H_r. Applying the formulae presented throughout this section:

$$\begin{aligned} -H_m = {} & 0.4866 \cdot [0.4024 \cdot \log_2(0.4024) + 0.5976 \cdot \log_2(0.5976)] + \\ & 0.1022 \cdot [1.0 \cdot \log_2(1.0)] + \\ & 0.0998 \cdot [1.0 \cdot \log_2(1.0)] + \\ & 0.1091 \cdot [1.0 \cdot \log_2(1.0)] + \\ & 0.2022 \cdot [1.0 \cdot \log_2(1.0)] = \cdots = -0.4731 \end{aligned}$$

Finally, the value of the entropy reduction measure can be computed:

$$H_r = 1.0 - \frac{H_m}{H_{max}} = 1.0 - \frac{0.4731}{1.585} = 0.7015$$

where the maximum entropy is obtained when all legal states have equal probability, i.e., when all of the state transition matrix elements have the same value, which in the current example would be 0.3333.

2.5.1 New algorithm to obtain a qualitative model of a complex system

As explained in the previous sections, the modelling engine of FIR determines a so-called *optimal mask* that indicates, which variables best explain any given output, and how much time delay these variables should have relative to the chosen output. Unfortunately, any algorithm that can find the optimal mask is necessarily of exponential complexity, i.e., the number of masks to be visited during the search for the optimal mask grows exponentially with the number of available input variables and with the allowed depth of the mask. For this reason, sub-optimal search algorithms are necessary for dealing with large-scale systems.

Sub-optimal mask search strategies were studied before. [Nebot and Jerez, 1997] analyse several variants of hill-climbing algorithms. Hill-climbing algorithms are of polynomial complexity, but often may end up with a sub-optimal mask of significantly inferior quality. [Jerez and Nebot, 1997] analyse the use of genetic algorithms as sub-optimal mask search strategies. Unfortunately, genetic algorithms, while sometimes working surprisingly well, cannot be guaranteed to converge in polynomial time. [de Albornoz, 1996] discusses a statistical technique based on cross-correlation functions. His algorithm converges in polynomial time, but only looks at linear relationships between variables, and therefore often finds a sub-optimal mask of vastly inferior quality.

In this sub-section, a new approach for reducing the model search space of FIR is proposed that can be viewed as a modification of the already existing hill-climbing methods. The presented approach reduces the model search space by using the recoded (qualitative) data proposing increasing depth mask candidates to FIR. This new algorithm can deal with previously unmanageable large-scale MISO systems. The algorithm selects those qualitative relations that are believed to contain most information about the system under study. As with classical FIR, the method can be extended for MIMO systems by applying it to each of the output variables of the system separately.

When using FIR to qualitatively model a system, the maximum *depth* and *complexity* of the mask must be chosen. The maximum mask depth is defined as the number of rows of the mask candidate matrix[1]. It determines the largest time constant of the system to be captured by the model. However, since only measurement data are available to base the model upon, it may not always be easy to estimate the largest relevant time constant of the

[1] In some earlier publications, the mask depth was defined as the number of time intervals covered by the mask, which differs from the definition used here by 1.

system. The proposed sub-optimal search algorithm will automatically determine an optimal value for the mask depth *d*.

The *complexity*, *c*, of the mask is the number of mask inputs, usually called *m-inputs*, used in the model[2]. The mask complexity represents a compromise between the *specificity* and the *predictability* of the model [Cellier, 1991a]. A low-complexity mask, i.e., a mask with a small number of *m-inputs*, makes it easy to make predictions, yet the predictions obtained in this way are not very specific. On the other hand, a complex model, if applicable, can make highly specific predictions; however, there may not be enough evidence gathered from the training data to justify such a prediction. Based on the previous experiences gained with use of the FIR methodology [Cellier *et al.*, 1996; de Albornoz, 1996], a maximum complexity between 4 and 5 offers a good specificity compromise.

The optimal mask search algorithm employed by FIR starts with evaluating the masks of lowest complexity, then proceeds by incrementing the complexity, until the maximum allowed mask complexity, *m*, has been reached. Even in the exhaustive search algorithm, the maximum complexity is usually specified in order to limit the search space. This compromise is justified, because the algorithm that computes the mask quality punishes high mask complexity sufficiently to make it rather unlikely that a mask of complexity higher than 5 would ever become the optimal mask. Because of this limitation, even the exhaustive search algorithm is strictly speaking of polynomial complexity. The number of masks to be visited is of the order of $(n \cdot d - 1)^m$, i.e., it is polynomial in the number of inputs, *n*, and the mask depth, *d*.

The sub-optimal search algorithm proposed in this section reduces this number further by reducing the number of '-1' elements in the mask candidate matrix. It starts with a mask of depth $d=1$. Since at this point in time, no information is known about the system yet, all potential inputs are set to -1 in the mask candidate matrix. The optimal (exhaustive) mask search algorithm is then employed to evaluate the quality of each mask of complexity c □ m. Masks are grouped in sets of equal complexity. For each complexity, *c*, the mask of highest quality is found. Its value is Q_{best}. The relative quality of any one of these masks is defined as $Q_{rel} = Q / Q_{best}$. All masks with a relative quality of $Q_{rel} > s$ are considered *good masks*[3]. All good masks of a given complexity are then investigated. If a given input is being used by at least *t* % of all good masks of a given complexity, it is considered a *significant input*[4].

In the next step of the algorithm, all significant inputs of every complexity are marked by '-1' elements. All insignificant inputs are marked by '0' elements. The depth, *d*, is now increased by one. Since nothing is known about the significance of inputs at the new top row, all elements of that row are marked as '-1' elements. The exhaustive search is now repeated with the new mask candidate matrix.

[2] In some earlier publications, the complexity of a mask was defined as the number of non-zero elements of the mask, which differs from the definition used here by 1.

[3] In the current implementation of the algorithm, $s = 0.975$.

[4] In the current implementation of the algorithm, $t = 10$ %.

For each mask candidate matrix, the optimal mask is determined. Its quality is Q_{opt}. It is the largest quality of all Q_{best} values of all considered complexities. The algorithm continues, until the Q_{opt} value no longer increases when d is incremented.

The algorithm that has been described in this section, like all sub-optimal search methods, is a *heuristic* search technique. There is no guarantee that the truly optimal mask will be found in this way. However, the algorithm is based on much experience and a lot of common sense. A significant input indicates that this input is useful in explaining the desired output. It is very likely that the optimal mask makes use of significant inputs only, or at least makes use primarily of significant inputs. Hence it is very likely that the algorithm will either find the optimal mask itself or at least one of insignificantly lower quality.

The proposed termination criterion is also meaningful, but it may sometimes fail. Some systems exhibit cyclic behaviour. For example, the water demand of the city of Barcelona has a strong weekly cycle. Cyclic behaviour can easily be detected by looking at the autocorrelation of each observed variable. If there is a strong cyclic behaviour, the mask depth should be chosen to cover at least one full cycle, i.e., the termination criterion may need to be modified accordingly, by specifying that the algorithm may not terminate until the mask covers at least one full cycle.

The computational complexity of FIR is exponential, so it is faster to compute N simple models than computing a unique complex model. The approach presented in this section exploits this characteristic to reduce the model search space. Results of applying this algorithm to a real system, a garbage incinerator process, can be found into Appendix II.1. As shown in the mentioned appendix, the computation reduction achieved is significant, and the presented approach is a good candidate to qualitatively model complex systems.

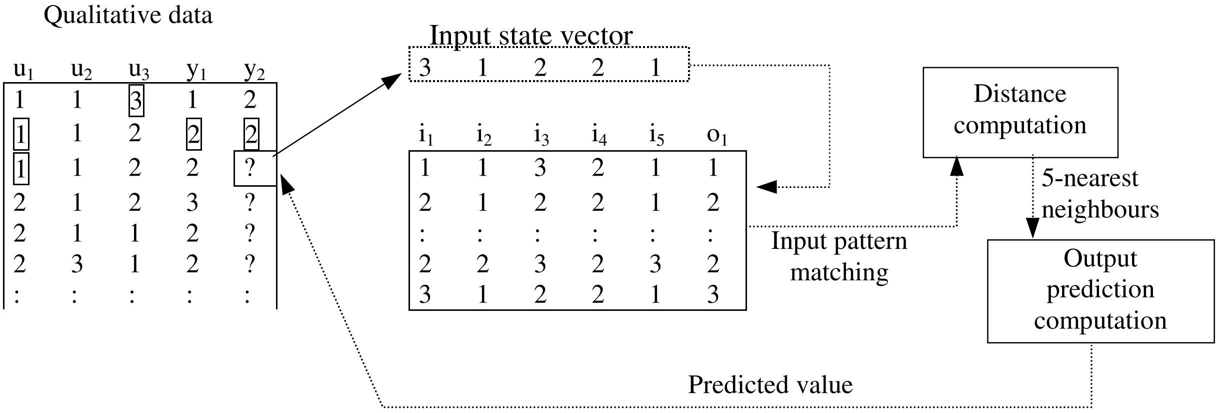
2.6 Qualitative simulation engine

Given a FIR qualitative model described by means of a mask and three behaviour matrices (constituting a set of fuzzy rules) for every m-output variable to be modelled, the task of the *Qualitative Simulation Engine* (QSE), is to forecast a value for each of the chosen variables. Fuzzy simulation extrapolates the m-output variable across time using the interpolated values of previous occurrences of similar behavioural patterns, i.e., class, membership, and side values of the selected m-output are inferred from the class, membership, and side values of the considered m-inputs. This inference process can be performed efficiently, since the search for optimal inference rules is limited to a discrete search space.

So, once a qualitative model, i.e., a mask, has been determined, it can be applied to the given qualitative data model, as specified by the class, membership, and side matrices, resulting in the class, membership, and side input-output matrices. The class input-output matrix, cf. the process shown in Figure 2-6, contains functional relationships within single rows that can be sorted in alphanumerical order. The membership and side input-output matrices are sorted accordingly. This operation results in another three matrices called the *behaviour matrices* of the system. The class behaviour matrix is a finite state machine that shows, for each m-input state, which m-output is most likely to be observed.

Predicting a new value from the gained knowledge is a straightforward procedure. Figure 2-7 illustrates the fuzzy prediction algorithm. The illustration assumes the same mask that had been used in Figure 2-6.

The procedure is as follows. At each sampling instant, the mask is shifted one step down along the class matrix of the fuzzified values of the variables. Classes of the m-inputs are read out from the mask, and a so-called input state vector is obtained. Then the class behaviour matrix is searched for coincidences on that input state vector and the associated membership and side functions are compared with the ones of the input state vector. The five nearest neighbours algorithm is then used to compute the class, membership, and side values of the m-output. Each m-output is an averaged value from the m-outputs associated with the five nearest neighbours.



u_1	u_2	u_3	y_1	y_2
1	1	3	1	2
1	1	2	2	2
1	1	2	2	?
2	1	2	3	?
2	1	1	2	?
2	3	1	2	?
:	:	:	:	:

i_1	i_2	i_3	i_4	i_5	o_1
1	1	3	2	1	1
2	1	2	2	1	2
:	:	:	:	:	:
2	2	3	2	3	2
3	1	2	2	1	3

Figure 2-7 Fuzzy forecasting process

Hence the fuzzy forecasting process predicts an entire qualitative triple from which a quantitative value can be regenerated. In the prediction process, 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 class behaviour matrix. For this purpose, a cheap approximation of the regenerated continuous signal is computed, i.e., a normalisation function, for every element of the new input state.

There exist different ways to compute this normalisation function for every input state. In this dissertation, the method described in [Mugica, 1995] has been adopted. For a thorough understanding of the process, let us recall the formula used in Section 2.4 where the recoding process was explained:

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

where x is the continuous variable to be fuzzified, μ_i is the algebraic mean between two neighbouring landmarks, and τ_i is determined such that the membership function degrades to a value of 0.5 at both of these landmarks (see Figure 2-4).

Each Gaussian is re-normalize separately to the range [0,1]. The process is illustrated in Figure 2-8. The shape factor τ needs to be determined such that the membership *Memb*$_i$ is reduced to 0.5 at the landmarks, thus:

$$0.5 = \exp(-\tau\cdot(0.5)^2) \qquad \text{i.e.,} \qquad \tau = -4\cdot\ln(0.5)$$

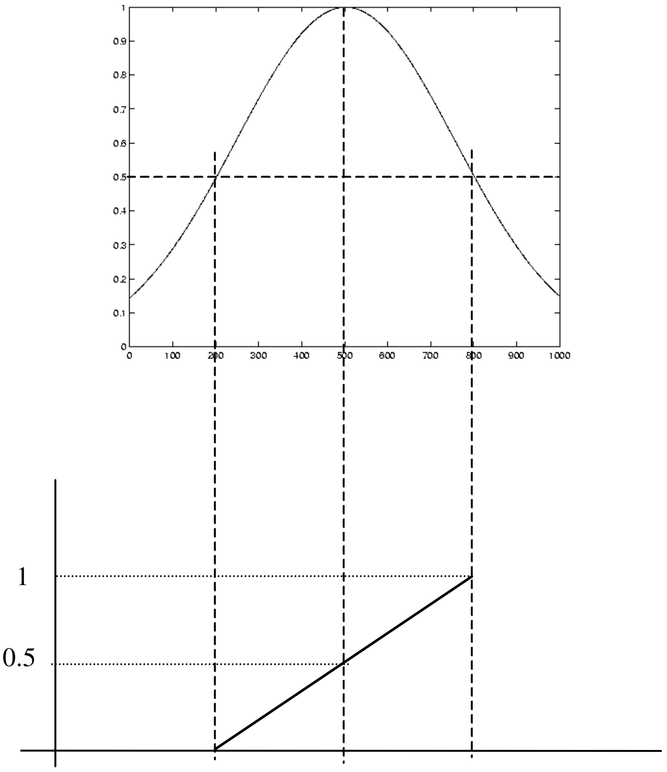


Figure 2-8 Re-normalisation of a Gaussian to compute a pseudo-regeneration value

Consequently:

$$Memb_i = \exp(4\cdot\ln(0.5)\cdot(p\text{-}\mu)^2)$$

where *x* has been renamed *p*, denoting a *pseudo-regeneration* value instead of the true value of the original real-valued variable (denoted by *x*). Solving this equation for *p*, we find:

$$p_i = side_i \cdot B \cdot \sqrt{\ln\left(memb_i\right)} + 0.5$$

where $B = (4 \cdot \ln 0.5)^{-1/2}$.

The *pseudo-regeneration* uses different formulae depending on the particular class to be re-normalized. The above formula is valid for the interior class only. The bottom and top classes use different formulae, since the membership functions associated with these classes are semi-Gaussians. For the leftmost (bottom) class, the following formula results:

$$p_i = C \cdot \sqrt{\ln(memb_i)}$$

and for the rightmost (top) class, the formula:

$$p_i = 1 - C \cdot \sqrt{\ln(memb_i)}$$

is found, where $C = (\ln 0.5)^{-\frac{1}{2}}$. Irrespective of whether the original real-valued signal was small or large, the corresponding p_i value always lies in the interval [0.0, 1.0].

The p_i values can be used to represent the relative magnitude of a particular qualitative triple. The p_i values corresponding to each of the variables that form an input state vector are stored next to each other to form another vector:

$$\mathbf{p} = \lfloor p_1, p_2, \cdots, p_j \rfloor$$

where $\boldsymbol{p}$ is called the *norm image* of the input state vector. In the given example, *j* m-inputs have been assumed to exist.

Norm images are also computed for each of the coincidences of the same input state vector in the class behaviour matrix, so a set of $\boldsymbol{p}_k$ vectors is obtained. Every $\boldsymbol{p}_k$ vector is slightly different, since for each recorded input, the class values are the same, but not their membership and side function values.

Using the vector $\boldsymbol{p}$ of the input state vector and the $\boldsymbol{p}_k$ vectors of the coincidence vectors, the distances of these coincidence vectors from the current norm image are then computed as the L_2 norms of the difference vectors, i.e., a measure of distance between the actual input pattern and the previously observed input states that match the current input pattern is obtained:

$$d_k = \|\mathbf{p} - \mathbf{p}_k\|_2 = \sqrt{\sum_{i=1}^{N} (p_i - p_{ik})}$$

The five previous recordings with the smallest distances, d_k, are identified and are used to predict the new m-output state. These five input/output vectors are called the *five nearest neighbours*. It may happen that the experience database contains less than five coincidence vectors. In that case, all coincidence vectors form part of the set of nearest neighbours, and the subsequently explained interpolation formulae are adjusted accordingly.

Each of the nearest neighbours is weighted so its contribution to the estimation of the new m-output state is a function of its proximity. There are two different weighting functions depending on whether a zero distance neighbour is found or not. If none of the five nearest neighbours has a zero d_k distance, the formula used in this dissertation is:

$$\omega_{abs\,k} = \frac{\left(d_{max}^2 - d_k^2\right)}{d_{max} \cdot d_k}$$

whereas the modified formula:

$$\omega_{abs\,k} = \begin{cases} 0.0; & d_k \neq 0.0 \\ 1.0; & d_k = 0.0 \end{cases}$$

is used if one or more of the neighbours has a zero distance with respect to the input state vector.

From these absolute weights[5], relative weights are computed:

$$\omega_{rel\,k} = \frac{\omega_{abs\,k}}{S_\omega}$$

where $S_\omega = \sum_{\forall k} \omega_{abs\,k}$. So the relative weights are real numbers in the range [0.0, 1.0] and they always add up to 1.0. Now it is possible to compute the new m-output state values. This is done using the following formula:

$$State_{out\_new} = \sum \omega_{rel\,k} \cdot \left(class_{out\,k} + p_{i\,out\,k}\right)^{@}$$

where $class_{out\ k}$ and $p_{i\ out\ k}$ are the class and the normalised membership values of each of the five nearest neighbours. The class value of the new m-output is computed as the integer part of the $State_{out\_new}$ value:

$$Class_{out\,new} = Integer(State_{out\_new})$$

and the real part of the value is the normalised membership value:

$$p_{i\,out\_new} = State_{out\_new} - Class_{out\,new}$$

The membership and side values of the new m-output state can be obtained by means of inverting the normalisation function.

[5] Different versions of FIR use different formulae for the computation of the absolute weights. In this dissertation, the formula first introduced in [Pan, 1994] has been used. Two alternate formulae, one based on a different distance measure, the other based on a similarity measure, have been analysed in [López, 1999].

[@] Note that this formulation can lead to an error on the new state class prediction when the five nearest neighbours are the same and have a membership value equal to 1.0. In this case, even a non-existing class may be predicted. In this dissertation the error has been fixed, the class of the new predicted m-output state is computed as $Class_{out\,new} = Integer(State_{out\_new}) - 1$

2.6.1 Variable acceptability interval

The new concept of *variable acceptability interval* or, *envelopes*, in short, is introduced in this section in the context of the FIR methodology and will be used in Appendix II.2 to detect structural changes in a simulated aircraft model. Apart from the predicted signal, two more signals are computed so as to get for each forecast value an interval of acceptability of the real trajectory value. So three trajectories are obtained, the predicted one, and an upper and lower bound forming an acceptability interval into which the predicted signal should fit as well as the real monitored signal if the obtained model is of sufficiently high quality. The interval formed in this way has two primary applications: fault detection and model validation.

To understand how the envelope can be obtained, the functioning of the forecasting process in FIR needs to be revisited. The QSE forecasts a value for each of the chosen output variables. At each sampling instant, the mask is shifted one step forward along the class matrix. Classes of the m-inputs are read out from the mask, and a so-called input state vector is constructed. Then the class behaviour matrix is searched for coincidences on that input state vector, and the associated membership and side functions of each record found are compared with the ones of the input state vector. The five nearest neighbours are identified and used to compute the class, membership, and side values of the output. Figure 2-7 gives a clear insight of this process. Each m-output is determined as an averaged value from the m-outputs associated with the five nearest neighbours, where the weights are determined based on the relative relevance (proximity or similarity) of each of the five nearest training data to the testing data record in the m-input space [Cellier, 1991].

The idea behind the envelopes approach [Mirats and Huber, 1999; Escobet *et al.*, 1999] is to compute, for each predicted value, an *interval of acceptability* of the real trajectory value. Up to now, a single (defuzzified) prediction was made at each point in time, which had been computed as an average of the m-output values of the five nearest neighbours in the training data base. Yet, it is perfectly defendable to make predictions in different ways. For example, it may make sense to consider the range of predictions made by the five nearest neighbours as an envelope of acceptable predictions. The closer the five nearest neighbours are to each other, i.e., the smaller the dispersion among them, the more narrow that envelope will be. On the other hand, the larger the dispersion among the five nearest neighbours, the wider the envelope will become. Let α and β be the minimum and maximum predictions made by any of the five neighbours, the envelope will then be defined by the range $[\alpha,\beta]$, a time-varying *interval of forecasting acceptability* associated with the predicted output variable. This information can be exploited for fault monitoring. The SAPS forecasting engine now returns three separate values at each time step: the predicted value of the m-output (a weighted average), the smallest acceptable prediction, α, and the largest acceptable prediction, β.

Notice that the width of the interval $[\alpha,\beta]$ provides information about how good the qualitative model is in terms of the dispersion among the five nearest neighbours (small values indicate that the five neighbours are close to each other, whereas a large interval denotes that the neighbours are sparse and that the prediction may possibly be inaccurate). It also provides an indication about whether the training data set size has been large enough.

Yet there is a second source of inaccuracy to be considered, namely the inaccuracy stemming from the reduced information contained in the selected mask. This inaccuracy can be estimated through the prediction error of a qualitative model. When using a FIR qualitative model to predict a system, a measure of the model error can be obtained comparing the forecast value with the real (fuzzified) value of the concerned variable. To improve the model error, optimal and sub-optimal FIR models can be used at the same time by means of parallel computation [López and Cellier, 1999]. If the mean square error, *mse*, is used, the *forecasting average error* of the FIR qualitative model is obtained. Since FIR operates on MISO systems, for a system with more than one output to be predicted, a model for each of those outputs will have been computed and, hence, a different *mse* value for each of the models will be found. Two approaches can be considered here in order to compute the envelopes for each of the outputs to be predicted. Each different *mse* can be used to calculate the envelopes, or the largest *mse* value can be used to calculate all the envelopes, thereby taking the worst case.

The *mse* value denotes the average inaccuracy of the weighted mean of the five nearest neighbours. To account for this second source of inaccuracy, the interval of forecasting acceptability is widened to the range [$\alpha(1-mse*P/100)$, $\beta(1+mse*P/100)$], where *P* is the predicted value of the output variable and *mse* is the forecasting average error of the qualitative model. This range is called *the interval of variable acceptability*.

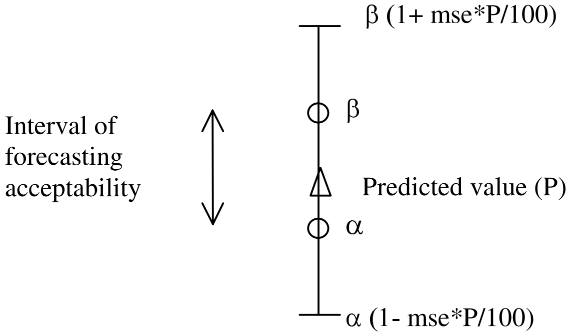


Figure 2-9 Interval of variable acceptability.

Denoting:

$$A = \alpha(1-mse*P/100)$$
$$B = \beta(1+mse*P/100)$$

the interval [A,B] of *variable acceptability* is the one within which the forecast and real values of the output trajectory will be considered to match. Successive values of A and B along the time axis constitute the *variable acceptability envelope* (in short: envelope). Whenever the real value leaves the variable acceptability envelope, an instantaneous error of the concerned output variable can be flagged.

Two experiments have been carried out using this new approach, applied to fault detection on a simulated aircraft model. Results are reported in Appendix II.1.

2.7 Defuzzification Module

The fourth and final module of the SAPS platform performs the reverse operation of the fuzzification module, converting qualitative triples back to real-valued data. The predicted qualitative triples are regenerated into quantitative estimates of the m-output variables. The collection of these quantitative estimates may be used as real predicted trajectories of the simulated variables.

In the fuzzy literature, this process is usually called *defuzzification* while in inductive reasoning terminology the process is called *regeneration*. The side value makes it possible to perform the defuzzification of the qualitative into quantitative values unambiguously and without information loss. The side value is a particular feature of the FIR methodology. Other types of fuzzifiers do not use the side information. Instead they use multiple recodings, associating with each real data point several qualitative pairs consisting of a class value and its associated membership value. Then, multiple class values for the predicted variable are forecast, and defuzzification is computed as an average of the different obtained class values weighted by their associated membership functions values. The approach of the side value used in the FIR methodology offers far better smoothing than other techniques, which makes it possible to get away with a smaller number of classes (three or five classes have been reported to be optimal for most FIR applications), which in turn reduces the model search space.

2.8 Conclusions

The contributions made in the FIR methodology have allowed improving the modelling time and the predictive capability of FIR. The sub-optimal mask computation algorithm is applied to the garbage incinerator example (see Appendices I.3 and II.1). The concept of variable acceptability interval has successfully been applied to improve fault detection following a previous research effort of the same group. This later application is reported in Appendix II.2.

3 Variable selectio n

3.0 Abstract

From now on, the terms static and dynamic model, and static and dynamic relationship will be used. Static is used in the sense that, for a variable trajectory (quantitative) or episode (qualitative), no relations between delayed versions of the variables are considered, i.e., only the actual value of the variable is taken into account. Dynamic means that relations between delayed trajectories or episodes shall be considered in the model. In this chapter, different statistically based techniques are presented that search for static linear relationships among variables, permitting to make a first reduction in the FIR model search space. Dynamic relations will be introduced to the model in Chapter 5. Chapter 3 presents a number of original contributions. Section 3.3 deals with the unreconstructed variance method. In subsection 3.3.2, this methodology is reworked with the purpose of reaching a new goal: a representative set of variables with a lower dimension than the original one is derived. In subsection 3.4.3, the state transition matrix, introduced in the previous chapter and used the entropy reduction index H_r and then the quality of a FIR model, is used to define a measure of similarity (correlation) between variables.

3.1 Introduction

A difficult problem when trying to model the output or outputs of a system from its inputs is knowing which subset of inputs to use in order to make a good prediction of the output or outputs to model. All potential inputs are not always necessary, because some may be redundant whereas others may not provide information that is useful for predicting the behaviour of the output or outputs of the system being studied. The problem becomes worse when dealing with large-scale systems, such as nuclear power plants, airplanes, or water distribution systems, since the list of potential inputs may be formidable. The modeller would like to have at his or her disposal tools with which to choose a representative subset of variables, representative of the plant in the sense that it accounts for all, or almost all, of the information that is contained in the system to be modelled and that is necessary for the task at hand. This is similar to the needs of a plant operator responding to an observed anomaly to know, what screen(s) to look at, or what page(s) of the emergency procedure manual to read.

In the line of the proposed objectives, this and the next two chapters deal with techniques that lead to three kinds of results:

A. To determine a representative subset of variables with a lower dimension than the system itself (an example could be sensitivity analysis).

B. To determine (a) subset(s)/subsystem(s) of variables that (is) are most related with a predetermined variable (such as one of the outputs of the system).

C. To determine (a) subset(s)/subsystem(s) of variables maximally related among themselves.

Yet for didactic reasons, the contents of the aforementioned chapters have been organised in accordance with the statistical approaches used and the conditions under which the results are obtained.

A first approach for reducing the model search space of the FIR qualitative modelling methodology is investigated with the aim of providing forecasts of trajectory behaviour of measured variables for control purposes. In particular, the present chapter deals with the problem of pre-selecting a set of candidate input variables and hence reducing the model search space of FIR. This is important since FIR, as stated in the previous chapter, employs an algorithm of exponential computational complexity in the identification of the best qualitative input/output model, and it therefore takes too much time, even on a big computer, to evaluate all possible qualitative models of a system.

Throughout the chapter, different procedures are applied to perform variable selection. At this point only static linear relations between variables are investigated. From now on, as outlined in the abstract section, the terms static and dynamic model, and static and dynamic relationship will be used. Static is used in the sense that, for a variable trajectory (quantitative) or episode (qualitative), no relations between delayed versions of the variables are considered, i.e., only the variable values at the actual time are taken into account. Dynamic means that relations between delayed trajectories or episodes shall be considered in the model. A static model is one that models the considered output from a subset of inputs and possibly other system outputs at time delay zero, i.e., the same time instant for which the model is trying to predict a new value. A dynamic model takes also into consideration past values of all variables involved in the modelling process including past values of the variable to be predicted[1].

Many approaches for the selection of variables have been presented in the literature using classical and Bayesian statistical techniques as well as other mathematical modelling tools such as neural networks. Principal Components Analysis (PCA) has been studied with this aim in [Jolliffe 1972; Jollife 1973] applied to artificial as well as real data sets. In [Allen, 1971], variable selection is performed using the mean square error of prediction of different possible regression models. In [Allen, 1974], variable selection is presented as a particular case of data augmentation. In [Mansfield *et al.* 1977], a regression model based on principal components is suggested, in which one variable is temporarily eliminated at a time computing the least square error for each of these regression models. The model offering the smallest least square error is then selected, and the corresponding variable is

[1] Strictly speaking, both static and dynamic models make use of 'past data.' If the output to be modelled is, say, $y_1(t)$, to be predicted from variables $x_1(t)$, $x_2(t)$, $x_3(t)$, and $y_2(t)$, it is evident that the observations on which the model is based reflect the dynamic behaviour of the real system, i.e., take their own past into account. The difference between the two types of models is that a static model will generate $y_1(t) = f(x_1(t), x_2(t), x_3(t), y_2(t))$, whereas a dynamic model will also use delayed versions of the variables, e.g., $y_1(t) = f(x_1(t), x_1(t-1), x_1(t-4), x_2(t), x_3(t), x_3(t-2), y_1(t-1), y_2(t-3))$.

permanently discarded. The procedure is repeated until the smallest least square error becomes too big. The admissible procedures to perform variable selection when a regression model is used are analysed and discussed in [Kempthorne, 1984; Kabaila 1997]. [Keller and Bonvin, 1992] present a method to select input and output variables using the principle of internal dominance proposed by Moore [Moore, 1981]. Different regression methods are compared in the task of selecting variables in [Riitta *et al.*, 1994; Lindgren *et al.*, 1995; Hoeting *et al.*, 1996; McShane *et al.*, 1997; Adams and Allen, 1998]. A geometric method for selecting variables in a geometric model based on simple arcs and lines is given in [Widmann and Sheppard, 1995]. [Chipman *et al.* 1997; Hoeting and Ibrahim 1998] used other approaches such as Bayesian and heuristic techniques with the purpose of selecting variables of a system. The canonical correlation analysis is explained and used to select variables in a study of [Al-Kandari and Jolliffe, 1997]. Also, work has been reported in this area using neural networks [Lisboa and Mehri-Dehnavi, 1996; Seixas *et al.*, 1996; Muñoz and Czernichow, 1998].

Different PCA based methods are discussed in Section 3.2 together with a brief introduction of the concepts behind PCA. This description supports the understanding of the method of unreconstructed variance offered in the subsequent section.

The method of the unreconstructed variance for the best reconstruction, described in [Dunia and Qin, 1998a] in a different context, is one of the methods presented for the purpose of selecting, which input variables to use to model a given system output. This technique is discussed in detail in Section 3.3, where a brief review of its mathematical foundations is provided. In Subsection 3.3.2, the operation of the technique is demonstrated by means of the steam generator example offered in Appendix I.2.

Section 3.4 discusses the use of statistical methods based on correlation coefficients. In this section, two correlation indices to select the variables that are most correlated with the output to be modelled are introduced. A simple linear correlation coefficient is explained in Subsection 3.4.1, whereas multiple correlation coefficients are treated in Subsection 3.4.2. In Subsection 3.4.3, a similarity, S_m, index is introduced as a way to measure the amount of common information, i.e., the correlation, between two variables. Although it is based on informational measures that are quite different from the correlation indices presented in the previous subsections, it is included in Section 3.4, because it offers an estimation of the similarity between variables.

The different methodologies presented throughout this chapter are provided together with the results of applying them to examples, in some cases the steam generator process explained in Appendix I.2, in others the garbage incinerator system introduced in Appendix I.3. Later on in Chapter 5, these results are combined with the FIR methodology so as to obtain different static qualitative models of the system. Each one of the methods advocated in this section proposes a set of selected variables from which a static candidate mask can be proposed and a static FIR model can be obtained. Of course, obtaining static qualitative models is not meaningful because none of them will be good enough to model a dynamic system. Yet, there exist good reasons for doing this, as the main objective of this dissertation is to reduce the model search space of the FIR methodology, and, in order to achieve this goal, the research must be done following some methods. In this dissertation, linear static relations between variables are explored first, then non-linear relations are tackled, and finally, time is added, i.e., dynamic relations are being investigated. In each stage, the reduction on the computational cost achieved when computing a FIR model is reported so as to assess that the employed methods lead to good results in front of the proposed objectives.

3.2 PCA based methods

A principal component analysis is concerned with explaining the variance-covariance structure of a set of variables through a few linear combinations of these variables. Its general objectives are data reduction and interpretation.

In order to reproduce the total system variability in a k variables system, k components are required. However, often much of this variability can be accounted for by a small number l ($l < k$) of the principal components. If so, there is almost as much information in the l components as there is in the original k variables. Thus, the original data set, consisting of n measurements on k variables, is reduced to a new data set consisting of n measurements on l principal components. Note that although data reduction is achieved, i.e., the same amount of information can be represented in a lower dimensional space, no variable selection is performed because every principal component is formed as a linear combination of all the variables. This also makes it difficult to physically interpret and intuitively understand the meaning of the principal components, a very important issue when dealing with the kind of systems that this research effort investigates.

As stated before, the principal components are particular linear combinations of the k variables $x_1, x_2, \ldots, x_k$. From a geometric point of view, these linear combinations represent the selection of a new coordinate system obtained by rotating the original coordinate axes formed by the variables $x_1, x_2, \ldots, x_k$. The newly formed axes represent the directions with maximum variability.

The principal components depend only on the covariance matrix Σ. Consider the following linear combinations:

$$lc_1 = \mathbf{a}_1'\mathbf{X} = a_{11}X_1 + a_{12}X_2 + \cdots + a_{1k}X_k \qquad \text{Eq. 3-1}$$

$$lc_2 = \mathbf{a}_2'\mathbf{X} = a_{21}X_1 + a_{22}X_2 + \cdots + a_{2k}X_k \qquad \text{Eq. 3-2}$$

$$\vdots \qquad\qquad\qquad \vdots$$

$$lc_k = \mathbf{a}_k'\mathbf{X} = a_{k1}X_1 + a_{k2}X_2 + \cdots + a_{kk}X_k \qquad \text{Eq. 3-3}$$

Hence:

$$Var(lc_i) = \mathbf{a}_i'\Sigma\mathbf{a_i} \qquad \text{Eq. 3-4}$$

The principal components are those uncorrelated linear combinations $lc_1, lc_2, \ldots, lc_k$, whose variances are as large as possible in Eq. 3-4. The covariance must be estimated from the available data. For variables x_i, x_j; $i, j = 1...k$, the sample covariance is computed as the average product of the deviations from their respective means:

$$Cov(x_i, x_j) = s_{ij} = \frac{1}{n}\sum_{m=1}^{n}(x_{mi} - \bar{x}_i)(x_{mj} - \bar{x}_j) \qquad \text{Eq. 3-5}$$

The first principal component is the linear combination with maximum variance, i.e., it maximises $Var(lc_i) = \mathbf{a}_i^{'}\Sigma\mathbf{a_i}$. The constraint $\mathbf{a}_1^{'}\mathbf{a}_1 = 1$ is imposed in order to eliminate the indeterminacy encountered when $\mathbf{a}_1$ is multiplied by any constant.

The second principal component is the linear combination $\mathbf{a}_2^{'}\mathbf{X}$ that maximises $Var(\mathbf{a}_2^{'}\mathbf{X})$ subject to $\mathbf{a}_2^{'}\mathbf{a}_2 = 1$ and $Cov(\mathbf{a}_i^{'}\mathbf{X}, \mathbf{a}_2^{'}\mathbf{X}) = 0$. These formulas are used to obtain all principal components so that in the i^{th} step, the i^{th} principal component is the linear combination $\mathbf{a}_i^{'}\mathbf{X}$ that maximises $Var(\mathbf{a}_i^{'}\mathbf{X})$ subject to $\mathbf{a}_i^{'}\mathbf{a}_i = 1$ and $Cov(\mathbf{a}_i^{'}\mathbf{X}, \mathbf{a}_p^{'}\mathbf{X}) = 0$ for $l < i$.

If the variables of the system are standardised to zero mean and unit standard variance using

$$Z_i = \frac{(X_i - \mu_i)}{\sqrt{\sigma_{ii}}} \qquad \text{Eq. 3-6}$$

where μ_i and σ_{ii} are the mean and the standard deviation of the i^{th} variable, the principal components may also be obtained from the sample correlation matrix of the data. The standardisation of the variables is not required mathematically but is almost required from a practical point of view. Consider for example a system from which variables with very different nature, such as temperature, flow, and pressure have been measured. Their scales may be orders of magnitude different, so their variance will not account for the same in the analysis. To correct this problem and allow all the variables to be weighted equally in the covariance analysis, the variables are standardised. In this case, the principal components can be obtained by means of a Singular Value Decomposition (SVD) as the eigenvectors of the correlation matrix of **X**. In general, the principal components obtained from the correlation matrix are different than those obtained from the covariance matrix. A full description of how the principal component analysis is performed can be found in [Jackson, 1991] or [Johnson and Wichern, 1992].

A difficult parameter to establish when performing a principal component analysis is the number of principal components to keep. There exist several methods advocated in the literature such as the unreconstructed variance method [Dunia, 1997], reviewed for other purposes in the next section, or the cross-validation method [Wold, 1978; Osten, 1988]. A comparative study between different methods for this purpose is given in [Valle *et al.*, 1999].

A wide range of publications has been written about the use and theory of the principal component analysis. For instance, in [Jeffers, 1967], two examples of application, on the physical properties of pitprops and the variation of alate adelges, are presented. A good survey of the principal components method can be found in [Glen *et al.*, 1989a], while in [Glen *et al.*, 1989b], a software package to perform this kind of analysis is explained. In [D'Ambra and Lauro, 1992], a PCA analysis of the dependent variable images on the subspace generated by the explicative data set is performed; the method is called constrained principal component analysis (CPCA). Another view of the CPCA method under the name of redundancy analysis is offered in [Israels, 1992]. Although the principal component analysis was not designed to perform variable selection, it can be used for this purpose. Some PCA based variable selection methods are given in [Jollife 1972; Jollife 1973]. In [Krzanowski, 1987], the principal components method is used to select an appropriate subset of variables that conveys the main features of the whole data set. In [Tanaka and Yuichi, 1997], the principal components are computed on a subset of variables but in a way that represents all the system variables.

In the remainder of this section, two PCA based variable selection methods are applied to the example of the steam generator presented in the Appendix I.2. Later on, in chapter 5, the obtained results will be compared with other variable selection techniques by means of the quality loss in the computed FIR models and the computation reduction achieved with each used variable selection method. The two methods presented are based on [Beale *et al.*, 1967; Jollife 1972].

If the classification given in the introduction section of this chapter is followed, both of the methods, as will be seen, lie in option A. The first method eliminates the variables with highest coefficients of the least important principal components. The second retains the variables with highest coefficients of the most important principal components. Both methods try to eliminate those variables that, theoretically, primarily account for noise in the system, and in this sense, the two methods obtain a representative subset of variables of the given system. The potential drawbacks of both of these methods is that they are based entirely on linear analysis, and therefore, they may identify non-linear relations among variables as system noise, thereby suggesting a non-optimal subset of variables to represent the underlying system.

The first method will be referred to as the *PCA_A* method. In this method, l variables, $l < k$, are to be retained, whereas the remaining k-l variables of the system are to be rejected. The value of the parameter l may be chosen in various ways; for example, it could be equal to the number of eigenvalues of the correlation matrix greater than some predefined value $\lambda_{0,}$ or it could be equal to the number of principal components necessary to account for more than some given proportion, α_0, of the total variation.

A principal component analysis is performed on all of the original k variables, and the eigenvalues of the data correlation matrix are inspected. Then, one variable is associated with each one of the last k-l components (the eigenvectors themselves) and rejected. The way for doing so is, beginning with the principal component corresponding to the smallest eigenvalue, to discard the variable with the largest coefficient in this component that has not been eliminated before. The algorithm proceeds in this way with all the chosen principal components. According to the references, 0.7 is a good value for λ_0.

Two versions of this method have been implemented. The first of them, named *PCA_A1*, uses the factors of the principal components as they are, i.e., preserving their sign. When applying the method to the steam generator system (see Appendix I.2 for a complete description of this example), it is found that variables 3 and 8 must be retained, whereas variables 4, 1, 7, 2, 5, and 6 (in this order) are to be discarded.

The second version, referred to as *PCA_A2*, makes use of the absolute value of the coefficients. When applying this method to the boiler data, it is found that variables 2 and 3 must be retained, whereas variables 4, 1, 7, 6, 5, 8 (in this order) must be discarded.

Both versions of the method find variable 3 to be important for the analysis of the system. The *PCA_A1* method finds also variable 8; the *PCA_A2* method uses also variable 2, whilst variable 8 is the last variable to be discarded in this second version of the method. For this system, the best subset of variables to model the output, see the Appendix I.2, has been found to be the subset formed by variables 2, 3, and 8. Notice that the *PCA_A2* method would have found the optimal set of variables to be retained if only a slightly smaller value of λ_0 would have been chosen.

The second method based on principal component analysis is the one named *PCA_B*. The basic idea is the same as in the previous case, but operates in backward mode. Again, a principal component analysis is performed on all the variables. This time, those principal components with eigenvalues larger than λ_0 are taken into account. Then, beginning with the vector corresponding to the largest eigenvalue, the variable with the largest coefficient in this principal component that has not been selected already before is preserved. The method proceeds with all the principal components chosen in this way. As stated in the references, 0.7 is also a good value for λ_0 in this case.

Once more, the results of two versions of this algorithm are presented using the steam generator system. The first of them, named *PCA_B1*, uses the coefficients of the principal components as they are, preserving their sign. When applying the method to the boiler data, it is found that variables 1 and 2 (in order of acceptance) are to be retained, while the remaining variables are to be discarded.

The second version of this method, named *PCA_B2*, uses the absolute value of the coefficients. When applying this method to the boiler data, it is found that variables 1 and 3 must be retained with all other variables being discarded.

In this case, the obtained subsets of variables are different from the subset that has been found to be the best for this system (see Appendix I.2), consisting of variables 2, 3, and 8.

3.3 Unreconstructed variance methodology

The method described in this section was developed at the University of Texas in Austin [Dunia and Qin, 1998b; Dunia, 1997] where the author was kindly invited for a three months period during the fall of 1998. The underlying methodology was previously used to select the number of principal components to keep in a PCA model, based on the best reconstruction of the variables. The purpose of the PCA model was to identify faulty sensors in a system, and to reconstruct sensor data values from measurements of sensors attached to other signals, exploiting the redundancy inherent in multiple sensor data streams. The methodology had not been designed as a tool for finding an input/output model of a system, though the two tasks are evidently related to each other.

3.3.1 Original method

The original method has a twofold objective: to determine the number of principal components to keep in a PCA analysis [Qin and Dunia, 1998], and to reconstruct faulty sensors measurements as well as identify the system fault from correct data of the system [Dunia *et al.*, 1996]. If variable selection is intended with this methodology, two subsets of variables are obtained. The first one corresponds to a subset of variables with a high linear correlation among them. The second subset is formed by variables with low linear correlation with the first set. Following the classification advocated in the introduction of the chapter, a subset of variables maximally related among each other is obtained, thus the method belongs to the C option.

When a PCA model is used to reconstruct missing or faulty values, the reconstruction error is a function of the number of intervening principal components. In order to determine the number of principal components to be used, the methodology proposes making use of the variance that the model cannot reconstruct; that is, it uses the variance of the reconstruction error.

Let us show some of the mathematical foundations underpinning the methodology for fault identification by variance reconstruction that in turn lead to a method for determining the number of principal components to keep in a PCA analysis. A full description of the method can be found in [Dunia, 1997]. The approach discussed in those papers makes use of a normal process model to decompose the sample vector into two parts:

$$x = \hat{x} + \tilde{x} \qquad \text{Eq. 3-7}$$

where $x \in \Re^{m}$ represents a normalised, zero mean and unit variance, sample vector of m sensors. The vectors $\hat{x}$ and $\tilde{x}$ are the modelled and residual portions of x, respectively. Consider there are n samples for each sensor and that a data matrix $\boldsymbol{X} \in \Re^{n \times m}$ is formed. By either the NIPALS algorithm [Wold *et al.*, 1987], or a singular value decomposition algorithm, the data matrix can be decomposed into score and loading matrices, $\boldsymbol{T}$ and $\boldsymbol{P}$, whose columns are the right singular vectors of $\boldsymbol{X}$. Hence, principal component analysis is used to calculate the projection of a particular sample vector onto the principal component subspace,

$$\hat{x} = \boldsymbol{P}\, t \;=\; \boldsymbol{P}\,\boldsymbol{P}^T x \;=\; \boldsymbol{C}\, x \qquad \text{Eq. 3-8}$$

where $\boldsymbol{P} \in \Re^{m\times\ l}$ is the loading matrix, and $t \in \Re^l$ is the score vector. The number of retained principal components is $l \geq 1$. The matrix $\boldsymbol{C} = \boldsymbol{PP}^T$ represents the projection on the l-dimensional principal component subspace, so $\hat{x}$ represents an l-dimensional vector. The residual $\tilde{x}$ lies in the residual subspace of m-l dimensions

$$\tilde{x} = (\boldsymbol{I}^{(m)} - \boldsymbol{C})\, x \qquad \text{Eq. 3-9}$$

The matrices **C** and **I**-**C** are idem-potent. The PCA model partitions the measurement space ($\Re^m$) into two orthogonal subspaces: the principal component subspace, which includes data variations in agreement with the measurement correlations, and the residual subspace, in which variations due to model errors and noise in the data, i.e., variation not explained by the model, are included. Hence the vector x is orthogonally decomposed into vectors $\hat{x}$ and $\tilde{x}$ by projecting onto the previously named subspaces.

This decomposition can be used to detect faults in a sensor and then reconstruct the faulty sensor data by means of data from other sensors in the system. The PCA analysis captures the measurement correlations of historical data when the process is in a normal operation. The violation of the variable correlations indicates an unusual situation. In a normal operation state, the vector x has a certain projection onto the principal component subspace, $\hat{x}$, as well as onto the residual subspace, $\tilde{x}$. When a correlation breakdown in the current sample x occurs, its projection onto the residual subspace increases, so $\tilde{x}$ reaches unusually large values compared to those obtained during normal conditions.

The sample vector for normal operating conditions is denoted by x^* (unknown when a fault has occurred). In the presence of a process fault $\Im_i$, the sample vector can be represented as:

$$x = x^* + f\,\xi_i \qquad \text{Eq. 3-10}$$

where ξ_i is a normalised fault direction vector, and the scalar f represents the magnitude of the fault. Here, only faults described by a single direction are considered. For example ξ_i^T $= [0\ 1\ 0\ ...\ 0]$ represents a failure of the second sensor in the sample vector x. Research has been done to extend the presented methodology to a more general approach based on multidimensional fault descriptions [Dunia and Qin, 1997a,b; Dunia and Qin 1998c].

The fault direction vector can be projected onto the two subspaces:

$$\xi_i = \hat{\xi}_i + \tilde{\xi}_i \qquad \text{Eq. 3-11}$$

where fault $\Im_j$ has been assumed to occur. The normal vector operation x^*, can be reconstructed from x along all possible fault directions. For each assumed fault $\Im_j$, the sample x is moved back in the direction ξ_i, such that its distance to the principal component subspace is minimised. Hence for each assumed j-fault, a reconstructed vector x_j is obtained moving x in the ξ_j direction,

$$x_j = x - f_j \xi_j \qquad \text{Eq. 3-12}$$

where f_j is an estimate for f. When the assumed fault coincides with the actual fault (i.e., $\Im_j=\Im_i$) the reconstructed vector is expected to be close to x^*. The distance between x_j and the principal component subspace is given by the magnitude of the square prediction error (SPE) for the reconstructed vector. For the true fault, the greatest reduction in the SPE is expected. The fault magnitude f_j is obtained by minimising SPE_j along the direction ξ_j.

$$SPEj \equiv // \tilde{x} //^2 = // \tilde{x} - \tilde{f}_j \tilde{\xi}_j^0 //^2 \qquad \text{Eq. 3-13}$$

$$\frac{dSPE_j}{d\tilde{f}_j} = 0 \quad \text{leading to } \tilde{f}_j = \tilde{\xi}_j^{0^T} \tilde{x} \qquad \text{Eq. 3-14}$$

where $\tilde{f}_j = f_j \left\| \tilde{\xi}_j \right\|$.

Now, the notion of unreconstructed variance [Dunia et al., 1996] can be presented. The notion of unreconstructed variance is based on the estimation of sensor i from all other sensor data. Of course, when computing this reconstructed value, there is always a portion of the faulty sensor information that cannot be reconstructed from the other sensor data. This portion is the unreconstructed variance of the sensor. If the assumed fault is the actual fault in Equation 3.1-8, that is, $j = i$, the following expression is obtained:

$$\tilde{f}_i = \tilde{\xi}_i^{0^T} (\tilde{x}^* + \tilde{f}\tilde{\xi}_i^0) \qquad \text{Eq. 3-15}$$

The addition of equations 3.3.1-4 and 3.3.1-6 illustrates the effect of f_i-f when comparing x_i with x^*

$$\left\| x^* - x_i \right\|^2 = (f - f_i)^2 = \left(\frac{\tilde{\xi}_i^T \tilde{x}^*}{\tilde{\xi}_i^T \tilde{\xi}_i} \right)^2 \qquad \text{Eq. 3-16}$$

The unreconstructed variance, u_i, in the direction ξ_i represents the variance of the projection x^*-x_i on the fault direction ξ_i

$$u_i \equiv \operatorname{var}\left\{ \xi_i^T (x^* - x_i) \right\} = \varepsilon\left\{ \| x^* - x_i \|^2 \right\} = \frac{\tilde{\xi}_i^T \varepsilon\left\{ x^* x^{*T} \right\} \tilde{\xi}_i}{\left(\tilde{\xi}_i^T \tilde{\xi}_i \right)^2} = \left(\frac{\tilde{\xi}_i^T \tilde{R} \tilde{\xi}_i}{\left(\tilde{\xi}_i^T \tilde{\xi}_i \right)^2} \right) \qquad \text{Eq. 3-17}$$

where $\tilde{R}$ denotes the covariance matrix of the normal residual. The condition of minimising u_i with respect to l

$$\min_{l} u_i \qquad \text{Eq. 3-18}$$

can be used to determine the number of principal components and the set of sensors to keep for process monitoring. The unreconstructed variance can be projected onto the two subspaces

$$u_i = \hat{u}_i + \tilde{u}_i \qquad \text{Eq. 3-19}$$

An illustration of how to minimise u_i with respect to l and m is given in [Dunia and Qin, 1998]. The authors have shown that $\tilde{u}_i$ is monotonically decreasing with respect to l, and $\hat{u}_i$ tends to infinity as l tends to m. Figure 3-1 illustrates this effect.

Equation 3.1-12 only provides the optimal l for $\Im_i$. If the set of all possible faults, $\{\Im_j\}$, is considered, the objective function is

$$\min_{l} q^T u = \min_{l}(q^T \tilde{u} + q^T \hat{u}) \qquad \text{Eq. 3-20}$$

where u represents the vector of unreconstructed variances for all $\Im_i \in \{\Im_j\}$, and q is a weighting vector with positive entries.

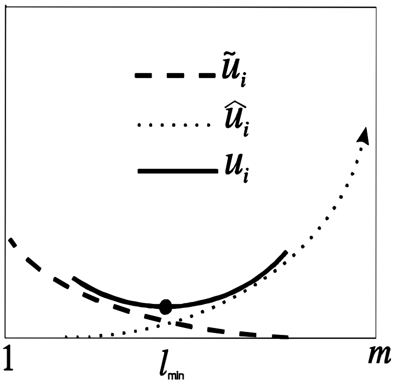


Figure 3-1 Unreconstructed variance as the summation of $\hat{u}_i$ and $\tilde{u}_i$.

Summarising, the methodology of unreconstructed variance has been developed to accomplish two main actions:

- Determine the number of principal components to keep when performing a principal component analysis.

- Reconstruct faulty sensors (or missing data) by using data from other sensors, whose signals are linearly correlated with the missing signal.

3.3.2 Reducing the FIR mask search space using the unreconstructed variance methodology

The unreconstructed variance methodology has been reworked with the aim of simplifying the FIR model search space. As mentioned in the previous section, if variable selection is intended with this methodology, two subsets of variables are obtained. The first one corresponds to a subset of variables with a high linear correlation among them. The second subset is formed by variables that exhibit low linear correlation with the first set. Following the classification advocated in the introduction of the chapter, a subset of variables maximally related among each other is obtained, thus placing the method in the C option.

In this section, the functioning of the unreconstructed variance methodology is demonstrated by means of an industrial steam generator with the aim of selecting a subset of variables. A full description of the steam generator process as well as the variables that FIR selects[2] to qualitatively model the NOx gas emission using a static model are provided in Appendix I.2.

The first step of the methodology requires the analysis of the available measurement data so as to discover what variables are well reconstructed from which others. For the PCA analysis to be applicable, the data must first be normalised to zero mean and unit variance. Given a system with *k* variables, every variable is reconstructed from the other *k-1* variables, and its unreconstructed variance is computed as a function of the number of retained principal components. The results for the case of the boiler data are tabulated in Table 3-I, where, in order to later obtain a fair comparison between different variable selection techniques, only the training data (85% of the total available data) were used in the analysis.

Var \ # PC	1 PC	2 PCs	3 PCs	4 PCs	5 PCs	6 PCs	7 PCs	8 PCs
x_1	0.0128	0.0128	0.0047	0.0036	0.0033	0.0025	0.0024	0.0062
x_2	0.0190	0.0118	0.0078	0.0074	0.0063	0.0047	0.0036	0.0107
x_3	0.9957	18.2113	1.4427	1.0621	1.0522	1.0407	0.8821	6.8374
x_4	0.0109	0.0079	0.0050	0.0038	0.0036	0.0032	0.0019	0.0019
x_5	0.0305	0.0304	0.0286	0.0283	0.0551	0.0538	0.0357	0.0508
x_6	0.0338	0.0335	0.0229	0.0189	0.0190	0.2857	0.2253	0.3810
x_7	0.0307	0.0306	0.0121	0.0120	0.0105	0.0107	0.0266	0.0278
x_8	0.0688	0.0676	0.0630	0.2825	0.2822	1.7516	4.7219	14.4137
Y	0.5763	0.5768	2.2764	2.2718	2.2282	2.2844	12.3508	147.682

Table 3-I Unreconstructed variance table for the boiler example.

The method then proceeds by summing up the unreconstructed variances of each column. The number of principal components for which the sum of the unreconstructed variances is a minimum is determined. For the boiler data, the obtained results are shown in Table 3-II.

[2] In this case, as described in the Appendix I.2, FIR works with the full set of the system variables and generates qualitative models of depth 1.

1.7786	18.9827	3.8633	3.6903	3.6603	5.4373	18.2503	169.412

Table 3-II Total unreconstructed variance for each number of principal components.

It turns out that, in this system, only one principal component needs to be retained. The method then throws out those variables that exhibit, for the optimal number of retained principal components, unreconstructed variances that are bigger than the value that would be obtained if the measurement data for those variables were replaced by their mean values. Due to normalisation, the theoretical threshold value is 1. In practice, it may be better to throw out variables with an unreconstructed variance above 0.8 or 0.9.

For the boiler example and looking at the *1 PC* column of Table 3-I, it can be seen that variable 3 needs to be thrown out. This means that any variable of the system, except for variable 3, can be reconstructed using a PCA model with a single PC that is made up from the information available through all variables except for variable 3, which should be ignored.

3.3.2.1 Modelling NOx output using the unreconstructed variance method

It can be seen from Table 3-I that reconstructing the *NOx* level (listed as variable 9 in the system description given in the Appendix I.2, and referenced as *Y* in the previous section) from the other variables is considerably more difficult than reconstructing any of the other signals with the exception of variable 3. Yet, the approach suggests that it is meaningful to construct a PCA model for variable 9, using a single principal component made up from variables 1, 2, 4, 5, 6, 7, and 8.

The results of the analysis are shown in Figure 3-2. The continuous line shows the measurement data of the *NOx* variable during the validation period. The dashed line shows the predictions made by the PCA model made up from the training data using a single principal component composed from the variables 1, 2, 4, 5, 6, 7, and 8. The Medium Square Error (MSE) value obtained from this model for the training data period is 0.7073, and for the validation period, it is 0.9979. The reader may notice that the prediction is indeed better for the training data than for the validation data.

The MSE value is considerably higher than in the case of the FIR model (presented in Appendix I.2). The reason is that the PCA model makes no attempt at replicating the high frequency oscillations exhibited by the real data. The model has clearly low-pass characteristics. Hence, even if the prediction looks good to the naked eye, the distances between the prediction and the real data are large most of the time, and consequently, the MSE value cannot be made small.

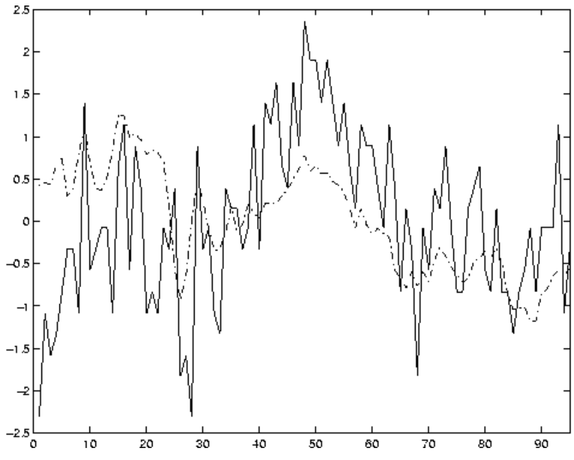


Figure 3-2 Prediction given by a PCA model using input variables 1, 2, 4, 5, 6, 7, and 8.

3.3.2.2 Modelling NOx output using an alternative set of variables

Did the reconstructed variance analysis do a good job at deciding which variables to retain in the PCA analysis? To answer this question, a second PCA model was made, also consisting of a single principal component, but made using the variables 2, 3, and 8, as proposed by FIR (see Appendix I.2). The results of this prediction are shown in Figure 3-3. This time, the MSE values are 0.5703 for the training data set, and 0.8306 for the validation data set.

Comparing Figure 3-2 and Figure 3-3 by naked eye, the predictions look kind of similar. Yet, the MSE values of the prediction of Figure 3-3 are considerably lower, i.e., FIR did a better job than the reconstructed variance analysis at deciding which variables need to be retained in order to obtain a decent prediction, even if the prediction is to be made by a PCA model.

In this second experiment the variables taken into account are the one discarded by the unreconstructed variance method, i.e., variable 3, and two of the variables this method retains, variables 2 and 8. It makes some sense that the results obtained in this second experiment were better than in the first one, where only the variables that the unreconstructed variance method selected were used. The group selected by the uncorrelated variance method consists of a set of highly linearly correlated variables, so containing similar, and therefore redundant, information about the studied system. On the other hand, variables discarded by the method at hand are not strongly correlated with the previous set of variables, thus containing different and possibly still useful information about the system. It seems plausible that an acceptable group of variables to model a system might consist of the discarded variables of the unreconstructed variance method and a subset of the variables that are kept to construct the PCA model. Unfortunately, the method does not provide information as to which of these variables ought to be retained.

A meaningful selection can be achieved, for instance, by means of FIR, as done in the presented example. A method that selects this kind of subset of variables would lie, following the classification advocated in the introduction of this chapter, in the A option, i.e., determining a representative subset of variables.

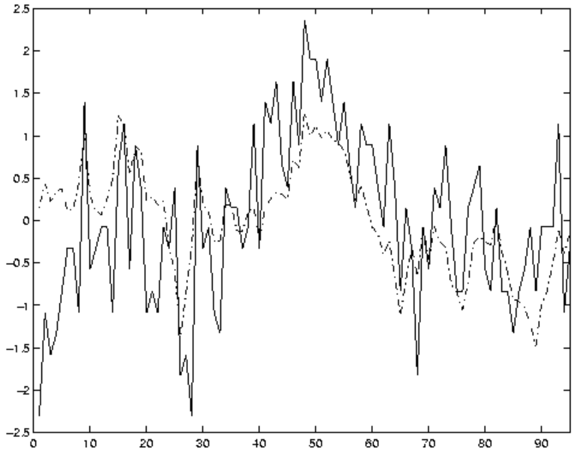


Figure 3-3 NOx output predicted using a PCA model built from variables 2, 3, and 8.

3.4 Correlation based methods

An obvious way to perform variable selection is to use the index of the Pearson correlation, i.e., to look for the strength of the relation between two or more variables. Correlation analysis is frequently being applied in a number of different fields such as biology [Kim *et al.*, 1998], astronomy [Aoki and Yoshida, 1999], economy [Tanaka *et al.*, 1999] and control [Gloss, 1998]. This list is meant to be representative rather than inclusive, as the technique is so widely used that an enumeration of all application areas would be quite meaningless.

This section is structured as follows: first some of the mathematical underpinnings of correlation analysis are stated; subsequently, variable selection using correlation analysis is demonstrated by means of the garbage incinerator example (Appendix I.3). This is explained in subsection 3.4.1. Then, in Subsection 3.4.2, a variable selection for the boiler system (Appendix I.2) is accomplished using multiple correlation coefficients. Finally in Subsection 3.4.3, a similarity index is introduced as a way to measure the amount of common information, i.e., the true (non-linear) correlation between two variables. Although it is based on informational measures far different from the basis of the correlation indices presented in previous subsections, it is included in this section because it gives an estimation of the correlation between variables.

3.4.1 Simple correlation matrix

The sample correlation coefficient, or Pearson's product-moment correlation coefficient [Bhattacharyya and Johnson, 1977], offers a measure of the linear association between two variables. This measure does not depend on the units of measurement. It can be interpreted as a standardised version of the sample covariance (Eq. 3-5), where the product of the square roots of the sample variances provides the standardisation. The sample correlation coefficient for the pair <x_i, x_j> is computed as,

$$r_{ij} = \frac{s_{ij}}{\sqrt{s_{ii}}\sqrt{s_{jj}}} = \frac{\sum_{m=1}^{n}(x_{mi}-\bar{x}_i)(x_{mj}-\bar{x}_j)}{\sqrt{\sum_{m=1}^{n}(x_{mi}-\bar{x}_i)^2}\sqrt{\sum_{m=1}^{n}(x_{mj}-\bar{x}_j)^2}} \qquad \text{Eq. 3-21}$$

The signs of the sample correlation and covariance are the same, but the first one is normally easier to understand because its magnitude is bounded. The most important properties of the correlation coefficient are:

- The value of the correlation coefficient, r_{ij}, is bounded between –1 and 1.

- It measures the strength of the linear relation between the two involved variables. When $r_{ij} \cong 0$, the two variables are not strongly linearly related. Otherwise, the sign of r_{ij} indicates the direction of the association; $r_{ij} > 0$ implies a tendency for one value of the pair to increase when the other value increases; and $r_{ij} < 0$ implies a tendency for one value of the pair to decrease when the other value decreases.

- The value r_{ij} remains unchanged if the measurements of the i^{th} and j^{th} variables are changed to $z_{im} = a \cdot x_{im} + b$, and $z_{jm} = c \cdot x_{jm} + d$, $m = 1 \ldots n$, respectively, provided that the multipliers a and c have the same sign.

Both correlation and covariance provide measures of linear association, or, in other words, association along a line. When using this kind of indices to study the possible relation between variables, the researcher should remember that non-linear relations may exist between variables that are not revealed by these descriptive statistics. Also, these statistics are very sensitive to outlier observations that may indicate a strong relation when in fact no or only little relation exists. In spite of these shortcomings, the correlation index is frequently used, as it is very easy to compute and may offer a first insight about which variables are related to what others when the structure of a system is being investigated.

Although no single variable contains complete information about the system under study, the variables describing a complex system are often correlated to each other. Here, a first variable selection stage of a large-scale system is presented by means of a simple correlation analysis. In this first approach, time it is not considered, i.e., only instantaneous values of variables are taken into account. The correlation analysis presented in this subsection forms part of a complete methodology of decomposing a complex systems into subsystems that will be fully described in due course, but that can be used in a stand-alone mode for eliminating the redundant variables of a system.

The proposed method is based on the simple idea to identify those variables that contain the same information about the system and drop them out of the subsequent analysis so simplifying a posterior model computation. A correlation analysis of the input data is performed, obtaining the data correlation matrix, so variables with a strong linear relationship can be detected. Afterwards, those variables with an absolute linear correlation coefficient equal or higher than an absolute upper limit r_0 are identified. The variables in these groups contain almost the same information about the studied process and hence, any one of them may be chosen to represent such information. In order to select this variable, different approaches can be used: choose it randomly, take the variable with the highest correlation with the output to be modelled, or select the one that maximises the measure *T* of energy transference [Conant, 1981]. Here, the criterion of choosing those variables maximising the variation coefficient has been adopted because it is easy to compute from statistics already computed for the correlation analysis[3]. This index is a normalisation of each variance variable using its mean as normalisation factor. For variable x_i this index can be computed as:

$$vc_i = \frac{std_i}{|\bar{x}_i|} = \frac{\sqrt{\mathrm{var}_i}}{|\bar{x}_i|} \qquad \text{Eq. 3-22}$$

The analysis has been applied to the data of the garbage incinerator system presented in Appendix I.2. The sample correlation matrix for the 19 input variables of the garbage incinerator system is given in Table 3-IV. For this example the value chosen for r_0 has been $r_0 = 0.75$ which has been considered high enough to denote whether a linear correlation exists or not. In the table, values equal or higher than r_0 are marked in bold.

When looking for variables with correlation equal or higher than r_0, the following sets are found:

X_1	X_9		
X_3	X_{14}		
X_8	X_{15}	X_{16}	X_{17}

Table 3-III. Groups of linearly correlated variables

[3] In order to select a unique variable from each cluster, alternative algorithms could be used, such as choosing the variable that exhibits the strongest correlation with the output. However the selection algorithm is not very critical and any reasonable algorithm will usually lead to similarly good FIR predictions.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}
X_1	1,00																		
X_2	0,12	1,00																	
X_3	0,01	0,18	1,00																
X_4	0,05	0,25	0,05	1,00															
X_5	0,01	0,02	0,06	0,01	1,00														
X_6	0,41	0,51	0,21	0,39	0,13	1,00													
X_7	0,08	0,55	0,31	0,34	0,22	0,71	1,00												
X_8	0,08	0,07	0,17	0,26	0,28	0,46	0,59	1,00											
X_9	**0,88**	0,27	0,08	0,06	0,00	0,18	0,29	0,07	1,00										
X_{10}	0,15	0,16	0,48	0,09	0,08	0,16	0,32	0,26	0,20	1,00									
X_{11}	0,01	0,54	0,02	0,31	0,07	0,46	0,54	0,31	0,17	0,49	1,00								
X_{12}	0,38	0,44	0,12	0,53	0,02	0,40	0,66	0,51	0,53	0,28	0,51	1,00							
X_{13}	0,13	0,62	0,26	0,37	0,14	0,56	0,72	0,53	0,32	0,37	0,70	0,69	1,00						
X_{14}	0,00	0,27	**0,83**	0,31	0,19	0,47	0,60	0,59	0,11	0,52	0,27	0,43	0,53	1,00					
X_{15}	0,04	0,17	0,29	0,43	0,22	0,46	0,55	**0,77**	0,09	0,02	0,19	0,53	0,54	0,66	1,00				
X_{16}	0,09	0,13	0,09	0,53	0,25	0,43	0,57	**0,78**	0,10	0,14	0,37	0,60	0,51	0,44	**0,75**	1,00			
X_{17}	0,08	0,16	0,08	0,54	0,25	0,46	0,60	**0,77**	0,11	0,14	0,38	0,62	0,51	0,44	**0,75**	**0,99**	1,00		
X_{18}	0,13	0,00	0,01	0,10	0,07	0,06	0,03	0,04	0,09	0,02	0,01	0,02	0,02	0,03	0,02	0,03	0,03	1,00	
X_{19}	0,04	0,00	0,03	0,03	0,25	0,17	0,20	0,35	0,02	0,09	0,11	0,06	0,23	0,20	0,28	0,27	0,24	0,04	1,00

Table 3-IV Input data correlation matrix for the garbage incinerator system.

Now, one variable can be chosen from each group to represent the system information each of the groups has about the system. As explained before, the variable with the highest variation coefficient is selected. Table 3-V lists the mentioned coefficients for each of the input variables.

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}
0.068	0.1097	0.0747	0.0026	0.1974	0.1058	0.0149	0.1417	0.1663	0.1790	0.1032	0.0885	0.0703	0.1501	0.1122	0.0831	0.0894	0.1475	0.0093

Table 3-V. Variation coefficient for the garbage incinerator input variables.

So the selected variables are X_9, X_{14} and X_8 respectively, and variables X_1, X_3, X_{15}, X_{16}, X_{17}, are discarded in this first step of the analysis. Notice that with a very simple, fast computation method, 5 variables have been already discarded because they have been found to be redundant in order to model this system. In the garbage incinerator system, 5 variables out of 19 represent 26% of the total number of variables. Therefore, whichever modelling methodology is used in a posterior stage, a substantial saving of computation time is achieved by means of using a cheap straightforward variable selection technique as the one advocated in this section. Obviously, some limitations exist with this method. The two main shortcomings of it are that time, i.e., dynamic relations, and non-linear relations have not been considered. Yet, as this variable selection forms part of a complete modelling methodology to be explained later on, those problems can be later tackled and fixed. For now, only simple variable selection while searching for linear static relations was intended.

3.4.2 Multiple correlation coefficients

In order to provide a descriptive measure of the global model fit in a multiple regression analysis, the quotient between the explained variability by the regression and the total variability is used. This measure is called the coefficient of determination or coefficient of multiple correlation, *R*. The *R* coefficient is bounded, $|R| < 1$, and $R = 1$ indicates that there exists an exact functional relation between the response variable and the explanatory variables. It represents the simple correlation coefficient between the original response variable and the modelled or estimated one.

Given a system with *k* input variables, there is an obvious way of choosing a subset *p*, $p<k$, of independent[4] variables when a multiple regression is performed. This is to search for those *p* variables that maximise the multiple correlation of the dependent variable with the *p* independent variables, i.e., that minimise the residual mean square. To obtain the optimal subset of variables with this method, all the possible regressions with all the possible subsets of input variables must be computed. This is an arduous task, unmanageable when dealing with a complex system. A method for computing all the possible regressions in an efficient manner is presented in [Schatzoff, 1968]. In [Allen, 1971], this kind of regression based variable selection is performed. However, this is not the index used in this subsection to perform variable selection, because, at this point, no multiple regression has been performed yet.

An extension of the mentioned regression analysis is to retain the subset of variables that maximises the minimum multiple correlation between the *p* selected variables and any of the *k-p* variables. Here, another meaning of the term multiple correlation is used also advocated in the literature, which is to consider simultaneously the simple correlation coefficient that a variable has with more than one variable at a time. This method was suggested by [Beale *et al.*, 1967], and it is capable to find the best possible subset of *p* variables, with the limitation that only linear relations are being searched, but it is very slow.

In this same line, other method that uses multiple correlation, reported in [Jollife, 1972], is a stepwise method that is faster than the one used in [Schatzoff, 1968] but that does not necessarily find the best subset of variables. This method is labelled *MC* for posterior reference in this thesis. The method works as follows: Suppose there are *k* variables. The first discarded variable is the one that exhibits the maximum multiple correlation with the remaining *k*-1 variables. The multiple correlation of a variable is calculated, in this case, as the sum of the individual sample correlation between that variable and any other variable. The algorithm is repeated with the remaining variables. At each stage with *q* variables remaining, the variable having the largest multiple correlation with the other *q*-1 variables is the next to be discarded. The algorithm terminates when the maximum multiple correlation between variables has decreased to a value below R_0.

[4] In the statistics context, the terms independent, explanatory or regressor variable are used for the input variables, i.e., those variables from which the output can be explained or modelled. The terms explained, response, regressed or dependent variable are used for the variable to be modelled, that is, the one considered as the system output.

This algorithm was applied to the steam generator system explained in Appendix I.2. Results are summarised in Table 3-VI. Variables 2 and 3 were retained by the method, while variables 1, 7, 6, 5, 4, and 8 were discarded (variables are listed in order of discarding).

Retained variables	x_2, x_3
Discarded Variables (in discarding order)	x_1, x_7, x_6, x_5, x_4, x_8

Table 3-VI Retained and discarded variables for the boiler process when multiple correlation coefficients are used as variable selection method.

The value used for R_0 was $R_0 = 0.15$, in accordance with the recommendation offered in [Jollife, 1972]. Notice that the last variable to be discarded was variable 8. Hence with a value of R_0 only slightly larger, this method would have found the same set of variables to be retained as FIR (variables 2, 3 and 8, as stated in the Appendix I.2).

With this method, a subset of variables maximally related among each other and minimally related with the set of discarded variables is obtained. The method would lie in the C option mentioned in the introduction to this chapter.

3.4.3 Variable similarity measure derived from the state transition matrix

In this subsection, a similarity measure is introduced as a way to measure the amount of common information, i.e., the linear as well as non-linear correlation, between two variables from a qualitative point of view. Although it is based on informational measures far different from the basis of the correlation indices presented in previous subsections, it is included here because it provides an estimate of the similarity between two variables.

A part of the FIR conceptual bases explained in Chapter 2, and specifically in Section 2.4, are recalled here in order to clarify the measure of correlation proposed in this sub-section. The reader may remember the basic operation of FIR:

- Quantitative data are obtained from the system in the form of real-valued variable trajectories.

- Each variable is expressed qualitatively by means of a three-component vector consisting of class, membership and side. This process is called fuzzification. In Figure 2-5, a schematic view of this process is shown.

- A mask is proposed. The class behaviour matrix and a cumulative confidence vector are obtained. Figure 2-6 shows the process of obtaining the class behaviour matrix. The matrices shown below give a generic example with three classes being used.

$$beh = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 2 \\ 3 & 1 \end{pmatrix} \qquad cum\_conf = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

- From the behaviour matrix and the cumulative confidence vector the state transition matrix is computed. The rows of this matrix represent the observed input states; its columns stand for the observed output states; and its elements denote the confidence an output state has for a given observed input state.

$$st = \begin{array}{c|ccc} in/out & '1' & '2' & '3' \\ \hline '1' & a & b & 0 \\ '2' & 0 & c & 0 \\ '3' & d & 0 & 0 \end{array}$$

- The total input confidence vector is then computed as a column vector where each element represents the sum of the corresponding row in the state transition matrix. Then, the state transition matrix and the total input confidence vector are normalised.

$$st = \begin{array}{c|ccc} in/out & '1' & '2' & '3' \\ \hline '1' & \frac{a}{a+b} & \frac{b}{a+b} & 0 \\ '2' & 0 & c/c & 0 \\ '3' & d/d & 0 & 0 \end{array} \qquad tot\_iconf = \begin{array}{c|c} '1' & (a+b)/T \\ '2' & c/T \\ '3' & d/T \end{array} \; ; T = a+b+c+d$$

- From these normalised matrices, the overall Shannon entropy of the mask, H_m, can be computed, from which, after having computed the maximum possible overall entropy H_{max}, the entropy reduction index H_r can be obtained.

$$H_m = -\left[\frac{a+b}{T}\left[\frac{a}{a+b}\log_2\left(\frac{a}{a+b}\right)+\frac{b}{a+b}\log_2\left(\frac{b}{a+b}\right)\right]+\frac{c}{T}\frac{c}{c}\log_2\left(\frac{c}{c}\right)+\frac{d}{T}\frac{d}{d}\log_2\left(\frac{d}{d}\right)\right]$$

- Finally the observation ratio O_r is computed and the quality of the studied mask obtained.

So the quality index of the mask is based on two quantities, H_r, representing an uncertainty reduction ratio, and O_r representing an observation ratio. Now we concentrate our interest on the H_r index. It is based on the Shannon entropy formula applied to the elements of the so-called state transition matrix and the total input confidence vector. The Shannon entropy measure is used to determine the uncertainty associated with predicting a particular output given any legal input state, [Shannon and Weaver, 1978]. It can be computed from the equation:

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

where *p(o|i)* is the conditional probability of a certain m-output state *o* to occur, given an m-input state *i*. The overall entropy of the proposed FIR mask is then computed as the sum:

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

where *p(i)* is the probability of that m-input state to occur. The normalised overall entropy reduction index, H_r, is then computed as

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

where H_{max} is the highest possible entropy, obtained when all observed states have the same probability to occur.

In the FIR context, the *p(o|i)* probabilities are estimated by the elements of the state transition matrix, while the *p(i)* probabilities are estimated by the elements of the total input confidence vector. In other words, each element of the state transition matrix provides a measure of confidence in an output state given an input state vector. If this matrix is generated for two variables, using a mask of the type (-1, +1), a measure of causality between the pair of variables can be derived from the corresponding H_r index, because it measures the degree of determinism associated with the cross-states of those two variables. In this case, the state transition matrix is of size $n \times m$, where n and m are the number of classes with which each of the two considered variables have been recoded (fuzzified). When two variables have a trend to be equal, their state transition matrix has a trend to be a diagonal matrix (with diagonal elements tending to 1 and all other elements tending to 0), i.e., less cross-states are found. On the other hand, when variables are uncorrelated, the state transition matrix has a trend to be a matrix where all elements have the same value (this value is the inverse of the number of classes used to recode the variable considered as m-output in the underlying analysis).

As an example, consider two variables that have been recoded using three classes. The state transition matrix obtained when a mask (-1, +1) is used may have up to nine possible different values corresponding to the nine possible states *{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)}*:

$$st = \begin{array}{c} in/out \\ '1' \\ '2' \\ '3' \end{array} \begin{array}{c} \begin{array}{ccc} '1' & '2' & '3' \end{array} \\ \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \end{array}$$

The more similar the variables are, the higher is the probability for states *{(1,1), (2,2), (3,3)}* to be large, and the higher is the probability for all other states to be small. After normalisation, the state transition matrix thus tends to an identity matrix. On the other hand, if the two variables are totally uncorrelated, the confidences in each output state given a certain input state are about the same, and consequently, the elements in each row of the state transition matrix tend to the same value, in the given example, 1/3.

When the overall entropy, H_m, is computed, the function $x \cdot log_2(x)$ has to be evaluated. The form of this function is depicted in Figure 3-4 where the function variable x assumes values in the interval [0,1].

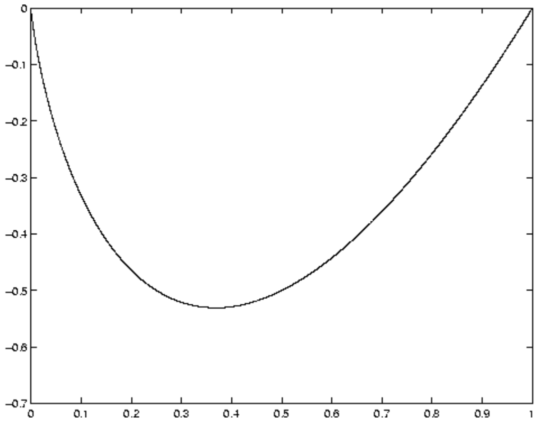


Figure 3-4 The $x \cdot log_2(x)$ function when x ε [0,1].

The H_m value represents the Shannon entropy of the system. The entropy is related to the magnitude of uncertainty of the system, i.e., it indicates how deterministic or stochastic the system under investigation is. The bigger the absolute entropy, the larger is the uncertainty associated with the output state for any given input state. With decreasing values of the absolute entropy, the input/output relationship becomes more and more deterministic. The H_r value is computed from the H_m entropy. It offers a measure of causality associated with input/output relations of the system under study. H_r is a quality measure. It is bounded between 0.0 and 1.0. Values close to 1.0 indicate highly deterministic input/output relations, whereas values close to 0.0 indicate that the uncertainty associated with the output state given any input state is almost complete, i.e., every legal value of the output state is almost equally likely to occur.

Let us now look at the case of a system with exactly one input and one output. The H_r index measures the degree of determinism of the input/output relationship. Yet for the task at hand, a stronger relationship is required. Two variables are *causally related* if the state transition matrix has one value close to 1.0 and all other values close to 0.0 in each row of its state transition matrix. Two variables are *similar* if and only if the values close to 1.0 are found along the main diagonal of the state transition matrix. Thus, two similar variables are always causally related, but two causally related variables are not necessarily similar. The H_r index is an indicator of the strength of the causal relationship between two variables, but not necessarily an indicator of their similarity.

A measure of dissimilarity, D_m, can indirectly be obtained by computing a distance between the state transition matrix and the identity matrix. This could be achieved by computing any norm of the difference between these two matrices:

$$D_m = norm(I\text{-}STM)$$

that is, the dissimilarity index D_m is the norm of the difference between the desired identity matrix and the true state transition matrix. The maximum dissimilarity is found for a totally deterministic system with one value equal to 1.0 in any non-diagonal element of each row of its state transition matrix. For example, if each of the two variables is recoded into three classes, possible state transition matrices that lead to maximum dissimilarity are:

$$STM_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad STM_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad STM_3 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

The maximum dissimilarity index, D_{max}, is computed using the same formula presented when computing D_m with any of the special STM's presented above. It has not been investigated in this dissertation, which matrix norm is most suitable for the task at hand. However, it is important to select a norm that computes the same value of dissimilarity for each of the state transition matrices above, and any other completely causal state transition matrix with 0 elements along the diagonal, and that the value obtained for these state transition matrices is indeed maximized. The most commonly used matrix norm, the L_2 norm, does not satisfy this requirement. The dissimilarity index for STM_1 and STM_2 above is different when using this norm. A good choice may be the *Frobenius norm*, which is defined as:

$$norm_F = sqrt(sum(diag(X'*X)))$$

as it indeed satisfies the requirements set above. For the example at hand,

$$D_{max} = norm_F\ [\ I(3)\ -\ STM_i\ ] = 2.4495$$

From the dissimilarity index, a similarity index, S_m, can be computed and used as a similarity measure between two variables.

$$S_m = 1.0 - \frac{D_m}{D_{max}}$$

S_m is a number in the range [0,1], where larger values denote higher similarity, i.e., S_m satisfies all the requirements to qualify as a quality measure.

Given a k variable system, a matrix of similarity indices between all pairs of variables can be obtained so as to study the common information that these variables have. This matrix is not necessarily a symmetric matrix as was the case for the sample correlation matrix obtained in previous sections. This is because the S_m index is computed using a FIR mask of the type (-1, +1). The obtained S_m index for the pair of variables (x_i, x_j) will not necessarily be the same as the one obtained for the pair (x_j, x_i), because in one case, x_i assumes the role of an m-input and x_j the role of the m-output, and vice-versa. Hence the criterion to decide if a variable is similar to another or not should be different to the one applied to the sample correlation matrix in Subsection 3.4.1.

This similarity index has been used to search for the possible redundant variables in the garbage incinerator example given in the Appendix I.3. The obtained matrix using the weighted state transition matrix is shown in Table 3-VII. As expected, the similarity matrix obtained is not symmetrical, as was the case with the correlation matrix obtained before (see Table 3-IV).

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}
X_1	1,00	0.32	0.41	0.40	0.30	0.34	0.45	0.43	**0.76**	0.51	0.42	0.53	0.40	0.39	0.31	0.41	0.41	0.51	0.45
X_2	0,32	1,00	0.45	0.43	0.36	0.44	0.52	0.37	0.41	0.56	0.63	0.52	0.62	0.36	0.33	0.33	0.34	0.38	0.43
X_3	0,41	0,44	1,00	**0.66**	0.35	0.41	0.48	0.46	0.32	0.43	0.54	0.50	0.54	0.53	0.48	0.45	0.43	0.41	0.42
X_4	0,41	0,32	**0,66**	1,00	0.35	0.38	0.47	0.49	0.36	0.32	0.54	0.50	0.53	0.53	0.50	0.48	0.47	0.40	0.43
X_5	0,30	0,45	0,46	0,45	1,00	0.31	0.38	0.32	0.28	0.43	0.37	0.35	0.29	0.43	0.36	0.35	0.36	0.42	0.36
X_6	0,34	0,55	0,55	0,52	0,39	1,00	0.50	0.49	0.33	0.34	0.55	0.49	0.52	0.58	0.46	0.47	0.47	0.42	0.49
X_7	0,44	0,37	0,54	0,50	0,37	0,30	1,00	0.46	0.44	0.46	0.63	0.32	0.62	0.47	0.42	0.43	0.49	0.41	0.50
X_8	0,43	0,44	0,46	0,49	0,31	0,37	0,29	1,00	0.38	0.35	0.48	0.35	0.61	0.62	**0.65**	0.63	0.61	0.39	0.51
X_9	**0,76**	0,61	0,37	0,36	0,29	0,36	0,37	0.38	1,00	0.39	0.42	0.49	0.48	0.33	0.31	0.32	0.35	0.41	0.46
X_{10}	0,53	0,32	0,47	0,49	0,28	0,44	0,47	0,51	0,53	1,00	0.61	0.34	0.41	0.46	0.43	0.34	0.43	0.42	0.42
X_{11}	0,42	0,53	0,54	0,52	0,37	0,29	0,39	0,46	0,44	0,47	1,00	0.44	**0.75**	0.39	0.48	0.47	0.42	0.34	0.32
X_{12}	0,54	0,61	0,53	0,53	0,29	0,39	0,48	0,44	0,49	0,34	0,59	1,00	0.61	0.47	0.44	0.44	0.45	0.46	0.44
X_{13}	0,40	0,37	0,48	0,50	0,33	0,53	0,42	0,62	0,42	0,40	**0,75**	0,61	1,00	0.35	0.42	0.40	0.42	0.42	0.32
X_{14}	0,39	0,33	0,45	0,48	0,47	0,29	0,42	**0,68**	0,33	0,35	0,46	0,44	0,46	1,00	**0.73**	**0.68**	**0.67**	0.48	0.44
X_{15}	0.37	0.33	0.42	0.47	0.46	0.38	0.43	**0.65**	0.31	0.33	0.45	0.44	0.48	**0.73**	1.00	**0.74**	**0.70**	0.38	0.53
X_{16}	0.40	0.36	0.42	0.40	0.39	0.47	0.46	**0.65**	0.33	0.44	0.43	0.43	0.44	**0.68**	**0.74**	1.00	**0.72**	0.47	0.44
X_{17}	0.41	0.39	0.41	0.42	0.36	0.37	0.44	0.44	0.35	0.33	0.42	0.45	0.41	**0.67**	**0.70**	**0.72**	1.00	0.31	0.32
X_{18}	0.51	0.43	0.42	0.39	0.32	0.42	0.41	0.51	0.47	0.41	0.45	0.43	0.43	0.41	0.31	0.44	0.44	1.00	0.42
X_{19}	0.45	0.42	0.39	0.37	0.34	0.42	0.39	0.43	0.43	0.42	0.41	0.43	0.46	0.53	0.43	0.49	0.47	0.45	1.00

Table 3-VII Input data similarity matrix for the garbage incinerator system.

The procedure to obtain this matrix is as follows: a mask of the type (-1, +1) is proposed to FIR for each pair of variables and its S_m index obtained. In the case at hand, there exist 19 input variables, so each time the proposed mask is a 19 element, vector in which there is a single –1 element in position *i*, $i = 1 \ldots 19$, and a +1 element in the *j* position, $j = 1 \ldots 19$, $i \neq j$.

Once the similarity matrix is obtained, a criterion has to be defined that identifies sets of similar variables in an unambiguous fashion. In this dissertation, those pairs with a similarity index higher than:

$$S_{m\_lim} = 0.75 \cdot (S_{m\_max} - S_{m\_min}) + \;S_{m\_min}$$

are identified as being significantly similar. The auto-similarity indices of the variables with themselves, i.e., the diagonal elements of the similarity matrix are excluded from this computation. For the incinerator system $S_{m\_lim} = 0.6476$. The pairs with $S_m \geq S_{m\_lim}$ are marked in bold in Table 3-VII. The subgroups of similar variables formed in this way are listed in Table 3-VIII.

X_1	X_9			
X_3	X_4			
X_{11}	X_{13}			
X_8	X_{14}	X_{15}	X_{16}	X_{17}

Table 3-VIII Subgroups of similar variables.

These results can be compared with those showed in Table 3-III. The obtained subgroups are very similar, although small differences exist between the two sets of subgroups.

3.5 Conclusions

A first FIR model search space reduction has been achieved by selecting a subset of input variables for time delay 0, i.e., only static relations have been taken into account. The purpose was to select a subset of input variables that best explain a given output variable, i.e., from which the output variable can best be reconstructed.

Different known statistical techniques have been compared for their aptitude to identify sets of related variables. Most of these techniques are based on either principal component analysis or correlation-based metrics. It was found that successful methods select input variables that are strongly correlated with the selected output variable yet are weakly correlated among each other. Most of the competitive approaches first identify sets of potential input variables that are strongly correlated among each other, then select one or more variables from each of these sets to reconstruct the output, by throwing out those variables from each set that are least correlated with the selected output variable.

In Subsection 3.4.3, a variable similarity measure is derived from the state transition matrix. The S_m index determines the similarity between pairs of variables, so subsets of variables of a system that are similar can be constructed. The primary advantage of this method over its traditional statistical counterparts lies in the fact that the state transition matrix recognizes similarity between variables irrespective of the precise nature of similarity, i.e., it does not rely on linearity. In this way, a better variable reduction can be achieved, because potential input variables that are strongly related to each other in a non-linear way can be recognized as belonging to the same cluster, although their linear correlation may be quite small, such that approaches that are based on linear analysis would place them in different clusters.

4 Subsystem determina tion

4.0 Abstract

Chapter 3 was devoted to the selection of a set of variables that can best be used to model (reconstruct) a given output variable, whereby only static relations were analysed. Yet even after reducing the set of variables in this fashion, the number of remaining variables may still be formidable for large-scale systems. The present chapter aims at tackling this problem by discovering substructures within the whole set of the system variables. Hence whereas the previous chapter dealt with the problem of *model reduction* by means of reducing the set of variables to be considered for modelling, the present chapter focuses on *model structuring* as a means to subdividing the overall modelling task into subtasks that are hopefully easier to handle. Section 4.2 analyses this problem from a system-theoretic perspective, presenting the *Reconstruction Analysis* methodology, an informational approach to the problem of decomposing a large-scale system into subsystems. Section 4.3 uses FIR to find a possible structure of a system. Section 4.4 deals with a singular-value decomposition method to form subsets/subsystems of linearly related variables. Section 4.5 proposes a procedure to study the possible non-linear relations between a given subset of variables and any additional variable. Again the study performed in this chapter only considers static relations.

4.1 Introduction

Up to this point, the dissertation looked at *variable selection* techniques with the aim of identifying a subset of variables that are best suited for reconstructing (modelling) a given output variable. These methods can be classified as *model reduction* techniques as their primary purpose is to aid the modeller in simplifying the resulting model of the output variable. These techniques serve as a precursor to the FIR modelling engine. Their aim is to reduce the FIR search space to make the FIR methodology better suited for dealing with large-scale systems. Yet even if the number of variables can be reduced dramatically by variable selection techniques, the set of remaining variables to be considered by the FIR modelling engine may still be far too large to allow the engine to converge within reasonable computation time.

Therefore a second precursor to FIR modelling should be performed: using those variables that are retained in the previously performed variable selection step for modelling the system, subsets of variables need to be found that are maximally related among each other. These subsets can then be used to determine sub-models of the overall model. The techniques pertaining to this second step, to be discussed in the present chapter, can be classified as *model structuring* techniques. The corresponding algorithms are often referred to in the literature as *structure identification* algorithms [Klir, 1985].

When analysing the variable selection approaches of Chapter 3, it turned out that several of the most successful approaches decomposed this problem into two stages. In a first stage, they determined groups of variables that are strongly correlated among each other, and in a second stage, they chose representative subsets of variables from each of these groups to model the selected output variable. The techniques discussed in Chapter 4 are related to the first of the two subtasks of Chapter 3, i.e., there is inevitably a certain methodological overlap between these two chapters. Most of the clustering techniques discussed in Chapter 3 could also serve for the task at hand in Chapter 4, and vice-versa. Yet the focus of the two chapters is distinct: whereas Chapter 3 concentrated on model reduction, Chapter 4 aims at model structuring.

A question arises here: why is it desirable to represent a complex system by means of a collection of subsystems?

Several arguments may be stated that support the concept of subdividing systems into sets of interconnected subsystems. First, it may be either impractical or impossible to measure all the variables relevant to the modelling task at equal frequencies. Some variables may change over time much more slowly than others, and consequently, should be sampled (measured) less frequently. If the data are collected in groups of related variables forming part of the same subsystem[1], it is only necessary to measure each of those groups consistently and coherently. Second, even if all of the data were sampled at equal frequencies, storing all of them in a single table is impractical, because all combinations of states would then need to be recorded, even for variables that are almost uncorrelated with one another. The superset of all legal states of the overall system is much larger than the concatenation of sets of all legal states of all subsystems. Another important reason is related to the easiness of the process design when a subsystem decomposition of the complex system is available. This is a very common feature used in engineering design. The subdivision of the overall system into parts enables the designer to find solutions for each of these parts separately then connect the individual designs to obtain a (usually sub-optimal) design for the overall system.

In the context of FIR, the primary motivation for subdividing a system into parts is similar to the last argument mentioned above. Let us assume that data are available to capture the dynamics of the entire system to be modelled. A subdivision of the system into parts (a subdivision of the set of variables into subsets) simplifies the FIR modelling task, and often makes an otherwise intractable problem tractable. Once the FIR models of the subsystems have been found, the behaviour of the overall system can be reconstructed from the individual behaviours of its parts.

This idea has its roots in the fact that, given a k-variables system, the cost of computing a unique k-variable model is much higher than computing a set of p models of $j_p < k$ variables. Different authors have tackled this problem in the past from an information systems point of view. In [Ashby, 1964], given a system with a large number of variables, a first analysis is performed that helps to reduce the number of relations between variables by means of imposing restrictions. Then in [Ashby, 1965], it is stated that complex systems are unmanageable and should be decomposed into subsystems. Also in this work, some of

[1] The reader may notice that, in general, the subsystems obtained by means of a decomposition method do not necessarily coincide with the physical subsystems of the complex system being investigated.

the formulae measuring the information exchange between variables are presented. In [Conant, 1972; 1976], the measure *T* of energy transference between variables is used to determine a probable subsystem decomposition. In [Broekstra, 1976-77], the information theory introduced previously by Ashby is used to decompose systems using modified procedures coined 'constraint analysis' by the author. In [Madden and Ashby, 1972], relations between variables are offered that define the representation of an *N*-variable system in terms of a set of *P*-variables subsystems, where *P<N*. In fact, it is that study which lays the foundations of what later would be introduced by Cavallo and Klir as the *Reconstructability Analysis* [Cavallo and Klir, 1979a]. [Gaines, 1977] offers a different yet parallel point of view to that of Klir's General System Problem Solving (GSPS) approach. The problem of structure identification, i.e., identification of subsystems, is tackled in [Broekstra, 1978]. In [Kengerlinskiy, 1978], an approach for decomposing a system into subsystems based on the Shannon entropy measure is provided. In [Krippendorff, 1978; 1981], yet another technique for determining the structure of a system is given. He also provides a comparison of his approach to those of Klir, Conant, and Broekstra.

Every one of these techniques requires that all of the available variables must be present in the set of obtained subsystems. Hence all of these techniques focus on the problem of model structuring rather than that of model reduction.

In this chapter, three different methods for obtaining subsets (subsystems) of variables are given. First in Section 4.2, the *Reconstruction Analysis (RA)* methodology, a GSPS[2] level-4 tool, is presented. The basic idea of RA is that of identifying, among all possible decompositions of a system, the one from which the behaviour of the overall system can best be reconstructed. The best decomposition is the one that leads to a reconstructed system, the behaviour of which is most similar to the original system behaviour. One of the requirements of this methodology is that all system variables are present in the set of obtained subsystems, classifying RA as a model structuring technique.

In Section 4.3, a FIR-based approach to model structuring is presented. Starting from the desired output variable, sequences of sub-models are constructed, until all variables are accounted for as either input variables or internal variables.

The third presented method is explained in Sections 4.4 and 4.5. In those sections, a methodology to decompose a complex system into subsystems is explained as a mixed model reduction and structuring approach. The proposed methodology is based on a combination of linear and non-linear statistical techniques. After performing a variable selection using the technique explained in Section 3.4.1, a singular value decomposition of the remaining variables of the correlation matrix is performed. This second step, discussed in Section 4.4, allows obtaining subsets of linearly related variables by means of the projection of the eigenvectors of this decomposition onto the principal axes.

While forming subsets of linearly related variables, it may happen that some of the variables retained after variable selection do not form part of any of the subsets. This is because the projection of those variables onto the principal axes is weak, i.e., they are not

[2] A brief description of the epistemological levels in the mark of Klir's GSPS framework is provided in Chapter 2 of this dissertation.

linearly related to the other variables. At this point, a possible non-linear relation between these excluded variables and the obtained subsets should be investigated. This is the topic tackled in Section 4.5. It forms the third stage of the proposed methodology: to search, via a non-linear transformation, the possible non-linear correlation between the variables in the subsets and those not present in any of them.

As this was the case in the previous chapter, all three methodologies presented in this chapter only deal with models of zero time delay, i.e., they only consider static relations among variables. The inclusion of time in the analysis, and the study of dynamic relations among the variables are left for Chapter 5 to deal with. In this upcoming chapter, the different methodologies presented along Chapters 3 and 4 are put together with the FIR methodology so as to propose two different FIR-based full modelling methodologies that tackle jointly the problems of dynamic model reduction and structuring.

Also in Chapter 4, like in all other chapters, the functioning of the different algorithms presented along the chapter is illustrated by means of applying them to realistic examples. The example used most widely in this chapter is the garbage incinerator system described in Appendix I.3. By using the same example across multiple model structuring methodologies, the functioning of the set of algorithms becomes more transparent, and the reader is able to acquire a deeper understanding of the methodological underpinnings of the proposed algorithms.

4.2 Reconstruction Analysis

Reconstruction analysis is a GSPS level-4 tool developed by Cavallo and Klir in the early eighties. It tackles two complementary problems associated with the relationship between a global system and the various sets of its subsystems. The first problem is referred to in the literature as the *reconstruction problem*. Given a global system, the aim is to determine which structure systems, each one based on a set of subsystems of the overall system, are adequate for reconstructing the global system with an acceptable level of approximation. This reconstruction problem is the one that most interests us in this dissertation.

The other problem is what has been called in the literature the *identification problem*. That is, given a set of subsystems by means of their behaviours, known to form part of a global system, to find the possible global systems that embrace those subsystems, so inferences about the unknown global system can be made.

Reconstruction Analysis, encompassing the two aforementioned problems, originated with the Reconstructability Analysis proposed in [Cavallo and Klir, 1979a,b; Cavallo and Klir 1981; Klir 1981]. The methodology was refined in [Klir, 1991; Cellier, 1991; de Albornoz, 1996]. Reconstruction analysis is closely related to the FIR methodology[3], because it deals with the system behaviour information expressed in qualitative terms that are either crisp or fuzzy. It allows identifying temporal causal structures of a system, i.e., it

[3] Reconstruction Analysis is fed with the same qualitative data used in the FIR methodology. Many concepts used in this chapter have already been explained in detail in chapter 2.

can be used to determine a subsystem decomposition of a system. Yet in practice, as it will be shown during the presentation of the Reconstruction Analysis tool, it is not useful for systems with more than say a dozen variables due to its exponential computational complexity.

4.2.1 The concepts of Reconstructability

Usually a model of a system is understood as a set of rules mapping a set of variables onto each other [Klir, 1985]. If an input-output model of a system is desired, this set of rules, assuming a MISO system, maps all the related input (and possibly internal) variables to the considered output so that the behaviour of the system can be obtained from past data and this set of mapping rules. This is basically what the FIR methodology does, and it lies on level 3 of the GSPS framework.

This kind of reasoning may be impractical in the presence of large numbers of variables. In this case, it would be better to obtain a description of the internal structure of the system in the form of rules that are used inside the model to map variables onto each other. To find the structure of the system, in the most abstract case, means to search for a subsystem decomposition of this system, then obtain a model for each one of these subsystems separately, and finally integrate the information of those subsystems in a way that describes the global system. This is what the RA methodology strives to accomplish.

Reconstructability Analysis is conceived as a package of methodological tools within the GSPS framework [Klir, 1985] that deals with the problem of defining the relationship between a global system and its various subsystems. This problem involves two kinds of epistemological levels[4]: the generative system and the structure system, both usually represented by means of their behaviour. As briefly stated before, two main problems are tackled within this set of tools: the identification problem, in which the structure system, i.e., a set of subsystems, is initially given, whereas the global system is unknown; and the reconstruction problem, in which the global system is known, and the subsystems are to be determined.

The tools needed to solve the identification problem are:

- Synthesis of the *reconstruction family* for a given set of subsystems. A reconstruction family is defined as the set of global systems that may be represented by the given set of subsystems in the sense that the behaviour of these subsystems may be obtained as a projection of the overall behaviour.

- Determination of the *reconstruction uncertainty*, which is also referred to as *identifiability quotient*. This metric, based on the size of the reconstruction family, is used to determine the uncertainty associated with the reconstruction of the global system from the given set of subsystems. Such a metric can be derived for either probabilistic or possibilistic systems.

[4] The epistemological levels of Klir have been briefly explained in Section 2.2. A quick reference to these levels can be found in Figure 2-3.

- Identification of the *unbiased reconstruction*. A unique global system is selected from the previously generated reconstruction family. The goal is to select a global system that contains, using Klir's words, '*all, but no more information than is contained in the structure system.*' Unbiased here means that, from the global behaviour, the known set of subsystem behaviours should be obtained when projecting the global behaviour onto the set of variables forming each of the subsystems. If local inconsistencies are detected in this step (in the sense that the behaviour projections do not totally coincide with the original subsystem projections), a tool is required to solve those inconsistencies.

In the case of the reconstruction problem, the tools needed are:

- Generation of *reconstruction hypotheses* for a given global system. By reconstruction hypothesis is meant a possible set of subsystems from which the behaviour of the global system can be reconstructed.

- A computation of the desirable projections of the given behaviour system.

- Computation of the distances between the given behaviour and those reconstructed from each of the reconstruction hypotheses. This distance computation will allow establishing a rank order among the different considered reconstruction hypotheses.

Now a brief description of how these tools have been implemented will be provided by means of a synthetic example.

4.2.2 Structure systems in Reconstruction Analysis

As previously explained, the aim of the presented methodology is to find a plausible subsystem decomposition of a complex system. In the sequel, it will be shown, how the methodology works. Consider, for example, a model, *M*, of a six variables system that is formed from two sub-models, M_1 and M_2. Figure 4-1 shows the *topological structure* of this system.

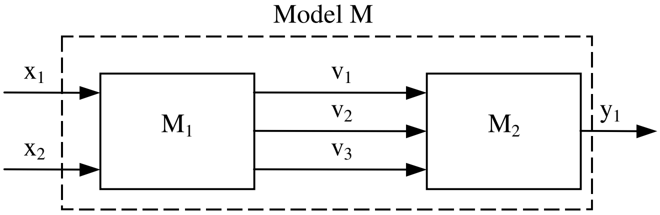


Figure 4-1 A model with two possible subsystems.

In the given example, the model is formed by six variables: two input variables, x_1 and x_2, that are the inputs of both the model M and the first submodel M_1; three internal variables, named v_1, v_2, and v_3, that are the outputs of the first submodel M_1 as well as the inputs to the second submodel M_2; and finally, one output variable, y_1, that is the output of the second submodel M_2 as well as of the model M.

The structure can be abstracted in RA by means of a so-called *composite structure*. A composite structure is a row vector, in which each element enumerates a variable of a subsystem, and different substructures are separated by 0 elements. In the given example of Figure 4-1, the composite structure would be expressed as:

$$c\_st1 = (1\ 2\ 3\ 4\ 5\ 0\ 3\ 4\ 5\ 6)$$

where the variables have been labelled from 1 to 6, beginning with x_1 and finishing with y_1 (x_1=1, x_2=2, v_1=3, v_2=4, v_3=5, y_1=6).

The reader may notice that, at this level of the methodology, no distinction is made any longer between inputs and outputs. Variables are simply labelled with an integer number and enumerated one by one as they are encountered in a submodel without distinguishing between inputs and outputs; thus, no causality is expressed by the composite structure[5]. The abstraction of a topological structure into a composite structure is unique and straightforward. The opposite is not true. A single composite structure may be representative of zero, one, or multiple topological structures.

Another way of expressing these kinds of structures, also used in RA, is by means of the so-called *binary structure*. It consists of an ordered enumeration of all the possible binary relations or connections among variables inside all subsystems. In the given example of Figure 4-1, the corresponding binary structure is:

$$b\_st1 = \begin{pmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \\ 2 & 3 \\ 2 & 4 \\ 2 & 5 \\ 3 & 4 \\ 3 & 5 \\ 3 & 6 \\ 4 & 5 \\ 4 & 6 \\ 5 & 6 \end{pmatrix}$$

[5] The causality of a mathematical model of a physical system is an artefact of the way in which models are commonly used by simulation code, rather than being a property of the physical system itself. Hence it is meaningful to offer a mathematical description of the system structure that reflects physical reality by not forcing the user to provide causality information.

The reader may notice that, of all possible connections among the six system variables, only two relations are missing, namely the connections (1,6) and (2,6), indicating that there do not exist direct connections between variables 1, 2 (the input variables of the system) on the one hand, and variable 6 (the output variable of the system) on the other.

There exist two special cases, the totally unconnected model and the totally connected model. The former is a model that contains a set of unrelated variables. Since there are no connections among variables, its binary structure is empty. The latter is a model without any internal structure. Every variable is related to every other variable. Consequently, its binary structure is complete.

The two forms of expressing an a-causal structure system presented so far are essentially equivalent and may be mapped into one another by means of two of the algorithms existing in the RA software package facility. Both representations co-exist in the methodology for reasons of convenience. The composite structure is more compact and easier interpretable by the human researcher. The binary structure is more convenient for algorithmic purposes, i.e., when using the structure information within higher-level algorithms and when comparing different structures to each other.

Yet, there exists a problem with the mapping of one structure into the other. While the transition from a composite structure to a binary structure is unambiguous and straightforward, the inverse operation is more difficult and does not necessarily lead to a unique composite structure. Consider, for instance, the topological structure represented in Figure 4-2, corresponding to a system with two inputs and two outputs decomposed into three subsystems.

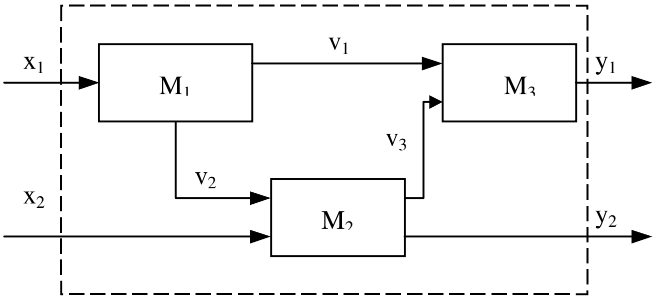


Figure 4-2 Seven-variable system with a model consisting of three submodels.

If the variables in Figure 4-2 are labelled from 1 to 7 beginning with x_1 and ending with y_2, (x_1=1, x_2=2, v_1=3, v_2=4, v_3=5, y_1=6, y_2=7), the composite structure corresponding to this system is expressed as:

$$c\_st2 = (1\ 3\ 4\ 0\ 3\ 5\ 6\ 0\ 2\ 4\ 5\ 7)$$

and its corresponding binary structure is:

$$b\_st2 = \begin{pmatrix} 1 & 3 \\ 1 & 4 \\ 2 & 4 \\ 2 & 5 \\ 2 & 7 \\ 3 & 4 \\ 3 & 5 \\ 3 & 6 \\ 4 & 5 \\ 4 & 7 \\ 5 & 6 \\ 5 & 7 \end{pmatrix}$$

Now consider the composite structure,

$$c\_st3 = (1\ 3\ 4\ 0\ 3\ 4\ 5\ 0\ 2\ 4\ 5\ 7\ 0\ 3\ 5\ 6)$$

which may be representative of the topological structure system depicted in Figure 4-3. The variables have been labelled in the same order as in the previously presented structure.

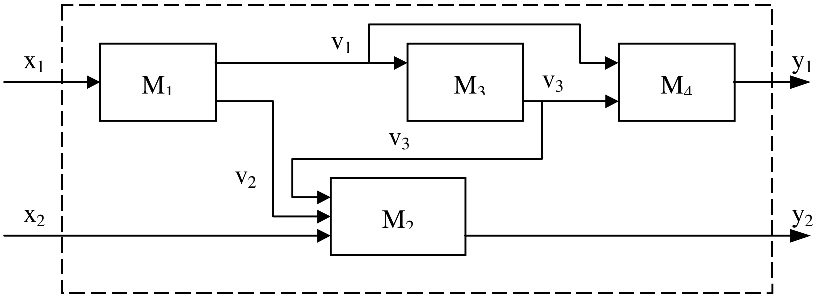


Figure 4-3 Seven-variable system with a model consisting of four submodels.

The binary structure corresponding to the composite structure *c_st3* is exactly the same as that found for the composite structure *c_st2*, labelled as *b_st2*. Thus the same binary structure *b_st2* can be mapped into either the composite structure *c_st2* or the composite structure *c_st3*. Hence the mapping of a binary structure into a composite structure is not always unique.

Looking at Figure 4-3 more closely, the reader may notice that submodels M_1, M_2, and M_4 are *essential* submodels. For example, the submodel M_1 is the only submodel referencing variable x_1. If submodel M_1 were eliminated, variable x_1 would no longer be part of any submodel. For similar reasons, the submodels M_2 and M_4 must be retained. In contrast, submodel M_3 references only variables that also form part of other submodels. Hence submodel M_3 is redundant and can be eliminated. The algorithm employed by RA for mapping binary structures into composite structures always identifies the *minimal* composite structure corresponding to a given binary structure, eliminating all redundant subsystems.

4.2.3 Tools available in Reconstruction Analysis

Given an observed behaviour of a system, the first step on the way to solving the structure identification problem is the generation of reconstruction hypotheses. In RA, the behaviour is defined in exactly the same fashion as it was done for the FIR methodology[6]. A reconstruction hypothesis is simply a proposed composite structure over the set of variables contained in the behaviour. The process of generating sets of reconstruction hypotheses shall be explained in due course. For now, let us concentrate on the algorithm used to evaluate the merits of a given reconstruction hypothesis.

Once a reconstruction hypothesis has been formulated, the behaviour of the global system is projected onto each of the subsystems obtaining the subsystem behaviours. The behaviours of the individual subsystems are then recombined, leading to a reconstructed overall behaviour. The quality of the reconstruction is evaluated by measuring the distance between the original system behaviour and the reconstructed one. Figure 4-4 illustrates this process.

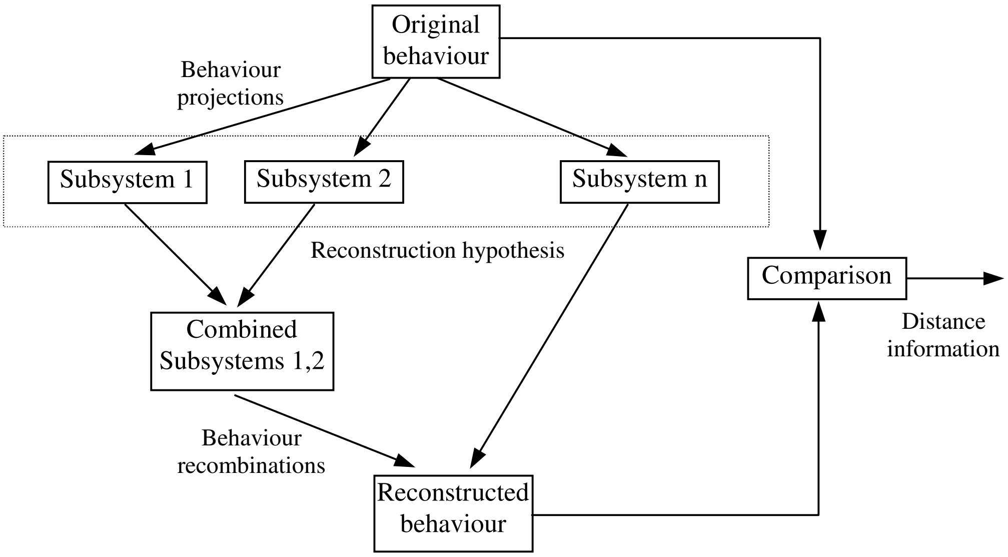


Figure 4-4 Reconstruction evaluation process.

Reconstruction analysis, just like FIR, can be applied to either crisp or fuzzy behaviours. In this dissertation, only the case of *Fuzzy Reconstruction Analysis (FRA)* shall be considered. The reader is referred to [de Albornoz, 1996; Cellier, 1991] for a discussion of Crisp Reconstruction Analysis.

Let us consider a global system expressed by means of its behaviour, B, and a confidence vector, c:

[6] The behaviour is expressed by means of a matrix that contains the ordered set of all observed states of the system together with a confidence vector expressing the likelihood of each state to occur.

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad c = \begin{pmatrix} 0.18 \\ 0.13 \\ 0.1 \\ 0.04 \\ 0.05 \\ 0.11 \\ 0.24 \\ 0.15 \end{pmatrix}$$

The given synthetic example consists of six variables. Only eight different system states have been recorded. The confidence vector is here normalized, such that the sum of all confidences is 1.0. In this case, the confidence of a state can be interpreted as an estimate of the probability of its occurrence.

Let us further consider the following reconstruction hypothesis:

$$c\_st4 = (1\ 2\ 3\ 4\ 5\ 0\ 3\ 4\ 5\ 6)$$

which was previously depicted in Figure 4-1.

The result of the fuzzy projection of the global behaviour onto the first subsystem space spanned by variables x_1, x_2, v_1, v_2, and v_3, is:

$$B1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad c1 = \begin{pmatrix} 0.18 \\ 0.13 \\ 0.1 \\ 0.04 \\ 0.05 \\ 0.11 \\ 0.24 \\ 0.15 \end{pmatrix}$$

To this end, variable y_1 is simply eliminated from the behaviour. Since the resulting states are all different, nothing more needs to be done.

The result of projecting the global behaviour onto the second subsystem subspace spanned by variables v_1, v_2, v_3, and y_1, is:

$$B2 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \qquad c2 = \begin{pmatrix} 0.15 \\ 0.18 \\ 0.13 \\ 0.1 \\ 0.24 \end{pmatrix}$$

After eliminating variables x_1 and x_2 from the behaviour and after reordering the states, it can be seen that several states are now listed multiple times with different confidence values. These states need to be collapsed, such that each state is listed only once in the subsystem behaviour. In FIR, the accumulated confidence of multiple observations of the same state had been defined as the sum of the individual confidences. In FRA, a different definition of accumulated confidence is being used. Here, the accumulated confidence is defined as the largest among the individual confidences [Cavallo and Klir, 1982]. This definition was chosen in order to avoid distortions in the metric used to evaluate the merit of each reconstruction hypothesis [Cellier and de Albornoz, 1998]. A consequence of this definition is that the subsystem confidence values can no longer be interpreted as estimates of the probability of occurrence, since the sum of the confidence values no longer adds up to 1.0.

The decomposition of the given global system now needs to be used to reconstruct the global behaviour. To this end, the subsystem behaviours have to be recombined. When the two behaviours, *B1* and *B2* are recombined, the following global behaviour is obtained:

$$B\_rec = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad c\_rec = \begin{pmatrix} 0.18 \\ 0.13 \\ 0.1 \\ 0.1 \\ 0.04 \\ 0.05 \\ 0.05 \\ 0.11 \\ 0.1 \\ 0.24 \\ 0.15 \end{pmatrix}$$

The way to recombine two different behaviours is as follows: if the two behaviours have no variable in common, i.e., they are unconnected, then the recombined behaviour is the superset of all combinations of behaviours of the two subsystems. Common variables, i.e., connections between the two subsystems, act as constraints that reduce the freedom of combining the two behaviours. In the given example, the variables v_1, v_2, and v_3 are common to both subsystems. Consequently, the first state of subsystem *B1*: [0 0 0 0 1] can only be recombined with states of *B2* that begin with [0 0 1]. There exists only one such state: [0 0 1 0], hence there only results a single recombined state from this combination: [0 0 0 0 1 0]. When the structure contains more than two subsystems, the recombination occurs in pairs, as shown in Figure 4-4. The reconstructed behaviour contains all states of the original behaviour, and possibly a few more.

How is the confidence of a recombined state computed? Here, FIR and FRA use the same formula: the confidence of a recombined state is computed as the smaller of the two confidences involved in the recombination [Cavallo and Klir, 1982]. If the structure has more than two subsystems, the recombined state, at the end of all recombinations, assumes the smallest value of the individual confidences involved in the recombination.

As mentioned above, there exists the possibility for new states to appear in the reconstructed global behaviour, states that have never been observed. In the given example, three new states have appeared, namely [0 0 1 0 1 1], [0 1 1 0 1 1], and [1 0 1 0 1 1].

Now a comparison of the obtained reconstructed global behaviour with the original global system behaviour has to be made in order to determine the quality of the obtained decomposition. For each of the two global behaviours, the so-called *ambiguity measure* is computed [Cavallo and Klir, 1982]. The two ambiguity measures are obtained as follows: Starting with the original behaviour, the smallest confidence value found in its confidence vector is divided by the number of observed states having a confidence larger or equal to this value. Added to this result is another term, which is the quotient of the distance between the smallest and the next larger confidence value divided by the number of states that exhibit a confidence larger or equal to the second lowest confidence. More terms are added in the same fashion, whereby the numerator is always the difference between the next higher confidence value and the previously considered one, and the denominator is always the number of states exhibiting a confidence larger or equal to the currently considered confidence. The resulting value is then subtracted from 1. For the given example, the ambiguity of the original behaviour is computed as follows:

$$a = 1 - \left(\frac{0.04}{8} + \frac{0.01}{7} + \frac{0.05}{6} + \frac{0.01}{5} + \frac{0.02}{4} + \frac{0.02}{3} + \frac{0.03}{2} + \frac{0.06}{1} \right) = 1 - 0.1034 = 0.8965$$

In an analogous manner, the ambiguity measure of the reconstructed global behaviour is computed as:

$$a\_rec = 1 - \left(\frac{0.04}{11} + \frac{0.01}{10} + \frac{0.05}{8} + \frac{0.01}{5} + \frac{0.02}{4} + \frac{0.02}{3} + \frac{0.03}{2} + \frac{0.06}{1} \right) = 1 - 0.09955 = 0.9004$$

Finally, the measure of distance, or error, between the original behaviour and the reconstructed one is defined as the difference between the two ambiguity measures:

$$Rec\_error = a\_rec - a = 0.0039$$

The ambiguity formula used expresses the closeness between the original behaviour and the reconstructed one in terms of the differences between the information that the reconstructed system contains in comparison with the original one. Therefore, it expresses the closeness between the behaviours in terms of the loss of information that takes place when the reconstructed behaviour replaces the original behaviour. The formula presented here to compute the distance measure (reconstruction measure) is a special case of the general distance computation formula for possibilistic systems [Klir, 1985]:

$$D(f, f^h) = \frac{1}{\log_2 |C|} \int_0^1 \log_2 \frac{|c(f^h, l)|}{|c(f, l)|} dl$$

In this formula, the used notation is:

f – behaviour function of the given system (confidence)
f^h – behaviour function of the reconstructed function
c(f,l) – the confidence of a given state to occur
C – the set of possible states

4.2.4 Algorithms to generate r econstruction hypotheses

In the previous section, the question of how reconstruction hypotheses are being generated was left open. The present section offers a response to this question. In order to find the optimal subsystem decomposition using FRA, all possible sets of subsystems of a given global system should be computed. This is a formidable task, feasible only for small-scale systems comprised of few variables. The number of possible subsystem decompositions grows exponentially with the number of variables. In fact, it grows even faster than the number of possible masks in the case of FIR. Three sub-optimal search algorithms have been reported in the open literature [Uyttenhove, 1978; Klir and Uyttenhove, 1979; Cavallo and Klir, 1979b, 1981b] that usually generate satisfactory system decompositions.

The first of these algorithms has been called *structure refinement*. This algorithm starts out with the totally connected binary structure, i.e., all possible binary relations between the system variables are considered. For the given example, the totally connected composite structure is defined as:

$$c\_st\_tot\_c = (1\ 2\ 3\ 4\ 5\ 6)$$

Evidently, the totally connected structure has no reconstruction error, since no reconstruction needs to be done. Then, at each step, one binary connection is severed at a time, and the reconstruction error for the resulting subsystem decomposition is computed. When for the present level of decomposition all binary connections have been severed one at a time, the one that results in the structure with the smallest reconstruction error is permanently removed. At the first step of the algorithm, the number of reconstructions that is to be computed in a *k*-variable system is given by:

$$\frac{k(k-1)}{2}$$

Then a second binary relation is severed. The algorithm follows the same algorithm, severing one binary relation at a time, until the one with the smallest reconstruction error has been found. The corresponding binary relation is then deleted permanently. The algorithm continues in the same fashion, until the reconstruction error of all of the candidate structures at the next level becomes larger than the limit imposed by the investigator.

In the current implementation of FRA, a quality measure based on the reconstruction error is introduced.

$$Q_i = 1 - \frac{rec\_err_i}{rec\_err_{tot\_unc}}$$

where rec_err_i is the reconstruction error associated with the i^{th} reconstruction hypothesis, and $rec\_err_{tot\_unc}$ is the reconstruction error of the totally unconnected structure, which exhibits the largest reconstruction error possible. For the given example, the totally unconnected structure is defined as:

c_st_tot_unc = (1 0 2 0 3 0 4 0 5 0 6)

The quality measure is a metric normalized to the range [0,1], where higher quality values denote better reconstruction hypotheses.

When the structure refinement algorithm is applied to the example at hand with a smallest tolerated quality of $Q_{min} = 0.1$, the following result is obtained:

```
STRUCTURE (Q = 1.0000)  (1,2,3,4,5,6)

 BEST STRUCTURE ON THIS LEVEL: (1,2,3,4,5,6)
 THE BEST QUALITY FOUND IS: 1.0000
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.9855)  (1,3,4,5,6) (2,3,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,4,5,6) (2,3,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,3,5,6) (2,3,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,3,4,6) (2,3,4,5,6)
STRUCTURE (Q = 0.9111)  (1,2,3,4,5) (2,3,4,5,6)
STRUCTURE (Q = 0.9808)  (1,2,4,5,6) (1,3,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,3,5,6) (1,3,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,3,4,6) (1,3,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,3,4,5) (1,3,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,3,5,6) (1,2,4,5,6)
STRUCTURE (Q = 0.8364)  (1,2,3,4,6) (1,2,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,3,4,5) (1,2,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,3,4,6) (1,2,3,5,6)
STRUCTURE (Q = 0.8364)  (1,2,3,4,5) (1,2,3,5,6)
STRUCTURE (Q = 1.0000)  (1,2,3,4,5) (1,2,3,4,6)

 BEST STRUCTURE ON THIS LEVEL: (1,2,4,5,6) (2,3,4,5,6)
 THE BEST QUALITY FOUND IS: 1.0000
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.9855)  (1,4,5,6) (2,3,4,5,6)
STRUCTURE (Q = 0.8364)  (1,2,5,6) (2,3,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,4,6) (2,3,4,5,6)
STRUCTURE (Q = 0.6010)  (1,2,4,5) (2,3,4,5,6)
STRUCTURE (Q = 0.9808)  (3,4,5,6) (1,2,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,5,6) (1,4,5,6) (2,3,5,6) (3,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,4,6) (1,4,5,6) (2,3,4,6) (3,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,4,5) (1,4,5,6) (2,3,4,5) (3,4,5,6)
STRUCTURE (Q = 0.8364)  (2,3,5,6) (1,2,4,5,6)
STRUCTURE (Q = 0.8364)  (2,3,4,6) (1,2,4,5,6)
STRUCTURE (Q = 0.6702)  (2,3,4,5) (1,2,4,5,6)
STRUCTURE (Q = 0.9061)  (1,2,4,6) (1,2,5,6) (2,3,4,6) (2,3,5,6)
STRUCTURE (Q = 0.7395)  (1,2,4,5) (1,2,5,6) (2,3,4,5) (2,3,5,6)
STRUCTURE (Q = 0.9061)  (1,2,4,5) (1,2,4,6) (2,3,4,5) (2,3,4,6)

 BEST STRUCTURE ON THIS LEVEL: (1,2,4,6) (2,3,4,5,6)
 THE BEST QUALITY FOUND IS: 1.0000
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.9739)  (1,4,6) (2,3,4,5,6)
STRUCTURE (Q = 0.7825)  (1,2,6) (2,3,4,5,6)
STRUCTURE (Q = 0.5425)  (1,2,4) (2,3,4,5,6)
STRUCTURE (Q = 0.9808)  (1,2,4,6) (2,4,5,6) (3,4,5,6)
STRUCTURE (Q = 0.9739)  (1,2,6) (1,4,6) (2,3,5,6) (3,4,5,6)
STRUCTURE (Q = 1.0000)  (1,2,4,6) (2,3,4,6) (3,4,5,6)
STRUCTURE (Q = 0.9518)  (1,2,4) (1,4,6) (2,3,4,5) (3,4,5,6)
STRUCTURE (Q = 0.8364)  (1,2,4,6) (2,3,5,6) (2,4,5,6)
STRUCTURE (Q = 0.8364)  (1,2,4,6) (2,3,4,6) (2,4,5,6)
STRUCTURE (Q = 0.6702)  (1,2,4,6) (2,3,4,5) (2,4,5,6)
STRUCTURE (Q = 0.8364)  (1,2,4,6) (2,3,4,6) (2,3,5,6)
STRUCTURE (Q = 0.7044)  (1,2,4) (1,2,6) (2,3,4,5) (2,3,5,6)
STRUCTURE (Q = 0.8364)  (1,2,4,6) (2,3,4,5) (2,3,4,6)

 BEST STRUCTURE ON THIS LEVEL: (1,2,4,6) (2,3,4,6) (3,4,5,6)
 THE BEST QUALITY FOUND IS: 1.0000
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.9739)  (1,4,6) (2,3,4,6) (3,4,5,6)
STRUCTURE (Q = 0.7825)  (1,2,6) (2,3,4,6) (3,4,5,6)
STRUCTURE (Q = 0.5425)  (1,2,4) (2,3,4,6) (3,4,5,6)
STRUCTURE (Q = 0.7419)  (1,2,4,6) (3,4,5,6)
STRUCTURE (Q = 0.9644)  (1,2,6) (1,4,6) (2,3,6) (3,4,5,6)
STRUCTURE (Q = 0.8543)  (1,2,4) (1,4,6) (2,3,4) (3,4,5,6)
STRUCTURE (Q = 0.7746)  (2,3,6) (3,5,6) (4,5,6) (1,2,4,6)
STRUCTURE (Q = 0.6351)  (4,5,6) (1,2,4,6) (2,3,4,6)
STRUCTURE (Q = 0.5134)  (2,3,4) (3,4,5) (4,5,6) (1,2,4,6)
STRUCTURE (Q = 0.7746)  (3,5,6) (1,2,4,6) (2,3,4,6)
STRUCTURE (Q = 0.4982)  (1,2,4) (1,2,6) (3,4,5) (3,5,6)
STRUCTURE (Q = 0.6351)  (3,4,5) (1,2,4,6) (2,3,4,6)

 BEST STRUCTURE ON THIS LEVEL: (1,4,6) (2,3,4,6) (3,4,5,6)
 THE BEST QUALITY FOUND IS: 0.9739
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.7474)  (1,6) (2,3,4,6) (3,4,5,6)
STRUCTURE (Q = 0.4776)  (1,4) (2,3,4,6) (3,4,5,6)
STRUCTURE (Q = 0.7273)  (1,4,6) (2,4,6) (3,4,5,6)
STRUCTURE (Q = 0.8850)  (1,4,6) (2,3,6) (3,4,5,6)
STRUCTURE (Q = 0.7273)  (1,4,6) (2,3,4) (3,4,5,6)
STRUCTURE (Q = 0.7571)  (1,4,6) (2,3,6) (2,4,6) (3,5,6) (4,5,6)
STRUCTURE (Q = 0.6134)  (1,4,6) (4,5,6) (2,3,4,6)
STRUCTURE (Q = 0.5002)  (1,4,6) (2,3,4) (2,4,6) (3,4,5) (4,5,6)
STRUCTURE (Q = 0.7571)  (1,4,6) (3,5,6) (2,3,4,6)
STRUCTURE (Q = 0.6337)  (1,4) (1,6) (2,3,4) (2,3,6) (3,4,5) (3,5,6)
STRUCTURE (Q = 0.6177)  (1,4,6) (3,4,5) (2,3,4,6)

 BEST STRUCTURE ON THIS LEVEL: (1,4,6) (2,3,6) (3,4,5,6)
 THE BEST QUALITY FOUND IS: 0.8850
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.6810)  (1,6) (2,3,6) (3,4,5,6)
STRUCTURE (Q = 0.4460)  (1,4) (2,3,6) (3,4,5,6)
STRUCTURE (Q = 0.6642)  (2,6) (1,4,6) (3,4,5,6)
STRUCTURE (Q = 0.6642)  (2,3) (1,4,6) (3,4,5,6)
STRUCTURE (Q = 0.7177)  (1,4,6) (2,3,6) (3,5,6) (4,5,6)
STRUCTURE (Q = 0.5745)  (1,4,6) (2,3,6) (3,4,6) (4,5,6)
STRUCTURE (Q = 0.4585)  (2,3) (2,6) (1,4,6) (3,4,5) (4,5,6)
STRUCTURE (Q = 0.6855)  (1,4,6) (2,3,6) (3,4,6) (3,5,6)
STRUCTURE (Q = 0.6149)  (1,4) (1,6) (2,3,6) (3,4,5) (3,5,6)
STRUCTURE (Q = 0.5783)  (1,4,6) (2,3,6) (3,4,5)

 BEST STRUCTURE ON THIS LEVEL: (1,4,6) (2,3,6) (3,5,6) (4,5,6)
 THE BEST QUALITY FOUND IS: 0.7177
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.5710)  (1,6) (2,3,6) (3,5,6) (4,5,6)
STRUCTURE (Q = 0.3458)  (1,4) (2,3,6) (3,5,6) (4,5,6)
STRUCTURE (Q = 0.5119)  (2,6) (1,4,6) (3,5,6) (4,5,6)
STRUCTURE (Q = 0.5337)  (2,3) (1,4,6) (3,5,6) (4,5,6)
STRUCTURE (Q = 0.5427)  (1,4,6) (2,3,6) (4,5,6)
STRUCTURE (Q = 0.4205)  (2,3) (2,6) (3,5) (1,4,6) (4,5,6)
STRUCTURE (Q = 0.6586)  (1,4,6) (2,3,6) (3,5,6)
STRUCTURE (Q = 0.5855)  (1,4) (1,6) (4,5) (2,3,6) (3,5,6)
STRUCTURE (Q = 0.5461)  (3,5) (4,5) (1,4,6) (2,3,6)
```

```
 BEST STRUCTURE ON THIS LEVEL: (1,4,6) (2,3,6) (3,5,6)
 THE BEST QUALITY FOUND IS: 0.6586
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.5311)  (1,6) (4,6) (2,3,6) (3,5,6)
STRUCTURE (Q = 0.3209)  (1,4) (4,6) (2,3,6) (3,5,6)
STRUCTURE (Q = 0.4710)  (2,6) (1,4,6) (3,5,6)
STRUCTURE (Q = 0.4866)  (2,3) (1,4,6) (3,5,6)
STRUCTURE (Q = 0.4744)  (5,6) (1,4,6) (2,3,6)
STRUCTURE (Q = 0.3698)  (2,3) (2,6) (3,5) (5,6) (1,4,6)
STRUCTURE (Q = 0.5606)  (1,4) (1,6) (2,3,6) (3,5,6)
STRUCTURE (Q = 0.4964)  (3,5) (1,4,6) (2,3,6)

 BEST STRUCTURE ON THIS LEVEL: (1,4) (1,6) (2,3,6) (3,5,6)
 THE BEST QUALITY FOUND IS: 0.5606
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.5168)  (1,6) (2,3,6) (3,5,6) (4)
STRUCTURE (Q = 0.3023)  (1,4) (2,3,6) (3,5,6)
STRUCTURE (Q = 0.3883)  (1,4) (1,6) (2,6) (3,5,6)
STRUCTURE (Q = 0.3945)  (1,4) (1,6) (2,3) (3,5,6)
STRUCTURE (Q = 0.3925)  (1,4) (1,6) (5,6) (2,3,6)
STRUCTURE (Q = 0.2760)  (1,4) (1,6) (2,3) (2,6) (3,5) (5,6)
STRUCTURE (Q = 0.4011)  (1,4) (1,6) (3,5) (2,3,6)

 BEST STRUCTURE ON THIS LEVEL: (1,6) (2,3,6) (3,5,6) (4)
 THE BEST QUALITY FOUND IS: 0.5168
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.2550)  (2,3,6) (3,5,6) (1) (4)
STRUCTURE (Q = 0.3510)  (1,6) (2,6) (3,5,6) (4)
STRUCTURE (Q = 0.3588)  (1,6) (2,3) (3,5,6) (4)
STRUCTURE (Q = 0.3538)  (1,6) (5,6) (2,3,6) (4)
STRUCTURE (Q = 0.2379)  (1,6) (2,3) (2,6) (3,5) (5,6) (4)
STRUCTURE (Q = 0.3646)  (1,6) (3,5) (2,3,6) (4)

 BEST STRUCTURE ON THIS LEVEL: (1,6) (3,5) (2,3,6) (4)
 THE BEST QUALITY FOUND IS: 0.3646
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.1671)  (3,5) (2,3,6) (1) (4)
STRUCTURE (Q = 0.2614)  (1,6) (2,6) (3,5) (3,6) (4)
STRUCTURE (Q = 0.2468)  (1,6) (2,3) (3,5) (3,6) (4)
STRUCTURE (Q = 0.3022)  (1,6) (2,3,6) (4) (5)
STRUCTURE (Q = 0.2063)  (1,6) (2,3) (2,6) (3,5) (4)

 BEST STRUCTURE ON THIS LEVEL: (1,6) (2,3,6) (4) (5)
 THE BEST QUALITY FOUND IS: 0.3022
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.1266)  (2,3,6) (1) (4) (5)
STRUCTURE (Q = 0.2049)  (1,6) (2,6) (3,6) (4) (5)
STRUCTURE (Q = 0.2130)  (1,6) (2,3) (3,6) (4) (5)
STRUCTURE (Q = 0.1522)  (1,6) (2,3) (2,6) (4) (5)

 BEST STRUCTURE ON THIS LEVEL: (1,6) (2,3) (3,6) (4) (5)
 THE BEST QUALITY FOUND IS: 0.2130
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.0713)  (2,3) (3,6) (1) (4) (5)
STRUCTURE (Q = 0.1784)  (1,6) (3,6) (2) (4) (5)
STRUCTURE (Q = 0.1257)  (1,6) (2,3) (4) (5)

 BEST STRUCTURE ON THIS LEVEL: (1,6) (3,6) (2) (4) (5)
 THE BEST QUALITY FOUND IS: 0.1784
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.0504)  (3,6) (1) (2) (4) (5)
STRUCTURE (Q = 0.0858)  (1,6) (2) (3) (4) (5)

 BEST STRUCTURE ON THIS LEVEL: (1,6) (2) (3) (4) (5)
 THE BEST QUALITY FOUND IS: 0.0858
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.0000)  (1) (2) (3) (4) (5) (6)

BEST STRUCTURE ON THIS LEVEL HAS QUALITY 0.0000
CONSIDERED TO LOW. SEARCH FINISHED

RESULTANT STRUCTURE: (1,6) (2) (3) (4) (5)
```

The second algorithm found in the open literature is the so-called *structure aggregation* algorithm. The strategy followed here is just the opposite of the one followed in the previously described algorithm. In this case, the algorithm starts out with the totally unconnected model, and at each time step, one binary connection is added. From all possible structure candidates, the one that most reduces the reconstruction error is considered the best, i.e., the binary relation that, when added, most reduces the reconstruction error is added permanently. The algorithm ends when the reconstruction error drops below the largest tolerable error specified by the modeller. For the given example, this algorithm, with $Q_{min} = 0.1$, leads to the following results:

```
STRUCTURE (Q = 0.0000)  (1) (2) (3) (4) (5) (6)

 BEST STRUCTURE ON THIS LEVEL: (1) (2) (3) (4) (5) (6)
 THE BEST QUALITY FOUND IS: 0.0000
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.0245)  (1,2) (3) (4) (5) (6)
STRUCTURE (Q = 0.0504)  (1,3) (2) (4) (5) (6)
STRUCTURE (Q = 0.0204)  (1,4) (2) (3) (5) (6)
STRUCTURE (Q = 0.0259)  (1,5) (2) (3) (4) (6)
STRUCTURE (Q = 0.0858)  (1,6) (2) (3) (4) (5)
STRUCTURE (Q = 0.0245)  (2,3) (1) (4) (5) (6)
STRUCTURE (Q = 0.0204)  (2,4) (1) (3) (5) (6)
STRUCTURE (Q = 0.0463)  (2,5) (1) (3) (4) (6)
STRUCTURE (Q = 0.0245)  (2,6) (1) (3) (4) (5)
STRUCTURE (Q = 0.0204)  (3,4) (1) (2) (5) (6)
STRUCTURE (Q = 0.0313)  (3,5) (1) (2) (4) (6)
STRUCTURE (Q = 0.0504)  (3,6) (1) (2) (4) (5)
STRUCTURE (Q = 0.0204)  (4,5) (1) (2) (3) (6)
STRUCTURE (Q = 0.0204)  (4,6) (1) (2) (3) (5)
STRUCTURE (Q = 0.0313)  (5,6) (1) (2) (3) (4)

 BEST STRUCTURE ON THIS LEVEL: (1,6) (2) (3) (4) (5)
 THE BEST QUALITY FOUND IS: 0.0858
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.1348)  (1,2) (1,6) (3) (4) (5)
STRUCTURE (Q = 0.1865)  (1,3) (1,6) (2) (4) (5)
STRUCTURE (Q = 0.1021)  (1,4) (1,6) (2) (3) (5)
STRUCTURE (Q = 0.1375)  (1,5) (1,6) (2) (3) (4)
STRUCTURE (Q = 0.1257)  (1,6) (2,3) (4) (5)
STRUCTURE (Q = 0.1130)  (1,6) (2,4) (3) (5)
STRUCTURE (Q = 0.1693)  (1,6) (2,5) (3) (4)
STRUCTURE (Q = 0.1185)  (1,6) (2,6) (3) (4) (5)
STRUCTURE (Q = 0.1130)  (1,6) (3,4) (2) (5)
STRUCTURE (Q = 0.1348)  (1,6) (3,5) (2) (4)
STRUCTURE (Q = 0.1784)  (1,6) (3,6) (2) (4) (5)
STRUCTURE (Q = 0.1130)  (1,6) (4,5) (2) (3)
STRUCTURE (Q = 0.1021)  (1,6) (4,6) (2) (3) (5)
STRUCTURE (Q = 0.1239)  (1,6) (5,6) (2) (3) (4)

BEST STRUCTURE OF THIS LEVEL HAS QUALITY: 0.1865
CONSIDERED ENOUGH. SEARCH FINISHED

RESULTANT STRUCTURE: (1,3) (1,6) (2) (4) (5)
```

The structures resulting from the application of the two algorithms are not identical, but similar. In the given example, the structure aggregation algorithm converged much faster, because the selected minimum quality was very low. If a higher value of Q_{min} had been selected, the structure refinement algorithm would have converged faster, whereas the structure aggregation algorithm would have required more time to reach the desired goal.

Finally, the third of the known sub-optimal search strategies is the so-called *single-step refinement* algorithm. This algorithm, just like the refinement algorithm, starts out with the totally connected structure. The first step of the single-step refinement algorithm is identical to that of the refinement algorithm. However, at the end of the first step, all relations are permanently severed that exhibit a reconstruction error that is smaller than the largest tolerated one, or in the current implementation, that exhibit a quality that is better than the lowest permissible one. Only a single step of the algorithm is performed. This algorithm operates on the reconstruction errors of individual omitted binary relations rather than on the quality of the resulting structure. Whereas the refinement algorithm and the aggregation algorithm are still of exponential complexity (though much faster than exhaustive search), the single-step refinement algorithm is of polynomial complexity.

When this algorithm is applied to the given example, allowing a largest reconstruction error of $e_{max} = 0.005$, the following results are obtained:

```
ERROR      BINARY RELATION OMITTED

0.000556          1 , 2
0.000000          1 , 3
0.000000          1 , 4
0.000000          1 , 5
0.003401          1 , 6
0.000734          2 , 3
0.000000          2 , 4
0.000000          2 , 5
0.000000          2 , 6
0.000000          3 , 4
0.006258          3 , 5
0.000000          3 , 6
0.000000          4 , 5
0.006258          4 , 6
0.000000          5 , 6

BINARY RELATIONS WITH AN ERROR LARGER THAN 0.0050 ARE CONSIDERED

RESULTANT STRUCTURE: (3,5) (4,6) (1) (2)
```

The FRA tools explained in this chapter were previously implemented by Adelinde Uhrmacher with improvements and refinements by Alvaro de Albornoz [de Albornoz, 1996]. The first version of the algorithms was programmed in Fortran, and the maximum number of variables allowed was ten. Hence the available code was not suitable for large-scale systems, in spite of the fact that at least the single-step refinement algorithm converges acceptably fast also for more than 10 variables. When searching for methods to decompose a complex system into subsystems, and investigating if FRA could be applied to this task, the algorithms presented here have been reviewed and recoded in the frame of this dissertation as Matlab functions using dynamic memory allocation. The new implementation is unlimited in the number of variables, which, at least in theory, enabled this researcher to apply the algorithms to his problem: the identification of structures in large-scale systems.

4.2.5 Application of Reconstruction Analysis to a real system

The FRA methodology has been applied to the steam generator that is described in the Appendix I.2. This is a case of a MISO system with nine variables.

In order to apply the previously described algorithms, it is necessary to first compute the behaviour matrix and the confidence vector for this system. This can be done using some of the internal functions of the FIR methodology that has been described in detail in Chapter 2. For this boiler system, all variables of which have been recoded into three classes, the observed behaviour (computed from the 632 data records gathered by the system sensors) has been found to have 164 different states. Some of those states, together with their computed confidence values, have been represented in the following table.

$$Boiler\_behaviour = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 3 \\ \vdots & & & & \vdots & & & & \vdots \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 3 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 1 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{pmatrix} \qquad conf = \begin{pmatrix} 18.7297 \\ 3.1882 \\ 2.6631 \\ 1.0059 \\ 0.5472 \\ .5007 \\ .5018 \\ \vdots \\ 1.0044 \\ 1.0000 \\ 4.3160 \\ 18.1925 \\ 17.0665 \end{pmatrix}$$

The reader may notice that the confidence values in the above table are non-normalized. The individual confidence values of an observation are real-valued numbers in the range [0.5,1.0]. FIR computes its accumulated confidence by adding up the confidence values of individual observations. Thus, a confidence value of 18.7297 in the above table denotes the fact that this particular state has been observed between 19 and 38 times. Although the confidence vector could have been easily normalized, this was not necessary. FRA works just as well with non-normalized confidence values as with normalized ones, since it operates on min/max norms and relative magnitude (ordering information) only.

The three procedures that were previously presented have been used to obtain decompositions of the system. Two of the methods, the single-step refinement algorithm and the structure aggregation algorithm, were run completely, whereas the third method, the structure refinement algorithm, was used only during some steps because of its computational complexity.

Let us begin with the first of the previously explained methods, the structure refinement algorithm. For this run, a minimum quality of $Q = 0.80$ was specified. The experiment was executed using the newly re-implemented FRA tools running in the Matlab environment in an interpretive fashion. The code was executed on a Sun UltraSparc-II workstation. As mentioned before, only four steps, i.e., four levels of structure refinement, were computed for this run, so the final result only excludes four binary relations. The reason for only allowing four steps to be computed is the required computation time. In a system with 9 variables, there exist 36 different binary relations. Hence at the first level, 36 possible structures need to be studied; at the second level, 35 possible structures need to be investigated; and so on. The algorithm, although sub-optimal, is of exponential complexity, and even with this small-scale example, it takes a lot of computation time to calculate all the necessary projections and behaviour recombinations by executing Matlab code in an interpretive fashion. A summary of the results obtained from this experiment is presented below.

```
STRUCTURE (Q = 1.0000)  (1,2,3,4,5,6,7,8,9)

 BEST STRUCTURE IN THIS LEVEL: (1,2,3,4,5,6,7,8,9)
 BEST QUALITY FOUND IS: 1.0000
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 1.0000)  (1,3,4,5,6,7,8,9) (2,3,4,5,6,7,8,9)
STRUCTURE (Q = 1.0000)  (1,2,4,5,6,7,8,9) (2,3,4,5,6,7,8,9)
STRUCTURE (Q = 1.0000)  (1,2,3,5,6,7,8,9) (2,3,4,5,6,7,8,9)
STRUCTURE (Q = 1.0000)  (1,2,3,4,6,7,8,9) (2,3,4,5,6,7,8,9)
STRUCTURE (Q = 1.0000)  (1,2,3,4,5,7,8,9) (2,3,4,5,6,7,8,9)

                                      ...

STRUCTURE (Q = 0.9971)  (1,2,3,4,5,6,7,8) (1,2,3,4,5,6,7,9)

 BEST STRUCTURE IN THIS LEVEL: (1,2,3,4,6,7,8,9) (2,3,4,5,6,7,8,9)
 BEST QUALITY FOUND IS: 1.0000
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 1.0000)  (1,3,4,6,7,8,9) (2,3,4,5,6,7,8,9)
STRUCTURE (Q = 1.0000)  (1,2,4,6,7,8,9) (2,3,4,5,6,7,8,9)
STRUCTURE (Q = 0.9999)  (1,2,3,6,7,8,9) (2,3,4,5,6,7,8,9)

                                      ...

STRUCTURE (Q = 0.9971)  (1,2,3,4,6,7,8) (1,2,3,4,6,7,9) (2,3,4,5,6,7,8)
(2,3,4,5,6,7,9)

 BEST STRUCTURE IN THIS LEVEL: (1,3,4,6,7,8,9) (2,3,4,5,6,7,8,9)
 BEST QUALITY FOUND IS: 1.0000
 SEARCH CONTINUES WITH THIS STRUCTURE
```

```
STRUCTURE (Q = 0.9999)  (1,4,6,7,8,9) (2,3,4,5,6,7,8,9)
STRUCTURE (Q = 0.9999)  (1,3,6,7,8,9) (2,3,4,5,6,7,8,9)
STRUCTURE (Q = 0.9999)  (1,3,4,7,8,9) (2,3,4,5,6,7,8,9)
STRUCTURE (Q = 0.9998)  (1,3,4,6,8,9) (2,3,4,5,6,7,8,9)

                                           ...

STRUCTURE (Q = 0.9999)  (1,3,4,6,7,8) (1,3,4,6,8,9) (2,3,4,5,6,7,8)
(2,3,4,5,6,8,9)
STRUCTURE (Q = 0.9971)  (1,3,4,6,7,8) (1,3,4,6,7,9) (2,3,4,5,6,7,8)
(2,3,4,5,6,7,9)

 BEST STRUCTURE IN THIS LEVEL: (1,3,4,7,8,9) (2,3,4,5,6,7,8,9)
 BEST QUALITY FOUND IS: 0.9999
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.9998)  (1,4,7,8,9) (2,3,4,5,6,7,8,9)
STRUCTURE (Q = 0.9999)  (1,3,7,8,9) (2,3,4,5,6,7,8,9)
STRUCTURE (Q = 0.9998)  (1,3,4,8,9) (2,3,4,5,6,7,8,9)

                                           ...

STRUCTURE (Q = 0.9999)  (1,3,4,7,8) (1,3,4,8,9) (2,3,4,5,6,7,8)
(2,3,4,5,6,8,9)
STRUCTURE (Q = 0.9971)  (1,3,4,7,8) (1,3,4,7,9) (2,3,4,5,6,7,8)
(2,3,4,5,6,7,9)

 BEST STRUCTURE IN THIS LEVEL: (1,3,4,7,9) (1,3,7,8,9) (2,3,4,5,6,7,9)
(2,3,5,6,7,8,9)
 BEST QUALITY FOUND IS: 0.9999
 SEARCH CONTINUES WITH THIS STRUCTURE
```

The four binary relations removed by the refinement procedure are (1,2), (1,5), (1,6) and (4,8), and the quality of the reconstructed system is still very high, *Q = 0.9999*.

The next procedure to be applied is the structure aggregation method. In this case, the run was completely performed. The termination condition for this algorithm was to find a structure system with a minimum quality of *Q = 0.80*. A summary of the results obtained when using this algorithm is presented below:

```
STRUCTURE (Q = 0.0000)  (1) (2) (3) (4) (5) (6) (7) (8) (9)

 BEST STRUCTURE IN THIS LEVEL: (1) (2) (3) (4) (5) (6) (7) (8) (9)
 BEST QUALITY FOUND IS: 0.0000
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.0039)  (1,2) (3) (4) (5) (6) (7) (8) (9)
STRUCTURE (Q = 0.0024)  (1,3) (2) (4) (5) (6) (7) (8) (9)
STRUCTURE (Q = 0.0039)  (1,4) (2) (3) (5) (6) (7) (8) (9)
                                           ...
STRUCTURE (Q = 0.0039)  (7,8) (1) (2) (3) (4) (5) (6) (9)
STRUCTURE (Q = 0.0037)  (7,9) (1) (2) (3) (4) (5) (6) (8)
STRUCTURE (Q = 0.0037)  (8,9) (1) (2) (3) (4) (5) (6) (7)

 BEST STRUCTURE IN THIS LEVEL: (1,7) (2) (3) (4) (5) (6) (8) (9)
 BEST QUALITY FOUND IS: 0.0039
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.0119)  (1,2) (1,7) (3) (4) (5) (6) (8) (9)
STRUCTURE (Q = 0.0087)  (1,3) (1,7) (2) (4) (5) (6) (8) (9)
STRUCTURE (Q = 0.0120)  (1,4) (1,7) (2) (3) (5) (6) (8) (9)
```

```
                                     ...

STRUCTURE (Q = 0.0118)  (1,7) (7,8) (2) (3) (4) (5) (6) (9)
STRUCTURE (Q = 0.0113)  (1,7) (7,9) (2) (3) (4) (5) (6) (8)
STRUCTURE (Q = 0.0113)  (1,7) (8,9) (2) (3) (4) (5) (6)

 BEST STRUCTURE IN THIS LEVEL: (1,4) (1,7) (2) (3) (5) (6) (8) (9)
 BEST QUALITY FOUND IS: 0.0120
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.0288)  (1,2) (1,4) (1,7) (3) (5) (6) (8) (9)
STRUCTURE (Q = 0.0217)  (1,3) (1,4) (1,7) (2) (5) (6) (8) (9)
STRUCTURE (Q = 0.0287)  (1,4) (1,5) (1,7) (2) (3) (6) (8) (9)

                                     ...

STRUCTURE (Q = 0.0286)  (1,4) (1,7) (7,8) (2) (3) (5) (6) (9)
STRUCTURE (Q = 0.0270)  (1,4) (1,7) (7,9) (2) (3) (5) (6) (8)
STRUCTURE (Q = 0.0270)  (1,4) (1,7) (8,9) (2) (3) (5) (6)

 BEST STRUCTURE IN THIS LEVEL: (1,4) (1,7) (6,7) (2) (3) (5) (8) (9)
 BEST QUALITY FOUND IS: 0.0289
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.0648)  (1,2) (1,4) (1,7) (6,7) (3) (5) (8) (9)
STRUCTURE (Q = 0.0483)  (1,3) (1,4) (1,7) (6,7) (2) (5) (8) (9)
STRUCTURE (Q = 0.0647)  (1,4) (1,5) (1,7) (6,7) (2) (3) (8) (9)
STRUCTURE (Q = 0.0289)  (1,4) (1,6,7) (2) (3) (5) (8) (9)

                                     ...

STRUCTURE (Q = 0.0596)  (1,4) (1,7) (6,7) (7,9) (2) (3) (5) (8)
STRUCTURE (Q = 0.0596)  (1,4) (1,7) (6,7) (8,9) (2) (3) (5)

 BEST STRUCTURE IN THIS LEVEL: (1,4) (1,7) (2,4) (6,7) (3) (5) (8) (9)
 BEST QUALITY FOUND IS: 0.0648
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.0648)  (1,7) (6,7) (1,2,4) (3) (5) (8) (9)
STRUCTURE (Q = 0.1039)  (1,3) (1,4) (1,7) (2,4) (6,7) (5) (8) (9)
STRUCTURE (Q = 0.1433)  (1,4) (1,5) (1,7) (2,4) (6,7) (3) (8) (9)
STRUCTURE (Q = 0.0648)  (1,4) (2,4) (1,6,7) (3) (5) (8) (9)

                                     ...

STRUCTURE (Q = 0.1282)  (1,4) (1,7) (2,4) (6,7) (7,9) (3) (5) (8)
STRUCTURE (Q = 0.1282)  (1,4) (1,7) (2,4) (6,7) (8,9) (3) (5)

 BEST STRUCTURE IN THIS LEVEL: (1,4) (1,7) (2,4) (5,6) (6,7) (3) (8) (9)
 BEST QUALITY FOUND IS: 0.1433
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.1433)  (1,7) (5,6) (6,7) (1,2,4) (3) (8) (9)
STRUCTURE (Q = 0.2219)  (1,3) (1,4) (1,7) (2,4) (5,6) (6,7) (8) (9)
STRUCTURE (Q = 0.1433)  (1,4) (1,5) (1,7) (2,4) (5,6) (6,7) (3) (8) (9)

                                     ...

STRUCTURE (Q = 0.2756)  (1,4) (1,7) (2,4) (5,6) (6,7) (7,9) (3) (8)
STRUCTURE (Q = 0.2756)  (1,4) (1,7) (2,4) (5,6) (6,7) (8,9) (3)

 BEST STRUCTURE IN THIS LEVEL: (1,4) (1,7) (2,4) (5,6) (6,7) (6,8) (3) (9)
 BEST QUALITY FOUND IS: 0.3161
 SEARCH CONTINUES WITH THIS STRUCTURE
```

```
STRUCTURE (Q = 0.3162)  (1,7) (5,6) (6,7) (6,8) (1,2,4) (3) (9)
STRUCTURE (Q = 0.4748)  (1,3) (1,4) (1,7) (2,4) (5,6) (6,7) (6,8) (9)
STRUCTURE (Q = 0.3162)  (1,4) (1,5) (1,7) (2,4) (5,6) (6,7) (6,8) (3) (9)
STRUCTURE (Q = 0.3162)  (1,4) (2,4) (5,6) (6,8) (1,6,7) (3) (9)

                                        ...

STRUCTURE (Q = 0.5975)  (1,4) (1,7) (2,4) (5,6) (6,7) (6,8) (7,9) (3)
STRUCTURE (Q = 0.5975)  (1,4) (1,7) (2,4) (5,6) (6,7) (6,8) (8,9) (3)

 BEST STRUCTURE IN THIS LEVEL: (1,4) (1,7) (1,9) (2,4) (5,6) (6,7) (6,8) (3)
 BEST QUALITY FOUND IS: 0.5975
 SEARCH CONTINUES WITH THIS STRUCTURE

STRUCTURE (Q = 0.5975)  (1,7) (1,9) (5,6) (6,7) (6,8) (1,2,4) (3)
STRUCTURE (Q = 0.9252)  (1,3) (1,4) (1,7) (1,9) (2,4) (5,6) (6,7) (6,8)
STRUCTURE (Q = 0.5975)  (1,4) (1,5) (1,7) (1,9) (2,4) (5,6) (6,7) (6,8) (3)
STRUCTURE (Q = 0.5975)  (1,4) (1,9) (2,4) (5,6) (6,8) (1,6,7) (3)

                                        ...

STRUCTURE (Q = 0.5976)  (1,4) (1,7) (1,9) (2,4) (5,6) (6,7,8) (3)
STRUCTURE (Q = 0.5975)  (1,4) (2,4) (5,6) (6,7) (6,8) (1,7,9) (3)
STRUCTURE (Q = 0.5975)  (1,4) (1,7) (1,9) (2,4) (5,6) (6,7) (6,8) (8,9) (3)

BEST STRUCTURE IN THIS LEVEL HAS QUALITY: 0.9642
CONSIDERED ENOUGH. SEARCH FINISHES

RESULTANT STRUCTURE: (1,4) (1,7) (1,9) (2,4) (3,9) (5,6) (6,7) (6,8)
```

The structure aggregation procedure terminates when the quality of the reconstruction is higher than the prescribed limit. In this case, the quality found is *Q = 0.9642* for a decomposition in which only two-variable subsystems have been formed.

Finally, the results obtained when using the single-step refinement algorithm are given:

```
ERROR   BINARY RELATION OMITTED

0.000037       1 , 2
0.000108       1 , 3
0.000092       1 , 4
0.000000       1 , 5
0.000090       1 , 6
0.000232       1 , 7
0.000125       1 , 8
0.000142       1 , 9
0.002886       2 , 3
0.000232       2 , 4
0.000274       2 , 5
0.000368       2 , 6
0.000177       2 , 7
0.000555       2 , 8
0.002532       2 , 9
0.000290       3 , 4
0.004500       3 , 5
0.001133       3 , 6
0.000225       3 , 7
0.008466       3 , 8
0.275960       3 , 9
0.000227       4 , 5
0.000213       4 , 6
0.000090       4 , 7
```

```
0.000166        4 , 8
0.000166        4 , 9
0.000754        5 , 6
0.000126        5 , 7
0.001280        5 , 8
0.004249        5 , 9
0.000159        6 , 7
0.000503        6 , 8
0.000992        6 , 9
0.000193        7 , 8
0.000091        7 , 9
0.012071        8 , 9

BINARY RELATIONS WITH AN ERROR LARGER THAN 0.1000 ARE CONSIDERED

RESULTANT STRUCTURE: (3,9) (1) (2) (4) (5) (6) (7) (8)
```

If only those relations that, when removing them, give a reconstruction error larger than 0.1 are considered, a single two-variable subsystem composed by variables 3 and 9 is encountered. Notice that these results make a lot of sense, as the FIR boiler model computed in Appendix I.2 only uses input variables 2, 3, and 8 to model the output (variable 9). If the limit for the reconstruction error is being decreased to 0.01, two subsystems are formed: (3,9) and (8,9). If the limit is being decreased further to 0.008, a single three-variable structure (3,8,9) is found. The next variable to be considered is variable 4. Variable 2 follows shortly afterwards. The FRA single-step refinement algorithm found a good decomposition for this boiler system within an acceptable computing time. Neither of the other two algorithms offered good results within reasonable computation time.

Note than even with a small-scale system, such as the nine-variable boiler system, the FRA procedures were not able to converge within reasonable time, except for the single-step refinement algorithm, which indeed arrived at a good result. When the FRA techniques were applied to the garbage incinerator system, a 20-variable MISO system, even the single-step refinement algorithm involved substantial computations leading to long waiting times. Of course, if a truly complex system, consisting of some hundred measured variables, were encountered, application of even the single-step refinement algorithm of FRA to obtain a decomposition of the system using the current Matlab implementation would surely require considerably more computation time than the practical life-span of any computer. This makes the available code impractical for the purposes of this dissertation.

A compiled C-code implementation of the single-step refinement algorithm might execute within acceptable time limits, but its implementation was not attempted in this dissertation, as the algorithms presented in the sequel gave good results for the considered test cases, while being much more computationally efficient. The author recommends a C-code implementation of the *single-step refinement* algorithm in a future research effort, as FRA is expected to exploit non-linear correlations in a more optimal fashion than the algorithms presented in the sequel.

4.2.6 Application of Reconstruction Analysis in conjunction with a variable selection technique

In this section, FRA is applied to the incinerator system (described in Appendix I.3), but only involving the 15 variables remaining from the correlation analysis presented in section 3.4.1. In that subsection, redundant variables, in a linear sense, were dropped out of a possible posterior modelling process. Using that result, it is possible to reduce the number of structures that FRA needs to investigate to determine the internal structure of the system. In this way, the computing time of the FRA engine can be significantly reduced when searching for a plausible structure within the set of system variables.

The variables discarded in Section 3.4.1 were X_1, X_3, X_{15}, X_{16}, and X_{17}, so 14 input variables remain for the analysis. The output variable needs to be added to the group of variables to be analysed, thus 15 variables need to be considered. Due to the size of the system, only the single-step refinement algorithm has been used.

With the available data from the incinerator system, the single-step refinement algorithm required roughly 11 hours of computing time on a Sun Ultra-Sparc II. The results from this experiment are listed below:

```
ERROR   BINARY RELATION OMITTED

0.000121     2 , 4
0.000031     2 , 5
0.000024     2 , 6
0.000011     2 , 7
0.011050     2 , 8
0.014203     2 , 9
0.000042     2 , 10
0.021082     2 , 11
0.016028     2 , 12
0.027111     2 , 13
0.010561     2 , 14
0.008524     2 , 18
0.000286     2 , 19
0.009398     2 , 20
0.000033     4 , 5
0.028183     4 , 6
0.012103     4 , 7
0.013151     4 , 8
0.012272     4 , 9
0.016134     4 , 10
0.014261     4 , 11
0.025047     4 , 12
0.013029     4 , 13
0.008572     4 , 14
0.006091     4 , 18
0.016835     4 , 19
0.005103     4 , 20
0.000067     5 , 6
0.000026     5 , 7
0.013272     5 , 8
0.000661     5 , 9
0.000017     5 , 10
0.021411     5 , 11
0.000019     5 , 12
0.000011     5 , 13
```

```
0.015141      5 , 14
0.004083      5 , 18
0.000075      5 , 19
0.011106      5 , 20
0.024196      6 , 7
0.022142      6 , 8
0.015137      6 , 9
0.000028      6 , 10
0.000001      6 , 11
0.023185      6 , 12
0.014601      6 , 13
0.000084      6 , 14
0.000079      6 , 18
0.031102      6 , 19
0.000174      6 , 20
0.021566      7 , 8
0.012323      7 , 9
0.000367      7 , 10
0.000221      7 , 11
0.022214      7 , 12
0.016339      7 , 13
0.020031      7 , 14
0.013191      7 , 18
0.021365      7 , 19
0.014242      7 , 20
0.023871      8 , 9
0.018399      8 , 10
0.000102      8 , 11
0.000115      8 , 12
0.006032      8 , 13
0.023143      8 , 14
0.000282      8 , 18
0.018324      8 , 19
0.018102      8 , 20
0.000067      9 , 10
0.014634      9 , 11
0.015811      9 , 12
0.018401      9 , 13
0.014753      9 , 14
0.000297      9 , 18
0.000214      9 , 19
0.000216      9 , 20
0.008464      10 , 11
0.015678      10 , 12
0.018031      10 , 13
0.013597      10 , 14
0.023159      10 , 18
0.000157      10 , 19
0.021783      10 , 20
0.014957      11 , 12
0.019011      11 , 13
0.021164      11 , 14
0.008869      11 , 18
0.008357      11 , 19
0.008176      11 , 20
0.024819      12 , 13
0.031104      12 , 14
0.000116      12 , 18
0.011067      12 , 19
0.000258      12 , 20
0.023164      13 , 14
0.000103      13 , 18
0.000037      13 , 19
0.000135      13 , 20
0.021305      14 , 18
```

```
0.000109      14 , 19
0.000099      14 , 20
0.002862      18 , 19
0.004012      18 , 20
0.002243      19 , 20

BINARY RELATIONS WITH AN ERROR LARGER THAN 0.1000 ARE CONSIDERED


RESULTANT STRUCTURE: (2) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14)
(18) (19) (20)




BINARY RELATIONS WITH AN ERROR LARGER THAN 0.0100 ARE CONSIDERED


RESULTANT STRUCTURE: (5,8,20) (5,11,14) (7,8,20) (7,14,18) (8,10,20)
(10,14,18) (2,8,9,14) (4,6,7,8,19) (4,6,7,12,19) (4,9,11,12,13)
(2,9,11,12,13,14) (4,6,7,9,12,13)





BINARY RELATIONS WITH AN ERROR LARGER THAN 0.0150 ARE CONSIDERED


RESULTANT STRUCTURE: (4,10) (10,13) (10,18) (14,18) (2,11,13) (2,12,13)
(4,6,12) (4,6,19) (5,11,14) (8,10,20) (9,12,13) (6,7,8,19) (6,8,9,12)
(7,8,12,14) (7,12,13,14)
```

Three different largest reconstruction error values were applied: $e_{max} = 0.1$, $e_{max} = 0.01$, and $e_{max} = 0.015$. Notice that the bulk of the computing time is needed to compute the relative strengths of the binary relations, not to find the final composite structure, thus evaluating the effect of different e_{max} values on the resulting composite structure is cheap.

Let us analyse the results of this experiment. When the largest reconstruction error of a single binary relation between variables to be omitted is set to $e_{max} = 0.1$, the algorithm does not find any binary relation to consider, so all of them are omitted, and a totally unconnected structure is obtained.

In contrast with $e_{max} = 0.01$, 55 out of the 105 possible binary relations are considered. The resulting subsystems, labelled $S1_{FRA0.01}$ to $S12_{FRA0.01}$, are listed in Table 4-I. Unfortunately, the result is difficult to interpret. Three different structures contain the system output, X_{20}. Which of those should be used to compute the output? Do the other two occurrences introduce feedback loops into the resulting topological structure? How are algebraic loops among variables avoided?

$S1_{FRA0.01}$	X_5	X_8	X_{20}			
$S2_{FRA0.01}$	X_5	X_{11}	X_{14}			
$S3_{FRA0.01}$	X_7	X_8	X_{20}			
$S4_{FRA0.01}$	X_7	X_{14}	X_{18}			
$S5_{FRA0.01}$	X_8	X_{10}	X_{20}			
$S6_{FRA0.01}$	X_{10}	X_{14}	X_{18}			
$S7_{FRA0.01}$	X_2	X_8	X_9	X_{14}		
$S8_{FRA0.01}$	X_4	X_6	X_7	X_8	X_{19}	
$S9_{FRA0.01}$	X_4	X_6	X_7	X_{12}	X_{19}	
$S10_{FRA0.01}$	X_4	X_9	X_{11}	X_{12}	X_{13}	
$S11_{FRA0.01}$	X_2	X_9	X_{11}	X_{12}	X_{13}	X_{14}
$S12_{FRA0.01}$	X_4	X_6	X_7	X_9	X_{12}	X_{13}

Table 4-I Incinerator subsystem decomposition obtained with FRA using $e_{max} = 0.01$.

To avoid this problem, the largest reconstruction error was increased to $e_{max} = 0.015$. In this case, 36 of the 105 binary relations are retained, and only one subsystem containing the output variable is found. The resulting substructures are listed in Table 4-II.

$S1_{FRA0.015}$	X_4	X_{10}		
$S2_{FRA0.015}$	X_{10}	X_{13}		
$S3_{FRA0.015}$	X_{10}	X_{18}		
$S4_{FRA0.015}$	X_{14}	X_{18}		
$S5_{FRA0.015}$	X_2	X_{11}	X_{13}	
$S6_{FRA0.015}$	X_2	X_{12}	X_{13}	
$S7_{FRA0.015}$	X_4	X_6	X_{12}	
$S8_{FRA0.015}$	X_4	X_6	X_{19}	
$S9_{FRA0.015}$	X_5	X_{11}	X_{14}	
$S10_{FRA0.015}$	X_8	X_{10}	X_{20}	
$S11_{FRA0.015}$	X_9	X_{12}	X_{13}	
$S12_{FRA0.015}$	X_6	X_7	X_8	X_{19}
$S13_{FRA0.015}$	X_6	X_8	X_9	X_{12}
$S14_{FRA0.015}$	X_7	X_8	X_{12}	X_{14}
$S15_{FRA0.015}$	X_7	X_{12}	X_{13}	X_{14}

Table 4-II Incinerator subsystem decomposition obtained with FRA using $e_{max} = 0.015$.

The output is computed using variables X_8 and X_{10}. How are those variables evaluated? Unfortunately, both X_8 and X_{10} appear in 3 of the remaining substructures. Which of those should be used to compute them? What about feedback structures? Quite evidently, the previously mentioned causality problem has not been solved, only delayed.

4.2.7 Understanding the differ ent types of structures in FRA

All throughout the chapter, we have talked about how the internal structure of a system can be found, up to now only using FRA-based algorithms. Three different structure types were introduced:

- The *topological structure*, represented by a block diagram showing subsystems with inputs and outputs as well as interconnections between them. An example of this kind of structure is shown in Figure 4-3.

- The *composite structure*, obtained from the topological structure by enumerating the variables of each subsystem, ignoring whether they are input or output, and ignoring the topology of connections between the subsystems. For example, the composite structure *c_st3* is derived from the topological structure represented in Figure 4-3.

- The *binary structure*, obtained from the composite structure by expanding the variables of each subsystem into a set of binary relations and then merging all these sets into one in order to eliminate the redundant binary relations. For instance, the binary structure *b_st2* can be derived from the composite structure *c_st3*.

It is already known that the conversion from the topological structure via the composite structure to the binary structure is unique and straightforward. Unfortunately, the reverse is not true. The conversion from the binary to the composite structure is not unique, and we need additional rules to decide, which of the possible composite structures is to be selected. For instance, the binary structure *b_st2* previously used in this chapter may correspond to either the *c_st2* or the *c_st3* composite structures.

The conversion from the composite structure to the topological structure is not unique either. There may exist 0, 1, or multiple topological structures representing the same composite structure.

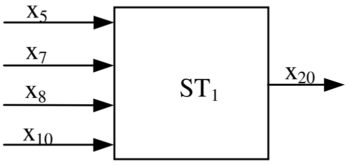
Whereas the binary structure is most useful for searching through a structure space and for comparing different structures with each other, what we ultimately need for a practical realisation is the topological structure. The single-step refinement algorithm of FRA will give us the relative strengths of all binary relations. It can be used to derive different binary structures (depending on the largest allowed reconstruction error of a binary relation to be omitted), which then in turn can be converted to a composite structure. Unfortunately, there is no known way to obtain a topological structure from that.

Since we need a topological structure, a different FRA-based algorithm is now proposed that can be used to derive the topological structure directly. It is based on the observation that the complete connectedness of a substructure only remains important as long as no causality is attached to it. Once it has been decided, which variable needs to be computed from the substructure, it is only important that the binary relations between the inputs and that output are strong. The binary relations among the different inputs are no longer of any major concern. In fact, it might be preferable that they are weak so that the model does not operate on unnecessary redundant information. The algorithm works as follows.

- We look at the relative strengths of all binary relations with the output variable. Those relations with strength larger than *x* are identified, where *x* may assume a value such as 0.01. For the example at hand, i.e., the incinerator system, we find the following significantly large binary relations with the output (the binary relations are listed using the indices of the variables only):

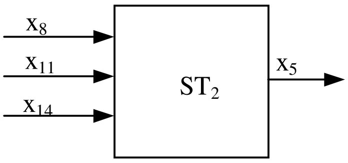
5, 20	err = 0.011106
7, 20	err = 0.014242
8, 20	err = 0.018102
10, 20	err = 0.021783

- Since individual submodels that are too complex are not desired, the 4 most important inputs are selected, if there are at least 4 inputs with strengths greater than *x*, otherwise, all inputs with strengths greater than *x* are selected. For the case of the incinerator system, exactly 4 binary relations with the output variable are found that comply with these requirements. Consequently, the first substructure is found to be:

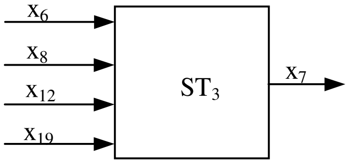


- The algorithm is repeated for every one of these inputs, excluding relations with those variables that had previously been used as outputs of subsystems. In the current situation, this only applies to X_{20}. The following substructures are found:

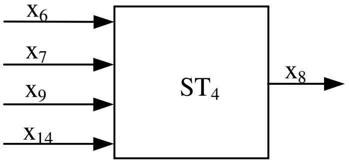
5, 8	err = 0.013272
5, 11	err = 0.021411
5, 14	err = 0.015141



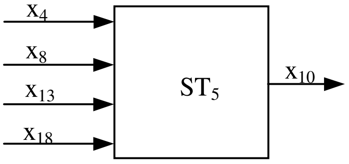
4, 7	err = 0.012103
6, 7	**err = 0.024196**
7, 8	**err = 0.021566**
7, 9	err = 0.012323
7, 12	**err = 0.022214**
7, 13	err = 0.016339
7, 14	err = 0.020031
7, 18	err = 0.013191
7, 19	**err = 0.021365**



2, 8	err = 0.011050
4, 8	err = 0.013151
5, 8	err = 0.013272
6, 8	**err = 0.022142**
7, 8	**err = 0.021566**
8, 9	**err = 0.023871**
8, 10	err = 0.018399
8, 14	**err = 0.023143**
8, 19	err = 0.018324

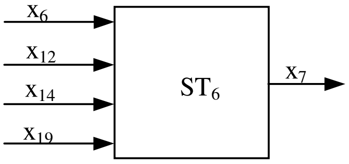


4, 10	**err = 0.016134**
8, 10	**err = 0.018399**
10, 12	err = 0.015678
10, 13	**err = 0.018031**
10, 14	err = 0.013597
10, 18	**err = 0.023159**



- Now it is necessary to eliminate algebraic loops. There is one algebraic loop in this example, because $X_8=f(X_7)$ and $X_7=f(X_8)$. The largest not selected relation among those of the two substructures is identified under the constraint to be different from the one that provokes the algebraic loop. For the studied example we find the relation (7, 14). We substitute the algebraic loop relation by the new one, thus, the ST_3 substructure it is replaced by:

4, 7	err = 0.012103
7, 8	err = 0.021566
7, 9	err = 0.012323
7, 13	err = 0.016339
7, 14	**err = 0.020031**
7, 18	err = 0.013191

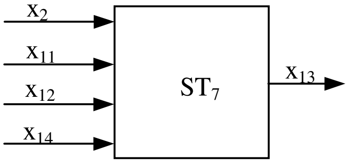


- In this step, the variables that have not been used in any of the found substructures need to be studied. For the given example all variables except X_2 have been used. Thus, the strengths of all variables that have not yet been used as outputs can be checked with that variable:

2, 9	err = 0.014203
2, 11	err = 0.021082
2, 12	err = 0.016028
2, 13	**err = 0.027111**
2, 14	err = 0.010561

- The strongest relation is with variable X_{13}, thus we make a model of that variable:

2, 13	**err = 0.027111**
4, 13	err = 0.013029
6, 13	err = 0.014601
9, 13	err = 0.018401
11, 13	**err = 0.019011**
12, 13	**err = 0.024819**
13, 14	**err = 0.023164**



By now a complete topological structure has been extracted from the single-refinement information of FRA. Before we had 15 variables, thus in order to model the output (X_{20}) from the full set of inputs, we would have needed 14 sensors. By now, we have 6 submodels. Each of them computes one variable, thus we only need 9 sensors now. We could easily continue adding more substructures, until the number of sensors has been reduced to the desired level. Of course, these additional models will be poorer and poorer in quality, because there may be strong relations with previously used outputs that cannot be used again. Thus, there is a compromise to be made.

The resulting structure of applying this algorithm to the incinerator system is depicted in Figure 4-5.

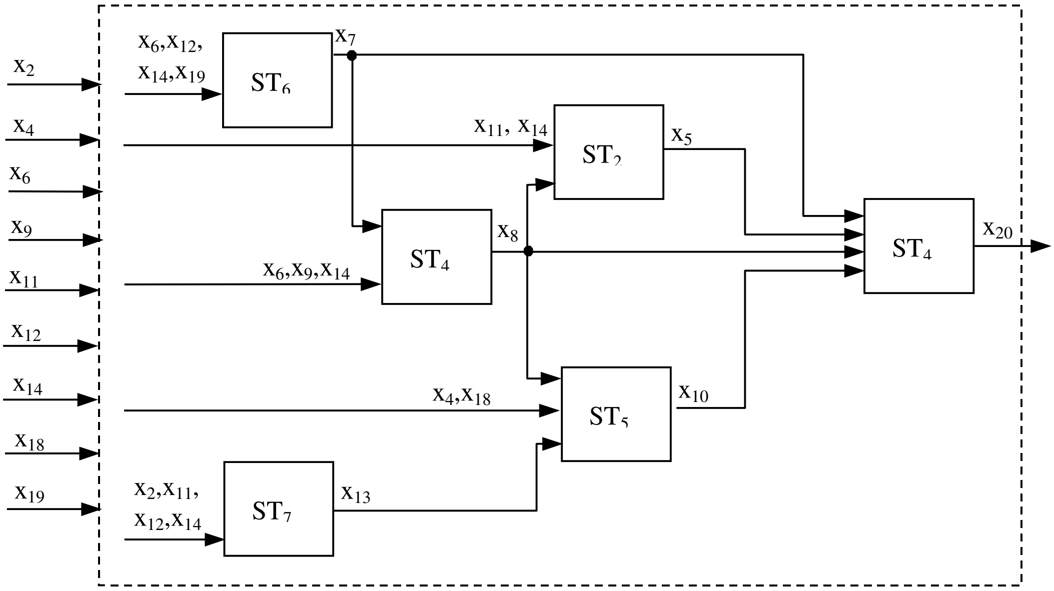


Figure 4-5 Topological structure obtained by means of a FRA based algorithm.

The proposed methodology offers a comprehensive approach to determining the substructure of a model, assuming that all variables need to be used (which makes sense, because these are the variables left over after the elimination step of Chapter 3). Once a structure has been found that accounts for all the variables, additional substructures can be added to reduce the number of true system inputs, i.e., the number of sensors that need to be used in the system. Yet, additional substructures should be added sparingly and after serious contemplation only, as these substructures will inevitably exhibit poorer tracking capabilities.

The single-step refinement algorithm found 55 important binary relations. The structure resulting from the proposed algorithm contains 46 binary relations, 37 of which are important, and 9 are unimportant.

4.3 Using FIR to find the structure of a system

Up to this point, FRA has been proposed as a technique for identifying the internal structure of a system. It was shown that its single-step refinement algorithm indeed could provide the information needed to determine a meaningful internal structure of a system. Unfortunately, the algorithm, at least in its current Matlab implementation, is far too slow to be used on a truly large-scale system.

In this section, an alternative algorithm based on FIR is proposed to derive the internal structure of the system under study. FIR is more efficient than FRA, and therefore, it is hoped that similarly good results can be obtained considerably faster using this methodology. In order for the reader to be able to compare the different algorithms with each other, also this new algorithm is applied to the garbage incinerator system.

The algorithm follows similar paths of reasoning as the previously introduced FRA-based algorithm. As before, the algorithm is applied to the subset of variables remaining after the variable elimination step of Chapter 3. As before, only static relations are considered for now.

How can the FIR methodology, a level-3 tool on the GSPS epistemological ladder, be used to determine the internal structure of a system, a problem that clearly calls for a level-4 tool? The answer to this question is straightforward. FIR, although predominantly located at level 3, still provides some level-4 information that can be exploited by an appropriate algorithm. The levels of Klir's epistemological ladder are idealized concepts. The practical GSPS tools do not fully coincide with this classification.

In the sequel, the FIR-based structure identification algorithm is presented.

- With the variables remaining after the variable elimination step of Chapter 3, a flat (static) FIR model of the output variable is constructed. Due to the complexity issue, FIR will select only a subset of the possible inputs, probably not more than 4 or 5 of them. For the example at hand, the incinerator system, there remain fourteen variables after the variable selection step performed in Section 3.4.1. Those variables are X_2, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}, X_{13}, X_{14}, X_{18}, and X_{19}, to which the output variable X_{20} should be added. When an optimal mask search is performed for this 15 variables system, FIR chooses the following model:

  $$X_{20} = \mathrm{f}_1\,(X_4, X_5, X_6, X_{12}, X_{19}) \qquad \mathrm{Q} = 0.2089$$

  Although we already know that X_{20} depends on all of the remaining 14 inputs, FIR only selects a subset of those variables, namely the ones that are most useful in interpreting the observations made about the output. The reader may notice that the set of variables selected by FIR is dramatically different from the one chosen by FRA. The two models have only one variable, X_5, in common. The variable with the strongest binary relation to the output, X_{10}, was not selected at all by FIR.

- Now it is necessary to determine the relative importance of each of the inputs used by the model that was previously found. In the given example, there are 5 inputs. The relative importance of the five inputs could, of course, be determined using FRA, but the same can also be accomplished using FIR in the following manner: One at a time, each of the five inputs is severed, and the quality of the resulting FIR model without the severed input is computed [Nebot and Mugica, 1996]. Thus, in the case at hand, the fuzzy quality of the five following models is computed:

$$
\begin{array}{lcl}
X_{20} = \mathrm{f}_2\,(X_4, X_5, X_6, X_{19}) & \rightarrow & \mathrm{Q} = 0.1752 \\
X_{20} = \mathrm{f}_3\,(X_4, X_5, X_{12}, X_{19}) & \rightarrow & \mathrm{Q} = 0.1735 \\
X_{20} = \mathrm{f}_4\,(X_5, X_6, X_{12}, X_{19}) & \rightarrow & \mathrm{Q} = 0.1521 \\
X_{20} = \mathrm{f}_5\,(X_4, X_6, X_{12}, X_{19}) & \rightarrow & \mathrm{Q} = 0.1339 \\
X_{20} = \mathrm{f}_6\,(X_4, X_5, X_6, X_{12}) & \rightarrow & \mathrm{Q} = 0.1165
\end{array}
$$

  Now, we have a measure of the importance of each of the inputs that model the output variable of the system. Since f_6 reduces the quality most, X_{19} was the most important input. Similarly X_{12} is the least important one.

- In this next step of the algorithm, a FIR model is generated for each of the inputs, starting with the least important one, excluding those variables as possible inputs that had previously been used as outputs. In the current situation, only X_{20} needs to be excluded. , The least important input for the incinerator system is X_{12}, and the FIR model found for this variable is:

$$X_{12} = \mathrm{f}_7\,(X_4, X_6, X_7, X_9, X_{13}) \qquad \text{with Q} = 0.4690$$

  Thus, we just found the next substructure:

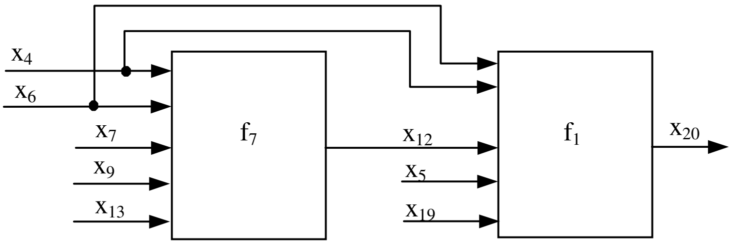


Figure 4-6 Second substructure found for the incinerator system using FIR.

- Now, we proceed with the next least important input of f_1, which happens to be X_6. Thus we make a FIR model, excluding all of those variables that were already outputs, namely X_{12} and obviously X_{20}. In this way, we find the next substructure. The procedure continues in the same fashion until all physical inputs of the system, in this case 14, appear among the inputs of the model. This gives us a complete structure. For the incinerator system, the following models are found for each of the inputs of the f_1 model:

$X_6 = \mathrm{f}_8\,(X_4, X_7, X_8, X_{13})$ with Q = 0.4922
$X_4 = \mathrm{f}_9\,(X_2, X_9, X_{13}, X_{14}, X_{19})$ with Q = 0.2765
$X_5 = \mathrm{f}_{10}\,(X_8, X_{10}, X_{11}, X_{14}, X_{18})$ with Q = 0.4696
$X_{19} = \mathrm{f}_{11}\,(X_7, X_8, X_{11}, X_{14})$ with Q = 0.1375

The given algorithm is based on the idea that it may not be economical to provide the system with sensors for all of the inputs (in this case, fourteen input variables). Instead, by providing an internal structure, sensors need only be provided for the true inputs, whereas the internal variables (such as X_{12} in Figure 4-6) can be estimated using FIR models. The least important inputs were modelled first, in order to reserve sensors for the more important ones.

For the incinerator system, when applying the algorithm explained above, the following structure is found:

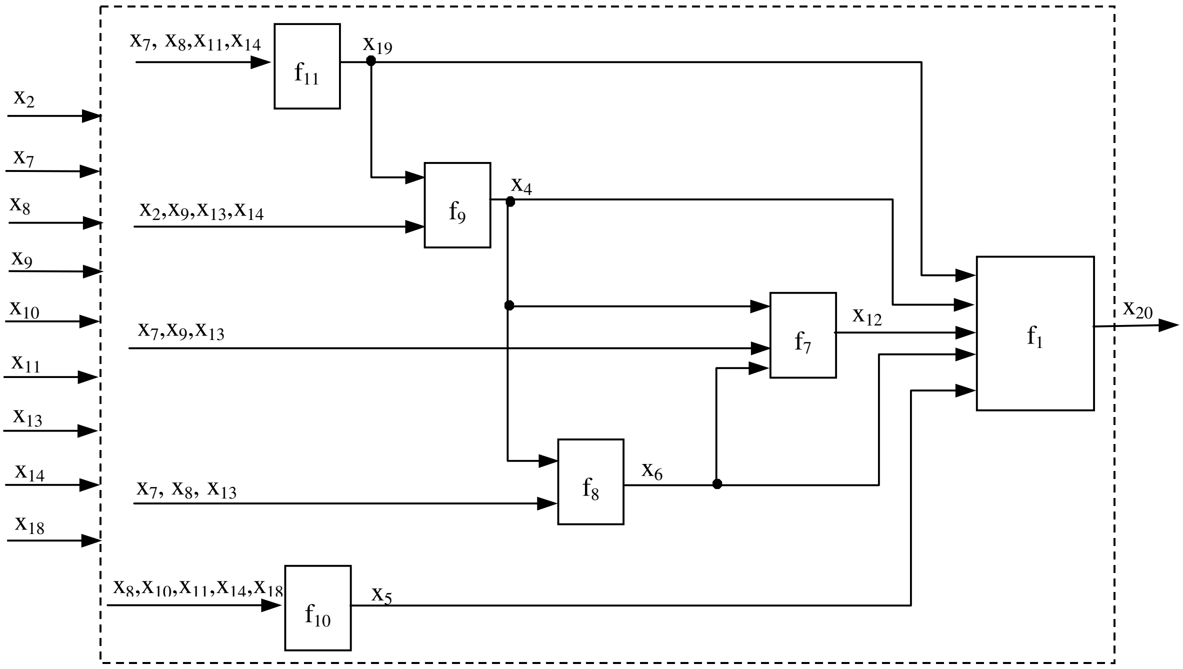


Figure 4-7 Possible structure for the incinerator system found with FIR.

The proposed methodology offers an alternative approach to determining the substructure of a model, assuming again at all variables need to be used. Just as in the previous case, once a structure has been found that accounts for all the variables, additional substructures can be added to reduce the number of true system inputs, i.e., the number of sensors that need to be used in the system. Yet, additional substructures should be added sparingly and after serious contemplation only, as these substructures will inevitably exhibit poorer tracking capabilities.

The structure resulting from the proposed FIR-based algorithm contains 62 binary relations, 39 of which are important, whereas 23 are unimportant.

4.4 Correlation matrix singular value decomposition

The aspects tackled in this section together with the work presented in Section 3.4.1 and the work that will be presented in the next section form part of a full statistics-based methodology for decomposing a complex system into subsystems. The work has been partially published in [Mirats and Verde, 2000]. In Section 3.4.1, a method based on simple linear correlation coefficients was presented with the purpose of performing a variable selection. With this variable selection, those variables that are redundant, in a linear sense, so containing the same information about the studied system, are dropped out of the posterior modelling process.

Now, with the variables remaining after the variable selection step, we want to form subsets (subsystems) of variables that are linearly correlated among each other. The process presented here to achieve this goal is based on the first steps of a principal component analysis.

First, the correlation matrix of the remaining variables together with the output has to be computed. In fact, it is not necessary to recompute this matrix, because in the previous variable selection step, the full correlation matrix for the system under consideration was already calculated. Since the linear correlation between two system variables is invariant to the presence or absence of other variables in the correlation matrix, it is only necessary to eliminate those rows and columns of the full correlation matrix that pertain to the removed variables.

Once the reduced correlation matrix has been found, a singular value decomposition of this matrix is performed, so its eigenvalues and eigenvectors are obtained. For the particular case of normalised data, the eigenvectors coincide with the principal components of the system. Then, those eigenvectors, known to be orthogonal, are projected onto the principal axes. Theoretically the projections onto all of the subspaces spanned by each pair of eigenvectors have to be examined, but, in practice, it is enough to take into account those projections only that primarily account for the system variance. Then, for each projection, the space spanned by the axes is divided into regions of 30º, and the variables with larger projection in the obtained regions are joined in the same subset[7]. The reason to use narrow angle regions is derived from the fact than the cosines of the angle between two vector variables on the factorial plane is proportional to the correlation between those two variables. In this way, subgroups of linearly related variables are obtained.

Let us show how the described process works using the example of the garbage incinerator system described in the Appendix I.3. The results from Section 3.4.1 are recalled here. For this example, variables X_1, X_3, X_{15}, X_{16}, and X_{17}, had already been eliminated because they contained redundant information. The remaining correlation matrix is:

[7] The regions are 30º wide beginning at 0º and 15º, that is regions [0,30], [30, 60]…[0, -30] [0, -60]… [15, 45], [45, 75]… and [-15, 15], [-15, -45]… have been considered for the analysis.

	X_{2}	X_{4}	X_{5}	X_{6}	X_{7}	X_{8}	X_{9}	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{18}	X_{19}	X_{20}
X_{2}	1,00														
X_{4}	0,25	1,00													
X_{5}	0,02	0,01	1,00												
X_{6}	0,51	0,39	0,13	1,00											
X_{7}	0,55	0,34	0,22	0,71	1,00										
X_{8}	0,07	0,26	0,28	0,46	0,59	1,00									
X_{9}	0,27	0,06	0,00	0,18	0,29	0,07	1,00								
X_{10}	0,16	0,09	0,08	0,16	0,32	0,26	0,20	1,00							
X_{11}	0,54	0,31	0,07	0,46	0,54	0,31	0,17	0,49	1,00						
X_{12}	0,44	0,53	0,02	0,40	0,66	0,51	0,53	0,28	0,51	1,00					
X_{13}	0,62	0,37	0,14	0,56	0,72	0,53	0,32	0,37	0,70	0,69	1,00				
X_{14}	0,27	0,31	0,19	0,47	0,60	0,59	0,11	0,52	0,27	0,43	0,53	1,00			
X_{18}	0,00	0,10	0,07	0,06	0,03	0,04	0,09	0,02	0,01	0,02	0,02	0,03	1,00		
X_{19}	0,00	0,03	0,25	0,17	0,20	0,35	0,02	0,09	0,11	0,06	0,23	0,20	0,04	1,00	
X_{20}	0,00	0,31	0,35	0,47	0,20	0,42	0,12	0,03	0,01	0,34	0,11	0,05	0,22	0,34	1,00

Table 4-III Remaining correlation matrix for the garbage incinerator system after performing a variable selection.

Now, a singular value decomposition of this matrix is performed. In our case, this was done using a built-in Matlab function, but there exist many other commercial software packages that allow performing this operation, so the algorithm can easily be implemented on a variety of different hardware and software platforms. Once the eigenvectors are obtained, the projection onto the principal axes is computed. For instance, Figure 4-8 shows the projection onto the first and second principal axes.

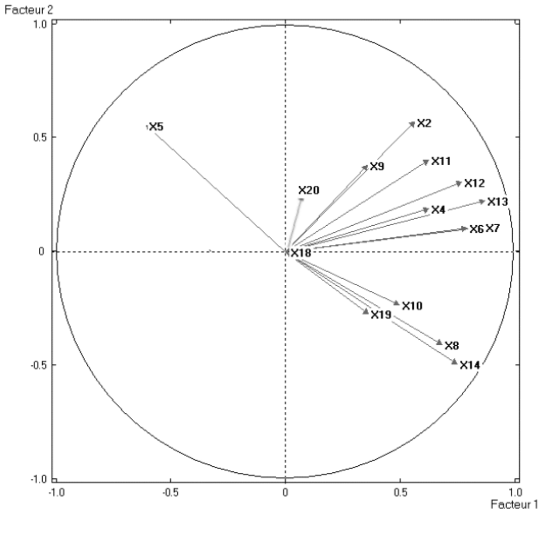


Figure 4-8 Incinerator system: projections onto the first and second principal axes.

Then as previously explained, each quadrant of the figure is divided into different 30° regions, and variables with large projection[8] in each of those regions are grouped together. The subsystems that are obtained from the projection shown in Figure 4-8 are presented in Table 4-IV.

X_5	X_8	X_{14}		
X_8	X_{14}			
X_4	X_6	X_7	X_{12}	X_{13}
X_6	X_7	X_{13}		
X_2	X_{11}	X_{12}		

Table 4-IV Subsystems resulting from the projection onto the first and second principal axes.

The reader may notice that X_5 got lumped together with X_8 and X_{14} in spite of the fact that X_5 is almost 180° off from the other two variables. The reason is that X_5 is strongly, albeit negatively, correlated with the other two variables.

Three additional projections have been taken into account in this example to obtain subsets of variables: the third plane versus the first, the second versus the third, and the fourth versus the first. Table 4-V shows all the subsets determined by those projections. The S_1 to S_6 subsets correspond to the projection of axes 1-3, S_7 to S_9 to that of axes 1-4, and S_{10} to S_{13} to that of axes 2-3. If more projections are investigated, it is found that they do not offer any additional information about subsystems.

S_1	X_{10}	X_{14}			
S_2	X_7	X_{12}	X_{13}		
S_3	X_4	X_6	X_8		
S_4	X_8	X_{14}			
S_5	X_4	X_6	X_7	X_{12}	X_{13}
S_6	X_2	X_{11}	X_{12}		
S_7	X_7	X_8	X_{13}	X_{14}	
S_8	X_4	X_{12}			
S_9	X_2	X_{11}	X_{13}		
S_{10}	X_4	X_6	X_8		
S_{11}	X_2	X_{11}	X_{13}		
S_{12}	X_5	X_{11}	X_{14}		
S_{13}	X_7	X_8	X_{13}	X_{14}	

Table 4-V Obtained static subsystems for the garbage incinerator system from different principal axes projections.

Some of the projections onto the principal axes provide redundant information about the subsystems to be formed. In the case of the considered example, as can be seen from Table 4-V, there exists a redundancy between some of the subsystems found. For example, subsystem S_2 is included in S_5, and therefore does not offer new information about the process being studied. Thus, subsystem S_2 can be dropped out for a posterior modelling analysis. Finally, the obtained subsystem decomposition for this system, after eliminating

[8] Here, a large projection means a normalised projection greater than 0.6. This limit has been derived heuristically.

the redundant ones, is shown in Table 4-VI. The remaining subsystems have been relabelled from S_1 to S_8.

S_1	X_5	X_8	X_{14}		
S_2	X_4	X_6	X_7	X_{12}	X_{13}
S_3	X_2	X_{11}	X_{12}		
S_4	X_4	X_6	X_8		
S_5	X_{10}	X_{14}			
S_6	X_2	X_{11}	X_{13}		
S_7	X_5	X_{11}	X_{14}		
S_8	X_7	X_8	X_{13}	X_{14}	

Table 4-VI Obtained static subsystem decomposition for the garbage incinerator system only taking into account linear relations.

So far, only linear relations among variables were considered. Yet, a purely linear analysis tool cannot be competitive when dealing with practical engineering systems, as non-linear relations among variables invariably play an important role in most of them.

In the next section, the method will be extended to also consider non-linear relations among variables.

4.5 Finding non-linear relations between variables and subsystems

In the previous section, a method for grouping variables, based on a singular value decomposition of the data correlation matrix, has been presented that only looked at linear relations between variables. The method was applied to the garbage incinerator system, and the subsystems that were formed in this way have been reported in Table 4-VI.

Unfortunately, statistical techniques are, in general, linear techniques. How can a statistical method be applied to a non-linear problem? If the problem is known, the answer to this question is simple: We first apply a non-linear transformation to the non-linear problem to make it linear, then apply the statistical method to the previously linearised problem. What if the problem is unknown, like in our case? The trick then is to find a non-linear transformation that has a tendency of making the unknown problem more linear in the process. This is what shall be attempted here.

We shall analyse each subsystem separately, and check whether any of the variables not currently included in that subsystem exhibits a non-linear correlation with the cluster of variables forming the subsystem. If this is the case, that variable is then added to the subsystem.

Not all non-linear relations will necessarily be found in this way. It could happen, for example, that two variables exhibit a non-linear correlation between themselves, yet neither of them is linearly related to any of the subsystems that were previously found. In that case, a new subsystem ought to be declared.

Let us assume the overall system consists of n variables, among which m subsystems have been identified. Let us denote the i^{th} subsystem as $S_i = \{X(i_1) \dots X(i_{ki})\},\ \ i \in 1,m$. It contains ki variables, namely the variables $X(i_1)$ up to $X(i_{ki})$. Parentheses and double indices are used here to avoid multiple subscripts. There are $n\text{-}ki$ variables not included in the i^{th} subsystem. Let us call the set of those variables $T_i = \{X(i_{ki+1}) \dots X(i_n)\},\ \ i \in 1,m.$ Thus, the set of all variables V is the concatenation of the sets S_i and T_i, $V = \{S_i, T_i\}$. Let $Z_j = X(i_{ki+j}) \subset T_i$ be one of the variables that are not included in the subsystem.

Then, in order to study a possible non-linear relation between a given subsystem S_i, or any subset of its variables, and a given variable Z_j not included in S_i, the normal linear correlation between (Z_j^*, ξ_i) is calculated, where $Z_j^* = spline(Z_j)$ is a non-linear transformation of variable Z_j, using a spline curve, and $\xi_i = linear(X(i_1) \dots X(i_{ki}))$ is a linear combination of the ki variables from the i^{th} subsystem.

Splines are usually curves required to be continuous and smooth. The splines are normally defined as piecewise polynomials of degree q, the function values and first $q\text{-}1$ derivatives of which coincide at the points where they join. The abscissa values of the points where neighbouring splines meet are called knots in the mathematical literature [Smith, 1979]. The term spline is sometimes also used to denote polynomials (splines without knots) and piecewise polynomials with more than one discontinuous derivative. Splines without knots are generally smoother than splines with knots, which in turn are generally smoother than splines with multiple discontinuous derivatives. Splines with few knots are generally smoother than splines with many knots; however, increasing the number of knots usually improves the fit of the spline function to the data. For more information about splines and their mathematical properties, the reader is referred to [Smith 1979] and [de Boor, 1978], which offer excellent insights into the properties of spline functions.

When a variable is fit by a spline curve, two possibilities can be considered: the first is to use a unique spline curve that fits the entire data vector, this has been the solution adopted here. The second, advocated by [Brieman and Friedman, 1985; Friedman, 1991], is to split the data into different regions, as when a regression tree is constructed [Siciliano and Mola, 1996; 1997], and find a spline that fits each of these regions. Constraints are imposed when computing the different splines in order to maintain continuity of the fit. In this way, the obtained fit is better than in the previous case, but the algorithm that decides into which regions the data should be divided, i.e., that constructs the regression tree, is very time consuming since all the possible splits are taken into account. For this reason[9], a unique spline curve was used in this research to non-linearly transform the data. In the case at hand, cubic splines ($q = 3$) with three knots have been used.

The spline transformation for variable Z_j is computed taking into account that the variable Z_j has to be analysed together with the variables of the subset S_i. This means that each time, we investigate the existence of a non-linear correlation between a variable Z_j and a given subset S_i, the computed spline transformation for variable Z_j may be different.

[9] Intuitively, it is supposed that the non-linear relation is invariant in the considered time interval.

How is this spline curve computed that takes into account the subsystem information? The trick is to consider each of the Z_j variables as a dependent (output) variable, and the $X(i_1)\ldots X(i_{ki})$ variables as independent (input) variables. Then a linear regression of the i^{th} subset variables is performed onto the non-linear transformation of the Z_j variable. An iterative algorithm that automatically adjusts the parameters of the non-linear spline transformation, Z_j^*, and those of the regression equation, ξ_i, has been used, so the best transformation [Verde, 1994] of variable Z_j is obtained. This algorithm is based on the alternating least squares of [Young *et al.*, 1976; Young, 1981], and it is internally provided in the SAS software statistical package [SAS, 1988].

Thence, the currently investigated variable, Z_j, is replaced with the newly found iteratively derived optimal non-linear transformation, Z_j^*, that fits best, in a least squares sense, the specified subsystem, S_i. In this way, a non-linear transformation of variable Z_j has been obtained that takes into account that it has to be analysed jointly with the set of variables S_i.

Previous to the transformation of variable Z_j, the linear correlation between Z_j and each of the variables of the S_i subset was weak. Now, the non-linear transformed variable, Z_j^*, may have increased its linear correlation with some of the variables of the S_i subset, or the computed linear combination, ξ_i.

So finally, to decide if variable Z_j exhibits a non-linear correlation with the considered subsystem S_i, the linear correlation between this non-linear transformed variable, Z_i^*, and each of the variables of the S_i subset as well as the linear combination ξ_i is computed. If any of those linear correlation coefficients is sufficiently high[10], the variable is considered to exhibit non-linear correlation with the corresponding subsystem, and is, thus, added to the subsystem.

Let us now return to the garbage incinerator example. In the previous section within this chapter, a subsystem decomposition of this system has already been obtained. Eight different subsystems, $S_1 \ldots S_8$, resulted, containing between two and five variables each. Some of the system variables, namely X_9, X_{18}, X_{19}, and X_{20} do not form part of any of these subsystems. At this point, possible non-linear correlations between the previously found subsets and those variables not included in them need to be investigated. To this end, the previously explained algorithm is applied to investigate if any of these variables can form part of any one of these eight subsystems.

With each of the eight subsystems, the best non-linear transformation, cubic spline based, is computed for each of the system variables not contained in the subsystem being investigated using the previously explained non-linear iteration algorithm. The correlation matrices between each of the spline-transformed variables and the subsystem variables are not reported here. Table 4-VII summarises all these computations, and shows where variables need to be added to the existing subsystems, because these spline-transformed variables have been found to exhibit non-linear correlation with the corresponding subsystem. In this table, variables that were already present in the previously found

[10] Here, the adopted criterion is the same that was used in Section 3.4.1 when looking for simple linear correlation between input variables.

subsystems have been labelled as '*lin*' whereas variables that are found to exhibit non-linear correlation with a given subsystem are labelled '*add*'.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
X_2^*	add	-	lin	-	-	lin	-	-
X_4^*	-	lin	-	lin	-	-	-	-
X_5^*	lin	-	-	-	-	-	lin	-
X_6^*	-	lin	-	lin	-	-	-	-
X_7^*	-	lin	-	add	add	-	-	lin
X_8^*	lin	-	-	lin	-	-	-	lin
X_9^*	-	add	-	-	-	add	-	add
X_{10}^*	-	-	-	-	lin	-	-	-
X_{11}^*	-	-	lin	-	add	lin	lin	-
X_{12}^*	-	lin	lin	-	add	-	-	-
X_{13}^*	-	lin	-	-	-	lin	-	lin
X_{14}^*	lin	-	-	-	lin	-	lin	lin
X_{18}^*	-	-	-	-	add	-	-	-
X_{19}^*	-	-	-	add	-	-	-	-
X_{20}^*	-	-	-	-	-	-	-	-

Table 4-VII Non-linear correlations between non-linear transformed variables of the incinerator and the previously found subsets.

The algorithm was executed multiple times, because once a variable has been added to a subset, it is possible that another variable now exhibits non-linear correlation with the already enlarged subset. The algorithm was terminated only when no more additional variables were added to any of the subsystems during an entire step. Finally, Table 4-VIII shows the subsystem decomposition of the garbage incinerator process.

S_1	X_2	X_5	X_8	X_{14}		
S_2	X_4	X_6	X_7	X_9	X_{12}	X_{13}
S_3	X_2	X_{11}	X_{12}			
S_4	X_4	X_6	X_7	X_8	X_{19}	
S_5	X_7	X_{10}	X_{11}	X_{12}	X_{14}	X_{18}
S_6	X_2	X_9	X_{11}	X_{13}		
S_7	X_5	X_{11}	X_{14}			
S_8	X_7	X_8	X_9	X_{13}	X_{14}	

Table 4-VIII Final subsystem decomposition for the garbage incinerator system including linear as well as non-linear static relations among variables.

The reader may notice that almost all of the variables now form part of some subsystem or other. The exception is the output variable, X_{20}, which still does not form part of any of the subsystems. Since there is only one variable left, there is no need to consider additional new groups of variables.

It is utterly annoying that, of all the variables in question, it is precisely the output that doesn’t form part of any system, i.e., that seems uncorrelated, both in a linear and in a non-linear sense, with the rest of the system, as this prevents us from deriving a topological structure for the incinerator system using the statistical approach. Of course, the same accident could have occurred in the FRA-based algorithm, although it did not in the example at hand.

Interpreting the results obtained as a composite structure, it is now possible to expand the structure into a binary structure, listing all the binary relations that exist among the variables of all eight of the subsystems. It turns out that this binary structure lists 50 binary relations, 44 of which are among the 55 important binary structures discovered by FRA, whereas only 6 of them are unimportant. It is evident that the proposed approach can be used as an alternative technique to the single-step refinement algorithm of FRA for discovering important binary relations among system variables.

How can the statistical technique be used to derive a topological structure, assuming the output variable forms part of the structure? One possible algorithm would be as follows.

- Starting with the output, all binary relations with the output that were considered by the statistical technique are offered to FIR as potential inputs, and FIR is asked to compute the optimal (static) mask for this subsystem. FIR may choose to make use of all of these binary relations as inputs, or it may select a subset of them.

- Each of these inputs is, in turn, treated as an internal variable, and all considered binary relations with that variable, except for the output, are offered to FIR as potential inputs for the next substructure. FIR chooses the set of input variables to be used.

- The algorithm continues until all variables have been referenced in the model either as output, as internal variables, or as external inputs.

Unfortunately, the operation of this algorithm cannot be demonstrated at hand of the incinerator system, because the statistical technique did not consider any of the binary relations with the output, either linear or non-linear, to be significant, at least not in a static sense. It shall be shown later that the technique indeed discovers significant correlations with the output, when time-dependent relations are allowed, i.e., when a dynamic model is being constructed.

As for the case of the previous section, the reader is informed that some aspects of the non-linear extensions to the statistical approach to substructure identification have already been outlined in [Mirats and Verde, 2000].

4.6 Conclusions

This chapter dealt with the difficult problem of structure identification in large-scale systems. Three different algorithms were proposed, each of which has the potential of discovering the internal topological structure of a system:

- an algorithm based on Fuzzy Reconstruction Analysis (FRA),
- an algorithm based on Fuzzy Inductive Reasoning (FIR), and
- an algorithm based on linear and non-linear correlation techniques.

Let us discuss briefly the pros and cons of these three algorithms?

The FRA-based algorithm is clearly the best from the point of view that FRA, when executed in an exhaustive fashion, looks at every possible projection of every variable onto every subspace formed by any subset of the other variables. Unfortunately, the number of variations to be considered grows so rapidly with the number of variables in the system that this approach is at best of academic interest. Only one highly suboptimal FRA-based algorithm has any chance of terminating within a reasonable time span: the single-step refinement algorithm. Unfortunately, even this algorithm becomes highly inefficient for an even modestly large number of variables. Hence FRA-based techniques can only be applied to small- to medium-scale systems.

FRA-based algorithms operate on the binary structure of the system. Whereas the composite structure can be derived from the binary structure, it is not possible to derive the topological structure (which is our aim) from the composite structure. A heuristic algorithm was proposed that derives a topological structure directly from the relative strengths of the binary relations found by the single-step refinement algorithm of FRA. The algorithm has two major drawbacks:

1. The number of inputs of each subsystem was limited to four. Why four? Why not three of five? Shouldn't the number of possible inputs depend on the quantity and quality of available observational data? FIR clearly does a better job at selecting inputs in a dynamical and rational fashion.

2. The algorithm only considers the strengths of binary relations between the individual inputs and the output of each subsystem. The relative strengths of the binary relations among the inputs are not considered. Hence FRA chooses those variables with the largest strengths of binary relations to the output, but maybe, two of those input variables have strong binary relations between them so that another input variable might offer more additional information in spite of exhibiting a smaller strength of its binary relation with the output. Again, FIR does a considerably better job than FRA in deciding the relative importance of different potential inputs. This is precisely what FIR was designed to do.

The FIR-based algorithm makes use of all the variables to select an optimal mask for the output variable. It then works its way back towards the inputs, treating each of the inputs of this submodel in turn as an internal variable, proposing a FIR-model for it. The algorithm continues until all variables are accounted for. The FIR-based approach has the advantage that it makes use of a methodology that was designed as a tool for the generation of simulation models with optimal prediction power. Hence the resulting FIR-model decomposition should offer excellent prediction capabilities. The FIR-based algorithm also has its drawbacks:

1. The algorithm, as proposed, considers the weaker inputs first as internal variables. This makes sense, because it allows sensors to be reserved for the stronger variables. Yet in the given example, all inputs of the last stage turned out to be made into internal variables. Wouldn't it then have made more sense to model the stronger inputs first? Doesn't FIR try to make a compromise in each model between completeness of available information (many inputs) and complexity of the model (few inputs), in order to optimise the usage of the available observational data? Isn't there a "conflict of interest" between choosing stronger *vs.* weaker variables first?

2. The main goal of this dissertation is to find techniques that reduce the workload for FIR by reducing the number of potential input variables to be considered. Yet already in the first step of this algorithm, a complete FIR problem is being solved, i.e., the main purpose of the investigation is defeated. Clearly, FIR is not suited as a tool for dealing with large-scale systems directly.

The statistical approach proposed in this chapter is clumsy and awkward. Its functioning is much less transparent than in the case of either FRA- or FIR-based algorithms. Statistical techniques are inherently linear techniques, and somewhat dubious tricks had to be invented in order to capture the non-linear information contained in the observational data in a meaningful way. Yet, the final results obtained are amazingly good. The statistical approach is capable of identifying almost the complete set of important binary relations between pairs of variables. Yet, and contrary to FRA, this technique executes rather rapidly and is perfectly capable of dealing with large-scale systems. Thus, in spite of its awkwardness, this technique may be the only technique available when dealing with a truly large-scale system, which we haven't done so far in this dissertation.

It may be interesting to compare the three approaches from the point of view of their coverage of the important binary relations. To this end, the binary relations of the incinerator system were reordered in terms of decreasing relative strengths, and next to each binary relation, it was marked whether or not it was captured by any or all of the three proposed algorithms.

Importance of binary relation	Binary relation	FRA-based method	FIR-based method	Statistical method
0.031104	12 , 14	√	□	□
0.031102	6 , 19	√	√	√
0.028183	4 , 6	□	√	√
0.027111	2 , 13	√	√	√
0.025047	4 , 12	□	√	√
0.024819	12 , 13	√	√	√
0.024196	6 , 7	√	√	√

0.023871	8 , 9	√	□	√
0.023185	6 , 12	√	√	√
0.023164	13 , 14	√	√	√
0.023159	10 , 18	√	√	√
0.023143	8 , 14	√	√	√
0.022214	7 , 12	√	√	√
0.022142	6 , 8	√	√	√
0.021783	10 , 20	√	□	□
0.021566	7 , 8	√	√	√
0.021411	5 , 11	√	√	√
0.021365	7 , 19	√	√	□
0.021305	14 , 18	□	√	√
0.021164	11 , 14	√	√	√
0.021082	2 , 11	√	□	√
0.020031	7 , 14	√	√	√
0.019011	11 , 13	√	□	√
0.018401	9 , 13	□	√	√
0.018399	8 , 10	√	√	□
0.018324	8 , 19	□	√	√
0.018102	8 , 20	√	□	□
0.018031	10 , 13	√	□	□
0.016835	4 , 19	□	√	√
0.016339	7 , 13	□	√	√
0.016134	4 , 10	√	□	□
0.016028	2 , 12	√	□	√
0.015811	9 , 12	□	√	√
0.015678	10 , 12	□	□	√
0.015141	5 , 14	√	√	√
0.015137	6 , 9	√	√	√
0.014957	11 , 12	√	□	√
0.014753	9 , 14	√	√	√
0.014634	9 , 11	□	□	√
0.014601	6 , 13	□	√	√
0.014261	4 , 11	□	□	□
0.014242	7 , 20	√	□	□
0.014203	2 , 9	□	√	√
0.013597	10 , 14	□	√	√
0.013272	5 , 8	√	√	√
0.013191	7 , 18	□	□	√
0.013151	4 , 8	√	√	√
0.013029	4 , 13	√	√	√
0.012323	7 , 9	√	√	√
0.012272	4 , 9	□	√	√
0.012103	4 , 7	□	√	√
0.011106	5 , 20	√	√	□
0.011067	12 , 19	√	√	□
0.01105	2 , 8	□	□	√
0.010561	2 , 14	√	√	√
0.009398	2 , 20	□	□	□
0.008869	11 , 18	□	√	√

0.008572	4 , 14	□	√	□
0.008524	2 , 18	□	□	□
0.008464	10 , 11	□	√	√
0.008357	11 , 19	□	√	□
0.008176	11 , 20	□	□	□
0.006091	4 , 18	√	□	□
0.006032	8 , 13	√	√	√
0.005103	4 , 20	□	√	□
0.004083	5 , 18	□	√	□
0.004012	18 , 20	□	□	□
0.002862	18 , 19	□	□	□
0.002243	19 , 20	□	√	□
0.000661	5 , 9	□	□	□
0.000367	7 , 10	√	□	√
0.000297	9 , 18	□	□	□
0.000286	2 , 19	□	√	□
0.000282	8 , 18	√	√	□
0.000258	12 , 20	□	√	□
0.000221	7 , 11	□	√	√
0.000216	9 , 20	□	□	□
0.000214	9 , 19	□	√	□
0.000174	6 , 20	□	√	□
0.000157	10 , 19	□	□	□
0.000135	13 , 20	□	□	□
0.000121	2 , 4	□	√	□
0.000116	12 , 18	□	□	□
0.000115	8 , 12	□	□	□
0.000109	14 , 19	□	√	□
0.000103	13 , 18	√	□	□
0.000102	8 , 11	√	√	□
0.000099	14 , 20	□	□	□
0.000084	6 , 14	√	□	□
0.000079	6 , 18	□	□	□
0.000075	5 , 19	□	√	□
0.000067	5 , 6	□	√	□
0.000067	9 , 10	□	□	□
0.000042	2 , 10	□	□	□
0.000037	13 , 19	□	√	□
0.000033	4 , 5	□	√	□
0.000031	2 , 5	□	□	√
0.000028	6 , 10	□	□	□
0.000026	5 , 7	√	□	□
0.000024	2 , 6	□	□	□
0.000019	5 , 12	□	√	□
0.000017	5 , 10	√	√	□
0.000011	2 , 7	□	□	□
0.000011	5 , 13	□	□	□
0.000001	6 , 11	□	□	□

Table 4-IX Capturing of binary relations by the three proposed structure identification algorithms.

All three algorithms do a good job at capturing the majority of the important binary relations. Let us look at the most important binary relation relating variables X_{12} and X_{14} with each other. This binary relation is captured by the FRA-based algorithm only. The way, all of these algorithms operate, by working their way back from the output to the inputs, a strong binary relation between inputs may be missed if neither of these inputs exhibits a strong binary relation with the output, as is the case in the given example. Yet, and this is interesting indeed, the FIR-based algorithm made use of X_{12} as an internal variable, but didn't include X_{14} among the inputs to be used by the optimal mask. Evidently, it found that X_{12} is better predictable from a combination of other variables. The FIR-based approach also employs a variety of less important variables as part of its inputs, because they still contain useful information that can be exploited. The reader is reminded that FIR is truly a simulation tool, whereas FRA and the statistical technique are structuring tools. It is very promising to recognize that techniques as different as FRA and the correlation method indeed come up with a relatively similar set of important binary relations.

5. Methodologies leading to a dynamic model

5.0 Abstract

In this chapter results from aspects discussed in Chapters 3 and 4 as well as new contributions are organised into (complete) methodologies to reduce the FIR mask search space. Section 5.2 presents different methods that lead to a static model; a comparison of model quality loss and computation alleviation with respect to FIR exhaustive static models is given. Subsection 5.2.1 discusses the results obtained with the methods explained in Chapter 3 as well as other regression based methods, all of them based on linear relationship search between variables. Subsection 5.2.2 deals with a proposal of methodologies that also take into account possible non-linear relations between variables. Section 5.3 deals with dynamic models constructed from searching linear as well as non-linear relationships between variables. An original methodology based on the results of Subsection 5.2.2.1 is proposed in Subsection 5.3.1, where time is included in the previously developed subsystem decomposition method. Finally, Subsection 5.3.2 offers a new energy-based approach to propose a reduced candidate mask to FIR.

5.1 Introduction

Chapters 3 and 4 dealt with model reduction and structure identification techniques for static models, i.e., for models whose input variables are all sampled at the same time instant as the output. Clearly, such models will rarely offer decent forecasting capabilities when applied to complex physical systems. It is simply a fact that most physical systems are dynamic in nature, and a decent model will have to take time into account. Why then our infatuation with static models through two entire chapters of this dissertation? What purpose does such a discussion serve?

The answer to this question is quite simple: Any methodology that can be used to generate static models can also be used to generate dynamic models. Figure 5-1 illustrates this truth. The same model that was already used in Chapter 2 to introduce the concept of a mask is reused here again. The system has five variables, i.e., it is a small-scale system. The mask is of depth three, i.e., the output at time *t* may depend on information sampled at time instants *t*, *t-δt*, and *t-2δt*. With a simple trick, the mask can be reduced to depth one. To this end, the data matrices (containing the class, membership, and side information) are triplicated. The copies are then concatenated to the original matrices from the right, each time shifted up by one row. The so modified system has now 15 variables instead of five, i.e., it is now a medium-scale system. However, a mask of depth one now contains the same information as the previous mask of depth three, i.e., instead of a dynamic model of a five-variable system, we can search for a static model of a 15-variable system.

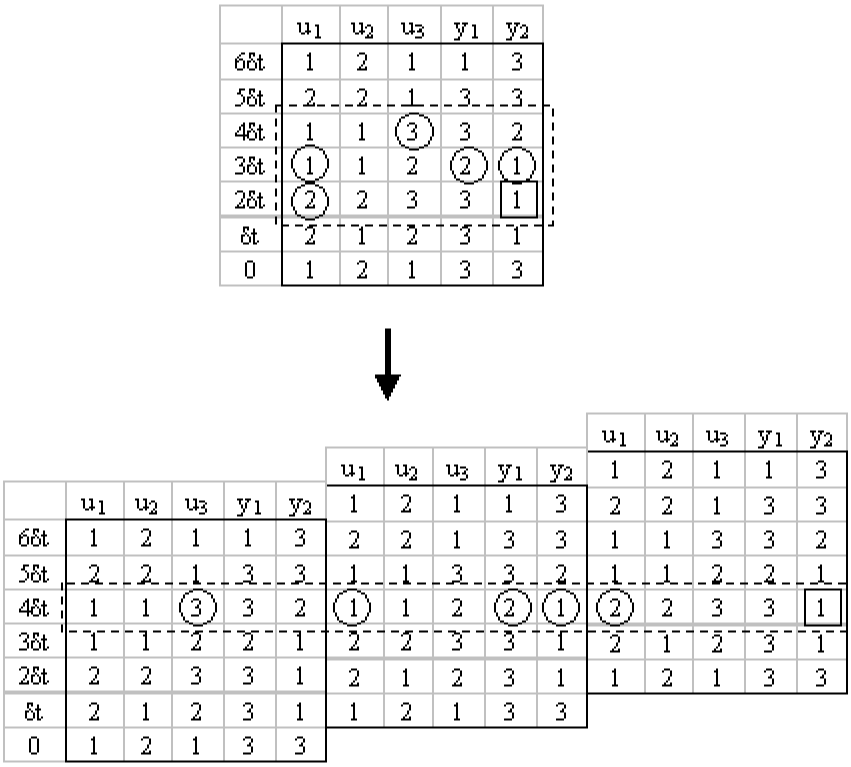


Figure 5-1 Expressing a 5-variable dynamic system by means of a 15-variable static one.

The same technique that was here demonstrated for the case of FIR can be applied to any inductive modelling approach. In order to allow the model inputs to be sampled at *k* different time instants, *k-1* copies of the data file(s) have to be made. These are concatenated to the original data from the right, each time moving the latest copy up by the time difference between the corresponding samples. In the process, an *n*-variable system turns into a *(k·n)*-variable system, i.e., a small-scale system gets converted to a medium-scale system, a medium-scale system becomes a large-scale system, and a large-scale system turns into a nightmare. If methodologies that can effectively and efficiently deal with large-scale systems were important for the identification of static models, they become truly essential when dealing with dynamic models.

Once the system has been converted in this way, any of the previously discussed techniques can be applied to it. Of course, the algorithms must be modified to take into account the issue of causality. No output may depend on an input at some future time; thus, causality adds a constraint on the legal combinations that need to be searched.

In the current discussion, FIR shall always serve as the baseline approach. The reason for this decision is that the research group to which this author belongs has convinced itself in numerous studies, such as [López, 1999], that FIR indeed offers excellent inductive modelling capabilities in that the forecasts obtained with FIR models are often the best and always very good. Thus, other techniques will be compared with FIR in terms of the reduction in computational complexity that they provide versus the reduction of model quality that the user has to accept. The purpose of the study is to identify computationally efficient algorithms that can be applied to large-scale systems and that can serve as precursors to a later FIR analysis, simplifying the subsequent FIR modelling task, such that it remains tractable.

The previous two chapters have demonstrated that FIR and FRA are techniques that are robust (in the sense that they do not require fine-tuning by the user), easy to use, and very general. The price to be paid for the sheer generality of these system-theory-based modelling techniques is their computational complexity. While there exist suboptimal search algorithms for both FIR and FRA, as investigated in [van Welden, 1999; Nebot and Jerez, 1997; Jerez and Nebot, 1997], some of these, such as the genetic algorithms, are unreliable in terms of their speed of convergence, whereas others, such as some of the faster hill-climbing techniques, are unreliable in terms of the quality of the suboptimal models that they generate. In this dissertation, we have looked at statistical approaches as precursors to FIR, since statistical techniques, though less general than FIR or FRA and therefore more difficult to manage, provide models of excellent quality at moderate computational cost, when applied correctly.

In Chapters 3 and 4, only a few statistical techniques have been looked at. In the current chapter, the statistical approaches to inductive modelling shall be investigated in a more thorough fashion to determine, which of them are most suitable as precursors to a subsequent FIR analysis.

As has been outlined in previous chapters, there are basically two lines of thought in the present dissertation, as depicted in Figure 5-2. The first one would be composed of those methods that do not reduce, a priori, the number of variables (m-inputs) present in the candidate matrix. Their aim is to simplify the mask search space of FIR trying to directly simplify the candidate mask that is proposed to FIR. This is based on reducing the number of '-1' elements of the candidate mask, and up to now it can be done using the increasing depth mask search algorithm proposed in Chapter 2. This algorithm starts out with a mask candidate of depth one. In response, FIR finds the best static model for the problem at hand. The mask depth is then iteratively incremented, each time adding a top row of '-1' elements to the mask candidate. Thus, instead of solving the dynamic modelling task in a single highly complex step, the problem is decomposed into a serious of computationally simpler problems, whereby the computational complexity of the iterative algorithm is still considerably lower than that of the original problem.

Along the same lines of thought, another possibility to reduce the computational complexity of a FIR model identification task, would be to gain knowledge about which are the most important delays in each of the considered system variables that contain the most complete information about the studied output. This idea will be further studied in Section 5.3.2 of this dissertation, where energy considerations are used to identify important delays. Each variable trajectory can be seen as the collection of values measuring a desired physical characteristic, such as the fuel flow through a pipe, plus an added noise, such as measuring noise or thermal noise. Each of these trajectories can be interpreted as a realisation of a stochastic process, i.e., there exists a deterministic as well as a random part in the available trajectory. With this interpretation, the energy of the signals can be computed and used to determine at which delays each input variable contains most energy relative to the output. In this way, a unique sparse candidate mask can be proposed to FIR so as to find a qualitative model of the underlying system.

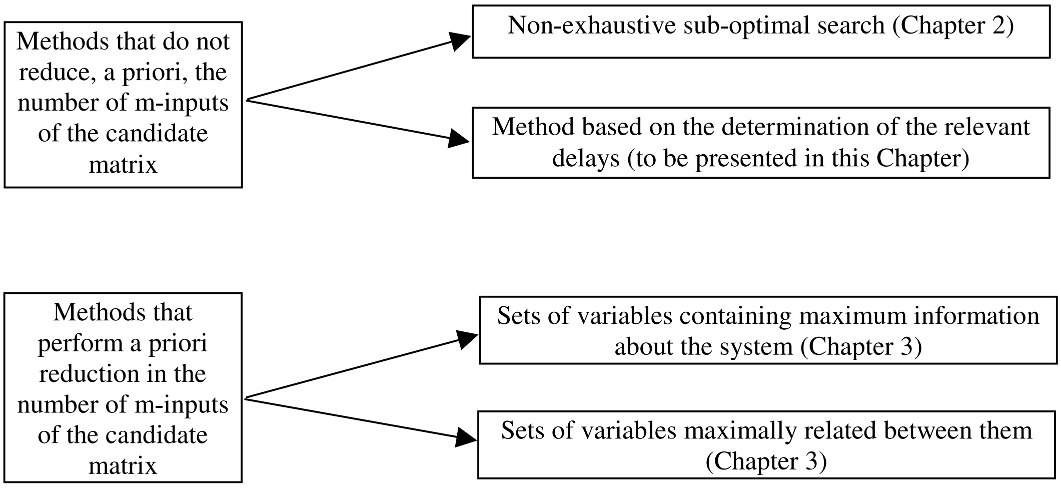

Methods that do not reduce, a priori, the number of m-inputs of the candidate matrix
Non-exhaustive sub-optimal search (Chapter 2)
Method based on the determination of the relevant delays (to be presented in this Chapter)
Methods that perform a priori reduction in the number of m-inputs of the candidate matrix
Sets of variables containing maximum information about the system (Chapter 3)
Sets of variables maximally related between them (Chapter 3)

Figure 5-2 The main lines of thought along the dissertation.

The second line of thought is the one exposed in Chapter 4, that is, to obtain a decomposition of the whole system into subsystems. This would allow obtaining a model of the system from its subsystems, which in turn simplifies the computational time needed for such an effort. Given a *k*-variable system, the cost of computing a unique *k*-variable model is much higher than computing some *p* models of $j_p < k$ variables. An efficient method for obtaining a decomposition can be formed using, in this order, the technique discussed in Section 3.4.1 followed by those of Sections 4.4 and 4.5. As already outlined in Section 4.5, the subsets of variables found in this way can then serve as candidate variables for a subsequent FIR analysis.

The chapter starts out by evaluating different statistical approaches to deriving static qualitative models of small-scale systems. In each case, the proposed technique is compared to FIR in terms of efficiency and resulting model quality. This is the content of Section 5.2. Then, in Section 5.3, it is explained how the subsystem decomposition method can be extended with time information, and the previously mentioned energy-based method, which inherently includes time information, is explained in detail. All of the examples used throughout the chapter are accurately described in Appendix I.

5.2 Methodologies that lead to a static model

As it was explained in Chapter 3, the term ‘static’ is used in the sense that, for a variable trajectory (quantitative) or episode (qualitative), no relations between delayed versions of the variables are considered, i.e., only the present values of the variables are taken into account. A static model is one that models the considered output from inputs and other system outputs at present time, i.e., the same time instant for which the model is trying to predict a new value of the output.

In that chapter, a first approach to reduce the model search space of the FIR qualitative modelling methodology was investigated with the aim of providing forecasts of trajectory behaviour of measured variables for control purposes. In particular, Chapter 3 dealt with the problem of pre-selecting a set of candidate input variables and hence reducing the model search space of the FIR methodology. To perform this variable selection, principal component analysis, the method of the unreconstructed variance, and correlation-based methods were used. Only linear static relations between variables composing the system were searched. Of course, a model that only contemplates linear static relations between its variables is not very useful for practical applications. Yet, it may help to identify, which are the most promising candidate variables to be used for modelling the output. We cannot ascertain that a discarded variable does not have any relation to the considered output, but we are sure that the chosen variables do have important relation to it.

This research presented in Section 5.2 is subdivided into two important subsections. The first one, Section 5.2.1, deals with methodologies that, either individually or in conjunction with other algorithms, can formulate a model of the system, but that only consider linear relations between variables. All these methods were first presented in [Mirats *et al.*, 2000]. The second one, Section 5.2.2, deals with methodologies that, using FIR as modelling engine, also consider non-linear relations between the system variables. All the techniques presented in Section 5.2 only deal with static relations, but as we meanwhile know, this constraint does not truly limit the generality of these algorithms.

All throughout the section, two situations should be distinguished. Some of the algorithms presented here, such as algorithms involving the use of regression coefficients, offer complete modelling approaches that can be considered alternatives to FIR. Other algorithms are auxiliary techniques used in conjunction with FIR. They are used exclusively to propose a candidate matrix to FIR, such that FIR can subsequently find a good model more economically than by means of exhaustive search.

5.2.1 Methods based on linear relationship searching

The different methodologies presented throughout this subsection are provided together with the results of applying them to the steam generator process explained in Appendix I.2. As outlined before, the methods used in this section only consider linear relations in the task of choosing or discarding variables for a posterior analysis[1].

The aim of this section is to evaluate the goodness of the different variable selection techniques used and should be compared to the variable selection achieved with the FIR methodology. In Section 5.2.1.1, static FIR models are obtained so as to compare the obtained results with the other methodologies.

Three different methods, presented in Subsections 5.2.1.2 to 5.2.1.4, are based on regression coefficients are used to analyse how to select a subset of variables to be retained within a model. These methods are ordinary least squares, principal component regression, and partial least squares. A general review of these methods can be found in [Jackson, 1991; Geladi and Kowalski, 1986]. Those regression coefficient methods do not need the FIR modelling engine, because they can formulate a linear static model by themselves. The contents of these three subsections is as follows:

- First, a brief description of the presented method is given.
- Then, the described method is used as originally formulated to model the output variable of the steam generator process. At this step, no variable selection is performed.
- Finally, a variable selection technique associated with the given method is used so as to obtain a set of selected variables. Then two models, using the described method in each of the subsections, are generated for the output variable of the boiler system: one using the selected variables from the given method, the other using the given method with the variables that FIR selects (as described in Appendix I.2).

In this way a comparison between the sets of variables that each of the presented techniques selects and the variables FIR that selects can be done.

Subsection 5.2.1.5 presents a clustering method that allows performing a variable selection. This method is then combined with FIR, because it is not a complete modelling methodology in its own right, so as to evaluate the goodness of the variable selection.

Finally in Subsection 5.2.1.6, all the derived sets of variables using the previously presented techniques, including the sets of variables obtained from the PCA methods discussed in Chapter 3, are used to propose to FIR a static candidate matrix. The quality of the obtained static models is then compared and discussed.

[1] Some of the methods used in this section will be combined with the FIR methodology. FIR is not limited to find linear relations; it also considers non-linear relationships between variables. Yet, if only a group of linearly related variables, derived from another method, is proposed to FIR as a candidate mask, we can talk of using FIR only considering linear relations.

Each one of the methods advocated in this section proposes a set of selected variables from which a static candidate mask could be proposed and a static FIR model obtained. Of course, as stated before, obtaining static qualitative models is not meaningful because none of them will be good enough to model a dynamic system. The reason for doing this is because the main objective is to reduce the model search space of the FIR methodology, and in order to attain this goal, the research must be done following some order. In this dissertation, first linear static relations between variables are explored, then non-linear relations are tackled, then time is added, i.e., dynamic relations are searched. In each stage, the reduction on the computational cost achieved when computing a FIR model as well as the quality of the resulting model are reported, so as to assess the success of the investigated technique relative to the proposed objectives.

5.2.1.1 FIR models excluding temporal relations

In order to be able to compare the quality and the computing reduction achieved when using different variable selection techniques, static *FIR* models of the boiler example are needed, that is, models excluding temporal relationships. The candidate mask of depth 1, allowing all the possible static relations between the system variables, proposed for this purpose is:

$$mcan = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 \end{pmatrix}$$

Table 5-I shows the retained masks for each of the allowed complexities, as well as the normalised MSE[2] value of the prediction using each of them.

Complexity	Model	Quality	MSE
2	(0 0 0 0 0 0 0 -1 1)	0.1532	1.1510
3	(0 0 0 0 0 0 -1 -2 1)	0.1523	1.2087
4	(0 0 -1 -2 0 0 0 -3 1)	0.1418	1.0974
5	(0 -1 -2 0 -3 0 0 -4 1)	0.0900	0.8234

Table 5-I Boiler FIR models without temporal relations.

FIR concludes that variables 1 and 6 can be safely discarded, whereas all other variables need to be retained for the time being. All of the static models found are of fairly low quality, and the MSE values resulting from their use are consequently rather high. This is not further surprising, as a static model cannot be expected to offer accurate predictions of future output values for this system.

Notice that, in the FIR methodology, the quality of masks can only be truly compared to each other as long as their complexities are the same. Moreover, for a given mask, the MSE value depends on the validation data set, and both considerations must be taken into account when attempting to extract conclusions from Table 5-I.

[2] Later on, some statistical techniques based on regression coefficients, such as OLS, PLS, and PCR are used to derive a linear static model of the same example. Those methods normalise data previous to analysing them. In order to compare FIR with those other methodologies using as index the MSE, that of FIR has to be normalised.

Section 5.2.1.1 was only added for completeness and readability. It only represents an excerpt of some results presented in Appendix I.2 in more detail.

5.2.1.2 Ordinary Least Squares method

Given n observations of an input/output system with k input or predictor variables and one output or response variable, the traditional regression model can be written as:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{e}$$

where **X** denotes an $n \ x \ k$ matrix of observations of the input variables normalised to zero mean and unit variance, **y** denotes an $n \ x \ 1$ vector of the output also normalised, **b** a $k \ x \ 1$ vector of regression coefficients, and **e** an $n \ x \ 1$ vector of residuals (also called perturbations in the literature, that is, **e** is the effect of all the variables that affect variable y and that are not included in the model). The least squares solution for **b** is:

$$\mathbf{b} = [\mathbf{X'}\ \mathbf{X}]^{-1}\ \mathbf{X'}\ \mathbf{y}$$

This expression is obtained under the following considerations about the residuals:

- Their expectation is zero, $E[e_i] = 0$, and their variance is constant, σ^2.

- Perturbations are independent, $E[e_i e_j] = 0$, $i \neq j$, and their distributions are normal.

Table 5-II shows the regression coefficients found when using the training data set, i.e., the 85% of the gathered data, to compute them. The left column shows the regression coefficients, the second column expresses the 95% confidence interval for each regression coefficient computed in accordance with equation (5.1), and the last column shows the percentage that each coefficient contributes to variable y.

$$\hat{\beta}_i \pm t_{n-k-1}(\alpha/2)\hat{s}_R \sqrt{q_{ii}} \qquad (5.1)$$

In equation (5.1), $\hat{\beta}_i$ accounts for the regression coefficients, $t(\alpha/2)$ is the Student-Fischer distribution with a probability threshold of $(1-\alpha/2)$ and $n\text{-}k\text{-}1$ degrees of freedom, $\hat{s}_R$ is the estimation of the residual standard deviation, and q_{ii} are the diagonal elements of the matrix $(X'X)^{-1}$.

Coefficient	95% confidence interval	% contribution
-1.8925	-3.0524 -0.7326	20.1015
2.3633	1.3890 3.3377	25.1023
0.3493	-0.1639 0.4346	3.7098
1.8467	0.5612 3.1322	19.6146
0.5028	0.0598 0.9459	5.3408
-0.1847	-0.6177 0.2483	1.9620
-2.1369	-2.7344 -1.5394	22.6976
0.1385	-0.0976 0.3746	1.4714

Table 5-II Regression coefficients using OLS with the boiler system.

The Ordinary Least Squares (OLS) technique is a very simple technique, and consequently, more refined techniques should be rejected if they cannot outperform OLS. Using an OLS model, it might be expected that the best prediction could be obtained when all variables (all regression coefficients) are being used. Figure 5-3 shows an OLS prediction of the validation data, i.e., the 15% of data not used for deriving the model, of the boiler system using all the input variables. As in other figures along the dissertation, the continuous line shows the measurement data of the output variable during the validation period. The dashed line shows the predictions made by the considered model The MSE value for the training data set is 0.6466, and for the validation data set, it is 1.1973.

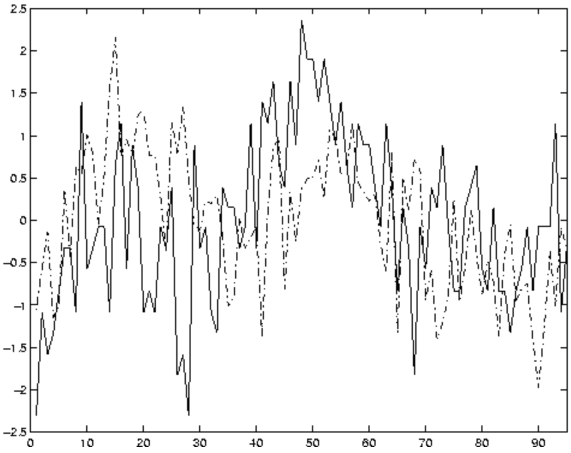


Figure 5-3 Boiler OLS prediction using all input variables.

Contrary to the PCA models presented in Section 3.3.2, when analysing the variable selection achieved with the unreconstructed variance method (see Figures 3-2 and 3-3), the OLS model represents also the higher frequency oscillatory components of the behaviour. Yet, the MSE values are still larger than in the case of the PCA model.

How can the regression coefficients be used to determine a subset of variables to be retained in a simplified model? Different variable selection methods can be found in the literature. In [Peña, 1989], a statistical analysis based on the *t* distribution is performed to check whether the i^{th} regression coefficient can assume a value of 0 or not with a given probability. In [Daling and Tamura, 1970; Lindgren *et* al., 1995], those variables with smaller coefficients in the regression equation are discarded. These two methods are similar to each other. When computing the confidence interval for a small coefficient, the probability that 0 is within this interval is high. In other words, when performing a *t-test* on this coefficient, the observed *t* value, computed in accordance with equation (5.2), will not be of significant magnitude, and therefore, the hypothesis $\hat{\beta}_i = 0$ cannot be refused. Consequently, this variable should not be taken into consideration within the model.

$$t_{obs} = \frac{\hat{\beta}_i}{\hat{s}_R \sqrt{q_{ii}}} \tag{5.2}$$

The criterion adopted here has been to select those variables with significant contribution to the regression equation, and to drop those with smaller contributions. The cut-off between selected and discarded variables is set to be 5% of contribution to the total regression line.

Applying this criterion to the obtained coefficients in Table 5-II, variables 1, 2, 4, 5, and 7 are to be retained. Their regression coefficients are recomputed after eliminating from the **X** matrix those columns corresponding to the discarded input variables. The newly obtained regression coefficients are tabulated in the left column of Table 5-III.

Coef. 1,2,4,5,7	Coef. 2,3,8
0.7389	0.6693
0.7051	0.0844
0.7575	-0.013
0.6024	
-2.152	

Table 5-III. Regression coefficients for the boiler variables 1, 2, 4, 5, and 7 (left column) and for variables 2, 3, and 8 (right column).

The top portion of Figure 5-4 shows the real and predicted validation output values when using a regression model with variables 1, 2, 4, 5, and 7. The MSE value is 0.7483 when predicting the training data set and 1.3250 when predicting the validation data set. The prediction turns out to be a little poorer than that obtained when keeping all variables in the regression analysis.

Remembering the results obtained with the PCA models (Chapter 3), it may be of interest to check what happens when those variables that FIR suggested to retain are kept in the regression model, i.e., variables 2, 3, and 8 (see Appendix I.2). The regression coefficients for this case are tabulated in the right column of Table 5-III. The bottom portion of Figure 5-4 shows the corresponding prediction. In this case, the MSE value is 0.6482 for the training data set and 1.0703 for the validation data set. The results are clearly better than using the variables that the OLS modelling approach suggested to retain, and in the case of the validation data set, the results are even better than keeping all variables in the regression model.

As it happened in the case of the PCA analysis performed in Section 3.2, the FIR qualitative modelling engine did a better job than the OLS modelling approach in terms of deciding, which variables should be kept in the model and which should be discarded from it. This is true not only for the purpose of a FIR qualitative simulation, but also when a PCA or OLS simulation is to be made.

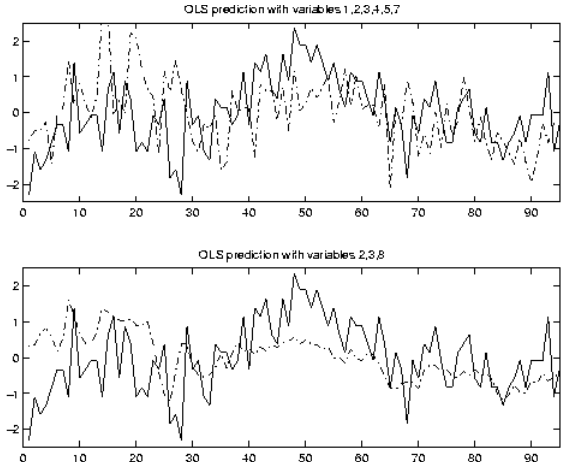


Figure 5-4 Boiler OLS predictions using variables 1, 2, 4, 5, and 7 (top), and variables 2, 3, and 8 (bottom).

5.2.1.3 Principal Components Regression model

In this methodology, the input variables are transformed to principal components before calculating the regression coefficients. The Principal Components Regression (PCR) model thus pre-processes the input data, converting them to a set of equivalent PCs. It then uses those PCs to estimate the output by means of an OLS approach.

A *PCR* analysis has been performed on the training data set of the boiler system, and a cross-validation method [Wold, 1978; Osten, 1988] has been used to determine the number of Latent Variables (LVs) to be kept in the regression model. To this end, the training data have been split into 10 blocks of equal size, and the Predictive Residual Error Sum of Squares (PRESS) value has been calculated for each of them.

Analysing the results obtained, it was decided to retain 4 of the possible 8 latent variables in the regression model, and the regression coefficients were subsequently calculated. Since the PCA analysis results in a linear transformation on the input space, it is possible to transform the resulting regression coefficients for the 4 PCs back to 8 equivalent regression coefficients for the original input variables. Table 5-IV shows those coefficients (first column) as well as the percentage that each coefficient contributes to the regression equation (second column).

Figure 5-5 shows the predictions obtained for the validation data set using a PCR model made of all input variables whereby the 4 most important PCs (latent variables) were retained. The resulting MSE values are 0.7189 for the training data set and 1.2798 for the validation data set.

Coefficients	% contribution
-0.0404	1.5098
-0.1418	5.2951
0.0618	2.3079
0.2670	9.9665
1.2325	46.0144
-0.3242	12.1031
-0.4718	17.6139
0.1390	5.1893

Table 5-IV Regression coefficients for all variables when using PCR with the boiler system.

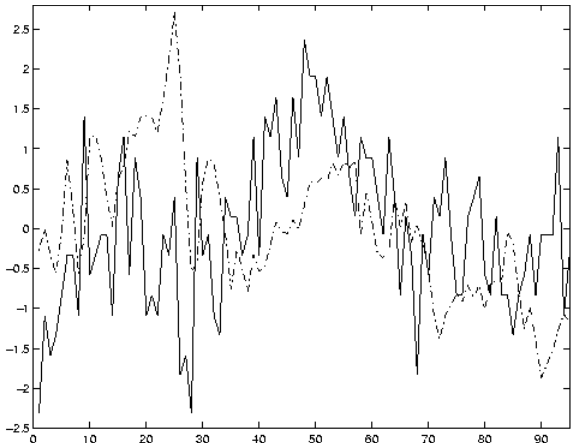


Figure 5-5 Boiler PCR model using all input variables retaining 4 LVs.

It may make sense to throw some of the input variables out from the beginning. Using the same criterion that was applied in the case of the OLS model, the regression coefficients of Table 5-IV suggest that variables 1 and 3 be discarded from the model, while variables 2, 4, 5, 6, 7, and 8 are to be retained. A PCR model for this reduced set of input variables was subsequently computed. Cross-validation revealed that, in this case, 5 of the 6 possible LVs ought to be retained in the regression model. The left column of Table 5-V lists the regression coefficients transformed back to the original input variables. The top portion of Figure 5-6 shows the prediction using this PCR model. The resulting MSE values are 0.6866 for the training data set and 1.1587 for the validation data set. The results are slightly better than for the PCR model involving all input variables.

Just like in the previous section, a third PCR model was then calculated using input variables 2, 3, and 8, as proposed by FIR in Appendix I.2. In this case, 2 of the possible 3 LVs are to be retained. The right column of Table 5-V shows the corresponding regression coefficients. The bottom portion of Figure 5-6 shows the predictions obtained. In this case, the resulting MSE values are 0.7532 for the training data and 1.0645 for the validation data.

Coef. 2,4,5,6,7,8	Coef. 2,3,8
0.7260	0.3271
0.9852	0.0697
0.6313	0.3297
0.0787	
-1.9695	
0.2074	

Table 5-V. Boiler regression coefficients for variables 2, 4, 5, 6, 7, and 8 (left column); and variables 2, 3, and 8 (right column).

The results here are less good for the training data, but consistent with the results obtained for the previous two methods, OLS in previous subsection and PCA in Chapter 3, they are better in the case of the validation data set. These results could be interpreted in such a way as to suggest that statistical techniques offer decent *interpolation* capabilities, but FIR exhibits better *generalisation* power.

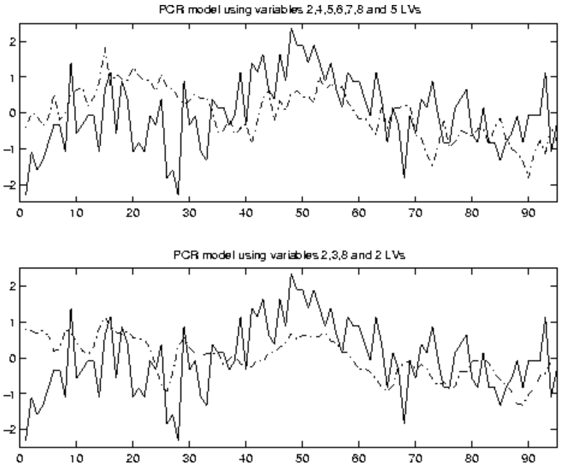


Figure 5-6 Prediction of the boiler output using a PCR model with variables 2, 4, 5, 6, 7, and 8 (top); and 2, 3, and 8 (bottom).

5.2.1.4 Partial Least Squares regression method

A brief description of the PLS technique is included in this subsection. For a full description, the reader is encouraged to review the extensive literature written on this methodology, for example [Jackson, 1991; Geladi and Kowalski, 1986] offer a good review on this issue.

The PLS (PLS based regression) technique operates in a similar form as PCR in the sense that a set of vectors is obtained from the predictor (input) variables. The main difference is that as each vector is obtained, it is related to the responses and the reduction of variability of the inputs. The estimation of the next vector takes into account this relationship, and simultaneously, a set of vectors for the outputs it is also obtained that takes into account such a relationship.

PLS has often been presented as an algorithm rather than a linear model, it is based on the NIPALS algorithm (a least squares algorithm for obtaining principals components). In this brief review of the method, the notation offered in [Geladi and Kowalski, 1986] has been used.

Consider $\boldsymbol{X}$ and $\boldsymbol{Y}$ real data matrices of size $n\ x\ p$ and $n\ x\ q$ respectively, representing n observations on p input and q output variables. The first step is to normalise both $\boldsymbol{X}$ and $\boldsymbol{Y}$ to zero mean and unit variance, then two operations are carried out together:

$$\boldsymbol{X} = \boldsymbol{TP} + \boldsymbol{E} \quad (\boldsymbol{T} \text{ has size } n\ x\ k,\ \boldsymbol{P} \text{ has size } k\ x\ p, \text{ and } \boldsymbol{E} \text{ has size } n\ x\ p)$$
$$\boldsymbol{Y} = \boldsymbol{UQ} + \boldsymbol{F^*} \ (\boldsymbol{U} \text{ has size } n\ x\ k,\ \boldsymbol{Q} \text{ has size } k\ x\ q\ , \text{ and } \boldsymbol{F^*} \text{ has size } n\ x\ q)$$

$k \leq q$ is the number of vectors associated with $\boldsymbol{X}$. $\boldsymbol{E}$ is the matrix of residuals of $\boldsymbol{X}$ at the k^{th} stage (when $k = p$, $\boldsymbol{E} = 0$). $\boldsymbol{F^*}$ is an intermediate step in obtaining the residuals for $\mathbf{Y}$ at the k^{th} stage.

In the singular value decomposition associated with PCA, matrices $\boldsymbol{Q}$ and $\boldsymbol{P}$ would be the characteristic vectors, and matrices $\boldsymbol{T}$ and $\boldsymbol{U}$ the principal component scores. These matrices do not have the same properties in PLS, but may still be thought of in the same vein; $\boldsymbol{T}$ and $\boldsymbol{U}$ are referred to as *X-scores* and *Y-scores,* respectively.

It is possible to use regression to predict the output block of variables from the input one. This is done decomposing the $\boldsymbol{X}$ block and building up the $\boldsymbol{Y}$ block. In PLS, a prediction equation is formed by:

$$\boldsymbol{Y} = \boldsymbol{TBQ} + \boldsymbol{F}$$

where $\boldsymbol{F}$ is the actual matrix of residuals for $\boldsymbol{Y}$ at the k^{th} stage, and $\boldsymbol{B}$ is a transformation matrix of size $k\ x\ k$.

It is possible to calculate as many PLS components as the rank of the $\boldsymbol{X}$ matrix, but not all of them are normally used. In order to decide how many components (also referred to as latent variables) to use there are several methods advocated in the literature. One of them is using the number of components that minimises a measure of PRESS (predictive residual error sum of squares).

Hence, the Partial Least Squares (PLS) method operates in similar ways as the PCR method. The PCR method transforms the input space into a set of PCs. However, it does not do anything to the outputs. The PLS method transforms both the inputs and the outputs to sets of PCs using a PCA analysis, and in addition, takes into account the relationship between the input space and the output space.

The PLS technique has been applied to the training data of the boiler system. Just like in the PCR method, it is necessary to decide how many latent variables are to be retained in the regression model. Hence cross-validation was used, splitting the available data into 10 blocks of equal sizes. The result of this study was that 4 of the possible 8 latent variables are to be retained in the regression model. As in the previous method, the regression coefficients found for the 4 retained PCs were then translated back to equivalent regression coefficients for the 8 original input variables. Table 5-VI lists, in its first column, the resulting regression coefficients, and in the second column, the percentage that each one of them contributes to the regression equation.

Coefficients	% contribution
-0.1660	2.4436
1.6099	22.6983
0.2671	3.9315
1.5364	22.6154
0.4340	6.3880
-0.3468	5.1052
-2.4332	35.8170
0.0001	0.0009

Table 5-VI Regression coefficients for all variables when using PLS with the boiler system.

Figure 5-7 shows the prediction of the validation data set for the *PLS* model using all input variables and retaining 4 latent variables. The resulting MSE values are 0.6536 for the training data set, and 1.1192 for the validation data set.

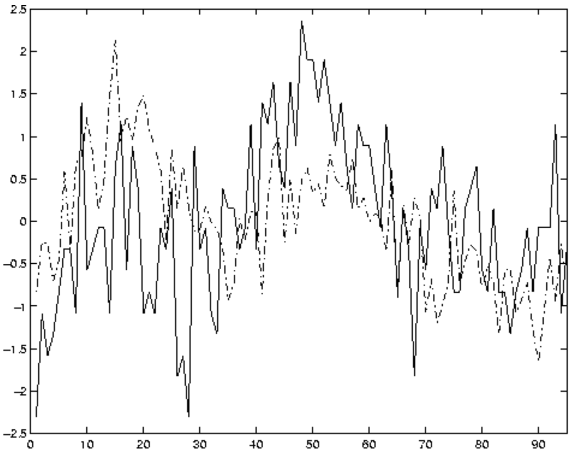


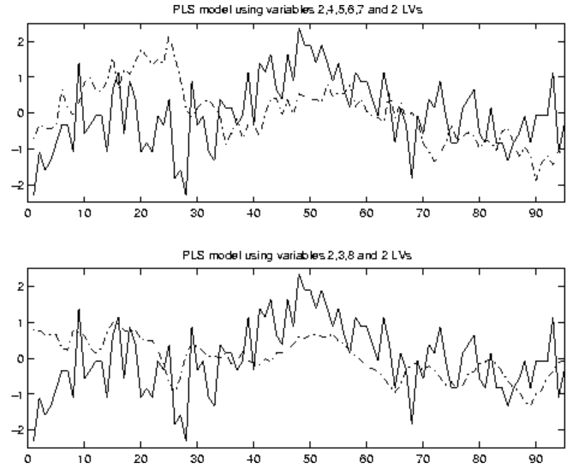
Figure 5-7 PLS model using 4 LVs and all physical variables.

Next, variable selection is performed. The same criterion is used as in the two previous sections. This time, variables 2, 4, 5, 6, and 7 are selected. The PLS model for these variables was then calculated. After performing the cross-validation test, it was decided to retain 2 of the possible 5 LVs. The left column of Table 5-VII lists the regression coefficients obtained for this group of variables. The top portion of Figure 5-8 shows the prediction of the validation data set using a PLS model in the 5 selected variables with 2 LVs retained. The resulting MSE values are 0.6967 for the training data set, and 1.2429 for the validation data set. Hence the results are a little poorer than in the previous case, where all variables had been used.

As in the previous two sections, a comparison was made with a PLS model in variables 2, 3, and 8, as proposed by FIR in the Appendix I.2. Cross-validation revealed that 2 of the possible 3 LVs ought to be retained. The resulting regression coefficients are listed in the right column of Table 5-VII. The bottom portion of Figure 5-8 shows the prediction obtained using this model. The resulting MSE values are 0.7527 for the training data set, and 1.0484 for the validation data set. As in the case of the PCR analysis, the results are poorer for the training data (reduced interpolation capability) but better for the validation data (improved generalisation power).

Coef. 2,4,5,6,7	Coef. 2,3,8
0.4469	0.3461
0.7674	0.0790
1.0934	0.3100
-0.4233	
-1.2280	

Table 5-VII. Boiler regression coefficients for a PLS model for variables 2,4,5,6, and 7 (left column); and variables 2,3, and 8 (right column).



PLS model using variables 2,4,5,6,7 and 2 LVs

PLS model using variables 2,3,8 and 2 LVs

Figure 5-8 Prediction of the boiler output using a PLS model with physical variables 2, 4, 5, 6, and 7 (top); and 2, 3, and 8 (bottom).

5.2.1.5 Methods based on cluste r analysis

The two last methods to be discussed in this section are based on cluster analysis. Although many clustering methods have been reported in the literature, only two of those methods have been included here.

Before discussing the results obtained, a sketch of the four steps of the clustering methods applied in this study is presented.

First, it is necessary to define a measure of similarity, say R_{XY}, between two groups of variables X and Y. Two such metrics will be used here in order to compare the two different clustering methods.

The first metric is for a single-linkage method and is given by

$$R_{XY} = \max_{\substack{i \in X \\ j \in Y}} r_{ij} \tag{5.3}$$

The second measure is for an average-linkage method and is given by

$$R_{XY} = \left(\sum_{i \in X} \sum_{j \in Y} r_{ij} \right) \Big/ (n_1 n_2) \tag{5.4}$$

where r_{ij} are the correlation coefficients between variables i and j; and n_1 and n_2 are the number of variables contained in groups X and Y, respectively.

Second, once the measure of similarity has been chosen, R_{XY} is computed for each of the $k(k\text{-}1)/2$ pairs of single variable groups of a k-variable[3] system.

Third step: if A and B are the two groups for which R_{AB} is a maximum, replace A and B by the single group $C = A \cup B$.

Fourth and last step: for each group not involved in the previous step, calculate R_{XC} and return to the third step.

The clustering process cycles between steps three and four until the number of groups has decreased to a sufficiently small value. Two more decisions need to be made: (1) when to stop with the clustering algorithm, and (2) which variable to choose from each of the obtained clusters.

A criterion for terminating the clustering algorithm is to continue with steps three and four until all R_{XY} between the remaining clusters fall below some level R_0. In accordance with [Jollife 1972; Jollife 1973], the following values for R_0 have been taken: 0.55, when the single-linkage method is used, and 0.45, when the average-linkage method is applied. Two methods of choosing variables from each cluster have been implemented:

[3] In this analysis all the system variables form part of either one group of another, i.e., $k = n_1+n_2$.

- **Inner-clustering** selects one of the first two variables forming each cluster.

- **Outer-clustering** selects the last variable that joined each cluster.

Table 5-VIII summarises the results obtained when applying this method to the boiler system. Using either the single-linkage method or the average-linkage method leads to the same results, thus, Table 5-VIII is valid for both metrics. The variables in each cluster are given in the order in which they joined the cluster.

Method	Inner clustering	Outer clustering
Variables kept	2,3	3,8
Clusters found	2,4,1,7,6,5,8 3	2,4,1,7,6,5,8 3

Table 5-VIII Boiler variable selection achieved using cluster analysis.

The reader may notice that both techniques selected a subset of the variables that FIR proposed to retain. Combining the inner and outer clustering methods, the set of variables determined by FIR would have been found in the case of the example at hand.

5.2.1.6 Using subsets of variables for static FIR predictions

To this point, the predictions made by statistical techniques, as presented in Subsections 5.2.1.2 to 5.2.1.5, were compared against prediction made by these very same techniques if the variables used to model the output of the boiler system would be those proposed by FIR in Appendix I.2 (Figure I-8). Such a comparison is unfair, because all statistical techniques presented in this section employed *static* models only, whereas FIR made use of a *dynamic* (time dependent) model. All techniques discussed in Subsections 5.2.1.2 to 5.2.1.5 could also have been used to generate dynamic models by simply duplicating and triplicating the sets of variables, i.e., the columns of the data matrix, shifting them up each time by one row, as explained in the introduction to this chapter. This form of adding time to the analysis will be used later for the case of decomposing the system into subsystems. The computational work for these methods would have been enlarged, because they would have had to deal with more variables in this way, but the predictions would certainly have been improved.

Yet, this was not the purpose of this section. The purpose was to find a set of bootstrapping techniques for FIR that, with little computational effort, could encounter subsets of variables to be considered in a FIR optimal mask search. To this end, static models, which can be obtained easily at low computational cost, have been generated, in the hope that the variables not used by these models would be less likely candidates also in a dynamic model search. Such an assumption makes sense due to the autocorrelation inherent in any physical signal. If *x(t)* is a poor choice for predicting *y(t),* then it is unlikely that *x(t-δt)* or *x(t-2δt)* would be very good choices either.

Method	Selected variables	Mask Qualities		Number of computed masks		MSE
		C4	C5	C4	C5	
None	All	0.6080	0.6196	82160	1581580	0.5522
FIR (Static)	2,3,4,5,7,8	0.6080	0.6196	37820	557845	0.5522
Unreconstr. var.	1,2,4,5,6,7,8	0.5943	0.4290	59640	1028790	0.7324
OLS	1,2,4,5,7	0.5943	0.4258	23426	292825	0.7516
PCR	2,4,5,6,7,8	0.5943	0.4290	37820	557845	0.7324
PLS	2,4,5,6,7	0.5943	0.4258	23426	292825	0.7516
MC*	2,3	0.6080	0.6167	2600	14950	0.5529
PCA_A1*	3,8	0.6057	0.6177	2600	14950	0.5845
PCA_A2*	2,3	0.6080	0.6167	2600	14950	0.5529
PCA_B1*	1,2	0.5943	0.4258	2600	14950	0.7516
PCA_B2*	1,3	0.6049	0.6123	2600	14950	0.5640
Inner clust.	2,3	0.6080	0.6167	2600	14950	0.5529
Outer clust.	3,8	0.6057	0.6177	2600	14950	0.5845

Table 5-IX Boiler FIR Dynamical models obtained from different reduced sets of variables.

Table 5-IX lists the results of employing the static methods obtained in previous sections as bootstrapping techniques for a dynamic model search using FIR.

The first row of Table 5-IX shows the results of performing an optimal mask search using all 9 variables and a candidate mask of depth 16, as it has been done in the Appendix I.2. Hence the corresponding candidate mask contains 16·9-1 = 143 potential inputs ('-1' elements). An exhaustive search was performed analysing the quality of masks consisting of up to four of these inputs plus the output. To this end, 1.581.580 masks had to be evaluated. The search consumed 160 minutes of computation time on a Sun Ultra Sparc II Workstation. The optimal masks of complexities 4 and 5 (columns C4 and C5) were tabulated with respect to their resulting mask qualities (columns 3 and 4). The number of different masks visited in the process of searching for the optimal mask of complexity 4 is listed in column 5, and the number of masks visited in search of the optimal mask of complexity 5 is presented in column 6. Column 7 shows the MSE error obtained in a prediction that combines the predictions made by the optimal masks of complexities 4 and 5. As discussed in [López, 1999], FIR not only makes a prediction of an output variable; it simultaneously provides a measure of confidence in its own prediction. In the simulation leading to the MSE value reported in column 7, predictions were made in parallel with the optimal masks of complexities 4 and 5, and in each step, the prediction accompanied by the larger confidence value was kept.

The subsequent rows tabulate the results obtained when using the optimal mask search algorithm but using the sets of variables proposed by the different studied methods. For example, the second row tabulates the results of the performed search where the variables to be considered were obtained using the static FIR models found in Section 5.2.1.1 of this chapter. Those models suggested that variables 1 and 6 are less likely candidates for input

[*] Those methods have been presented in chapter 3. MC stands for Multiple correlation, PCA stands for principal components analysis.

variables. Consequently, variables 1 and 6 were discarded from the set of potential inputs. This is shown in column 2. The mask candidate now contains 0 elements (forbidden connections) in columns 1 and 6, and consequently, it only contains 16·7-1 = 111 potential inputs. The subsequent exhaustive search through all masks compatible with the mask candidate matrix and the constraint of not having more than 4 inputs resulted in a search through 557.845 masks. Hence the computational effort was about 1/3 of the one needed for the experiment described in the previous paragraph. It consumed 53 minutes of execution time on a Sun Ultra Sparc II Workstation. It turned out that FIR did a good job at discarding variables. The resulting masks of complexities 4 and 5 are exactly the same as found using the three times more expensive search through all possible masks.

Rows three to six correspond to the results obtained with a static FIR model using the set of variables proposed by each of the techniques discussed in Subsections 5.2.1.2 to 5.2.1.5. The regression techniques of Subsections 5.2.1.2 to 5.2.1.4 as well as the unreconstructed variance methodology explained in Chapter 3, were not well suited for the task at hand. All of these techniques threw out variable 3, which turned out to be essential in making good FIR predictions. This variable was discarded, because it exhibits a relatively poor cross-correlation with the other variables. Consequently, these statistical techniques considered the variable of lesser relevance. This decision let to optimal masks of reduced quality, and as was to be expected, the use of these masks in a FIR prediction led to substantially larger prediction errors.

Obviously, cross-correlation only evaluates the strengths of *linear* relationships, whereas the FIR forecasting engine exploits also *non-linear* relationships among variables. Yet, this does not fully explain the comparatively poor performance of these techniques, since even the described statistical modelling techniques, using perfectly linear regression models, exhibit better prediction results when they are applied to the set of variables selected by FIR (which includes variable 3) than when they are based on their own variable selection. Evidently, and in spite of its relatively poor cross-correlation with the other variables, variable 3 still contained valuable information that could be exploited in predictions.

The final set of rows summarises the results of using FIR with the variables selected by each of the techniques presented in Sections 3.2 and 3.4.2, as well as the clustering methods discussed in Subsection 5.2.1.5. These techniques performed considerably better than those based on regression coefficients or unreconstructed variance. Except for method PCA_B1, all methods resulted in optimal masks that were either the truly optimal ones, or at least of almost equal qualities. In accordance with this finding, also the resulting MSE values were close to optimal. Why did these techniques work better? The reason is that they did not attempt to eliminate variables with poor cross-correlation to the output. Instead, they eliminate variables with strong cross-correlation to other inputs. This makes sense, because if two inputs are strongly correlated with each other, they contain almost identical information, and therefore, either one of them will suffice to explain the output. This strategy works even in the case of non-linear systems and for use by non-linear prediction algorithms.

The techniques presented in Chapter 3 and the clustering methods are considerably more aggressive in throwing out variables than the other algorithms presented in this chaopter. Since the set of remaining variables is small, the optimal mask search, for the given example, can be performed quickly. These searches are completed on a Sun Sparc II Workstation within less than 2 minutes, i.e., they execute about 100 times faster. All of these techniques exhibit another important advantage. They sort the variables in order of increasing or decreasing importance. Hence it would be easy to add one more variable and repeat the optimal mask search to check whether or not the mask quality improves. This could still be done rather inexpensively.

Now, another experiment of interest can be performed with the different obtained sets of variables. Those can be used to propose to FIR a candidate mask with a given depth[4], so as to obtain a dynamic FIR model, but only setting to '-1' the columns corresponding to the variables selected by each of the methods at a time. The loss of prediction quality incurred when computing FIR dynamic models from the different subsets of variables is summarised in Table 5-X. The columns labelled *P* stand for the percentage of mask quality lost by the different variable subsets relative to that of the exhaustive FIR model of equal complexity. *P* is computed as:

$$P = \frac{Q_{FIR\,dynamic} - Q_{MODELi}}{Q_{FIR\,dynamic}} \cdot 100$$

The columns labelled *N* represent the percentage of MSE error increase due to the selection of different subsets of variables relative to that of the exhaustive FIR model. *N* is computed as:

$$N = \frac{MSE_{MODELi} - MSE_{FIR\,dynamic}}{MSE_{FIR\,dynamic}} \cdot 100$$

The values in the rows labelled C+ were computed combining the predictions made by the optimal masks of complexities 4 and 5 as explained earlier.

	FIR (dynamic)		FIR (static)		Unrecons. variance		PCR		OLS, PLS	
	Q	P	Q	P	Q	P	Q	P	Q	P
C4	0.6080	0	0.6080	0	0.5943	2.2	0.5943	2.2	0.5943	2.2
C5	0.6196	0	0.6196	0	0.4290	30.7	0.4290	30.7	0.4258	31.3
	MSE	N	MSE	N	MSE	N	MSE	N	MSE	N
C+	0.5522	0	0.5522	0	0.7324	32.6	0.7324	32.6	0.7516	36.1

[4] Here a depth of 5 has been used so as to be consistent with the analysis performed in the Appendix I.2

	MC*, PCA_A2*, Inner clustering		PCA_A1*, Outer Clustering		PCA_B1*		PCA_B2*	
	Q	P	Q	P	Q	P	Q	P
C4	0.6080	0	0.6057	0.4	0.5943	2.2	0.6049	0.51
C5	0.6167	0.5	0.6177	0.3	0.4258	31.3	0.6123	1.2
C*	MSE	N	MSE	N	MSE	N	MSE	N
	0.5529	0.1	0.5845	5.9	0.7516	36.1	0.5640	2.14

Table 5-X Loss of prediction quality due to selection of variable subsets.

From Table 5-X, it is evident that there exists a strong positive correlation between the percentage-wise reduction in mask quality and the corresponding increase in prediction error, at least for the example at hand.

Table 5-XI shows the reduction in computing effort attained when using FIR with each one of the proposed subsets of variables. Each column stands for the reduction in the number of masks to compute and it has been calculated as:

$$R = \frac{\#_{FIR\,dynamic} - \#_{MODELi}}{\#_{FIR\,dynamic}} \cdot 100$$

where # is the number of masks to be evaluated for a given complexity. The last column in Table 5-XI relates to methods *MC, PCA_A1, PCA_A2, PCA_B1, PCA_B2*, inner and outer clustering, because they all achieve the same reduction of the FIR model search space.

	FIR (dynamic)	FIR (static) PCR	Unreconstructed variance	OLS, PLS	MC, PCA_A1...
C4	0	53.97	27.41	71.49	96.84
C5	0	64.73	34.95	81.49	99.05

Table 5-XI Model search space reduction attained with each of the methods applied to the boiler system.

[*] Methods analysed in chapter 3 for which in the present Chapter only results are given and compared to other methodologies.

5.2.2 Methods based on linear and non-linear relationship search

The method described in this subsection is based on the application of different ideas previously explained in Chapters 3 and 4. In fact, applying in order the method described in Subsections 3.4.1, followed by variants of those methods presented in Sections 4.4 and 4.5, leads to a methodology that allows obtaining a FIR model of a large-scale system, whereby the algorithm presented here keeps the computational burden within acceptable bounds. The method is applied to the garbage incinerator system detailed in Appendix I.3. The reason for using this example rather than follow the study with the boiler system was that the latter only has 9 variables whereas the former has 20 physical variables. This makes the garbage incinerator system a better academic example so as to demonstrate the applicability of the proposed technique. The research presented in this section was first published in [Mirats and Verde, 2000].

The algorithms of Sections 4.4 and 4.5 are slightly modified here due to the radically different goals of Chapters 4 and 5. The purpose of Sections 4.4 and 4.5 was the design of an alternative method to FRA for identifying the set of important binary relations governing the system under investigation, with the aim of deriving, in a more economical fashion, the topological structure underlying the system under study. The purpose of the present subsection is to design an algorithm that can support FIR in determining a suboptimal mask of high quality for predicting future values of the output variable. As almost all FIR models, this suboptimal mask will usually contain no more than 5 to 7 input variables, i.e., there is no need or even desire to include all remaining variables in the model, as this was the case in Chapter 4.

5.2.2.1 Subsystem decomposition algorithm

The first step of the developed method is to perform a rough variable selection via correlation analysis of the input data. As is well known, correlation as well as covariance measures provide measures of linear association, i.e., association along a line. When using such indices to study possible relations among variables, the modeller must take into account that possible non-linear relations may exist that are not revealed by these descriptive statistics. These non-linear relations are investigated in a posterior step of the methodology.

As discussed already in Chapter 3 and confirmed in Subsection 5.2.1, it is impossible to develop linear statistical techniques that can decide on their own, which variables to retain for a posterior analysis. Every one of the successful techniques did just the opposite. They all concluded, which variables can safely be eliminated from the posterior analysis. The reason for this is simple. A linear statistical technique by itself cannot know whether or not important non-linear relations exist, and therefore, cannot decide that a variable exhibiting negligible linear correlation with the other variables can be discarded. All it can do is eliminate input variables that are strongly linearly correlated with other input variables, since they evidently contain primarily redundant information.

Hence, the first step of the proposed method is based on the simple idea to identify clusters of input variables that roughly contain the same information about the system and drop all of them except one out of the subsequent analysis so simplifying a posterior model computation. One such technique was explained in Subsection 3.4.1: a correlation analysis of the input data is performed, obtaining the data correlation matrix, so variables with a strong linear relationship can be detected. Afterwards, those input variables with an absolute linear correlation coefficient equal or higher than an absolute upper limit r_0 are identified. The input variables in these groups contain almost the same information about the studied process and hence, one of them may be chosen to represent this information. The criterion used to select this variable, as described in Subsection 3.4.1, is to maximise the variation coefficient. This index is a normalisation of each variance variable using its mean as normalisation factor.

Then, as explained in Section 4.4, the previously found correlation matrix can be reduced by eliminating those of its rows and columns that correspond to already eliminated variables. The reduced correlation matrix is then enlarged by adding one row and one column to it representing the output variable. A singular value decomposition of the so modified correlation matrix is then performed, so its eigenvalues and eigenvectors are obtained. Subsequently, the (orthogonal) eigenvectors are projected onto the principal axes. Theoretically, the projections onto all of the subspaces spanned by each pair of eigenvectors have to be examined, but in practice, it suffices to take into account only those projections that contribute to the system variance the most. For each projection, the axes are divided into sectors of 30º, and the variables with larger projection in the obtained sections are joined into the same subsets. In this way, subgroups of linearly related variables are obtained. The algorithm of Section 4.4 is more sensitive than that of Subsection 3.4.1, so that new linearly related groups of input variables are discovered in spite of the fact that the most strongly related input variables had already been dropped. Furthermore, clusters involving the output variable are now found as well.

The algorithm of Section 4.4 is modified in that only the sectors starting at 0º and multiples of ±30º are considered here, accepting the fact that groups formed by variables located in the vicinity of the dividing landmarks may be missed in this way. The reason for this simplification is additional speedup. The simplification is acceptable, because the purpose here is a quick identification of a promising set of variables to be offered as candidate inputs to a subsequent FIR analysis, rather than the desire to identify every one of the important binary relations between variables, as was the case in Chapter 4.

The resulting (slightly smaller) set of subsystems is presented in Table 5-XII.

S1	X_8	X_{14}			
S2	X_4	X_6	X_7	X_{12}	X_{13}
S3	X_2	X_{11}	X_{12}		
S4	X_4	X_6	X_8		
S5	X_{10}	X_{14}			

Table 5-XII Garbage incinerator static subsystem decomposition.

The fact that some groups of variables have been missed reduces the number of important binary relations identified by the algorithm, but does not significantly hamper the ability of FIR to subsequently identify a high-quality model of the output variable.

Yet, as already explained in Section 4.5, no statistical analysis can be successful in identifying a promising set of candidate input variables to FIR that only takes linear relations into account. All that such techniques can accomplish is to exclude redundancy, which unfortunately, will not suffice, as the set of remaining variables would in most cases still be too large in the case of a large-scale system. In Section 4.5, a technique was introduced that enables the modeller to identify non-linear relations among variables using linear statistical approaches. The trick was to apply a non-linear transformation to the non-linear relation that strengthens the linear component of the transformed relation.

The same approach is used here. However, rather than investigating possible non-linear relations between each group and every variable not already belonging to that group, the search shall be limited to investigating potential non-linear relations between the previously found groups on the one hand and variables not belonging to any group on the other. The reason for this simplification is again speedup. The aim here is to ensure that every variable that contains a linear or non-linear relation with any other variable shall be included in at least one group. The purpose is no longer the identification of the complete set of important binary relations. It is sufficient to ensure that promising candidate inputs are dutifully discovered and included in the set of variables offered to FIR to be used in its posterior analysis.

For the garbage incinerator system, the best non-linear transformation, cubic-spline-based, for the variables X_5, X_9, X_{18}, X_{19}, and X_{20}, i.e., those variables not contained in any of the five subsets S_1 - S_5, has been computed. The correlation matrices between each of the transformed variables and the subsystem variables are not reported here. Table 5-XIII, summarises all of these tables and shows whether the transformed variables have been found to have a non-linear relationship with the different subsystems or not.

	S_1	S_2	S_3	S_4	S_5
X_5^*	Yes	-	-	-	-
X_9^*	-	Yes	Yes	-	-
X_{18}^*	-	-	-	-	Yes
X_{19}^*	-	-	-	Yes	-
X_{20}^*	-	-	-	-	-

Table 5-XIII Non-linear correlations between not chosen variables of the incinerator and the five previously identified subsets.

Now, those variables with non-linear correlation with any of the subsets are joined to the corresponding subsets. Table 5-XIV shows the resulting subsystem decomposition of the garbage incinerator process.

S_1	X_5	X_8	X_{14}			
S_2	X_4	X_6	X_7	X_9	X_{12}	X_{13}
S_3	X_2	X_9	X_{11}	X_{12}		
S_4	X_4	X_6	X_8	X_{19}		
S_5	X_{10}	X_{14}	X_{18}			

Table 5-XIV Final subsystem decomposition for the garbage incinerator system including linear as well as non-linear relations between variables.

FIR models of different complexities are computed for every variable contained in at least one subsystem. For each variable, a number of candidate masks is proposed to FIR that contains only those variables as possible inputs that are joined with that variable in one of the groups. For example, since the variable x_8 forms part of two groups, two candidate masks for variable x_8 are proposed containing '-1' elements in the positions corresponding to x_5 and x_{14} on the one hand, and x_4, x_6, and x_{19} on the other. With each of these mask candidate matrices, the optimal mask is computed by FIR. The qualities of these optimal masks are then evaluated, and the best of these models is retained as the proposed suboptimal FIR model for that variable.

For comparison purposes, optimal FIR models for each of these variables are also computed, offering to FIR all variables as potential inputs. To compare the results obtained, the quality of the FIR models is used. A measure of the relative loss of quality of the suboptimal FIR models with respect to the optimal ones is computed.

Figures 5-9 and 5-10 display the percentage of lost quality in each case, computed as

$$\frac{Q_{opt} - Q_{subopt}}{Q_{opt}}$$

where Q_{opt} is the quality of the optimal mask when a full candidate mask is used, and Q_{subopt} is the quality of the optimal mask when a candidate mask with only a subset of potential input variables is used. In these figures, each point is labelled with a variable number, representing the variable that has been modelled from the others. Some of the variables do not appear in the figures, because it is not possible to compute its FIR model for the corresponding complexities (for example, x_5 is only related to x_8 and x_{14}, and therefore, only models of complexities 2 and 3 can be computed).

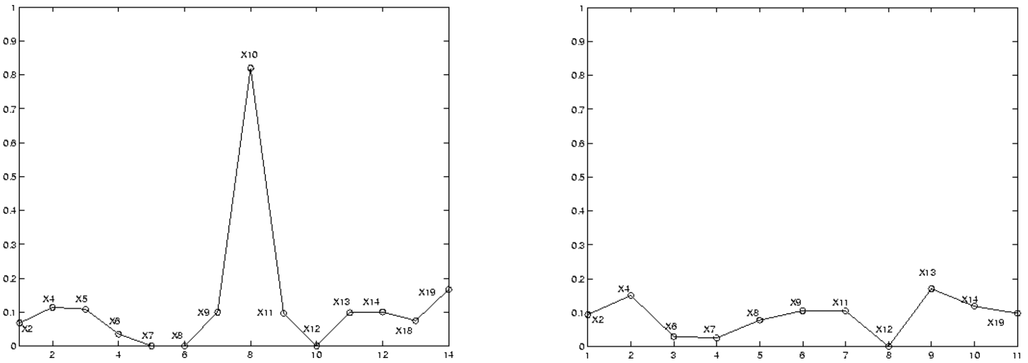


Figure 5-9 Loss of quality for models of complexity 3 (left) and 4 (right), when each of the incinerator system variables is modelled from other variables with which they are related through one or more subsystems.

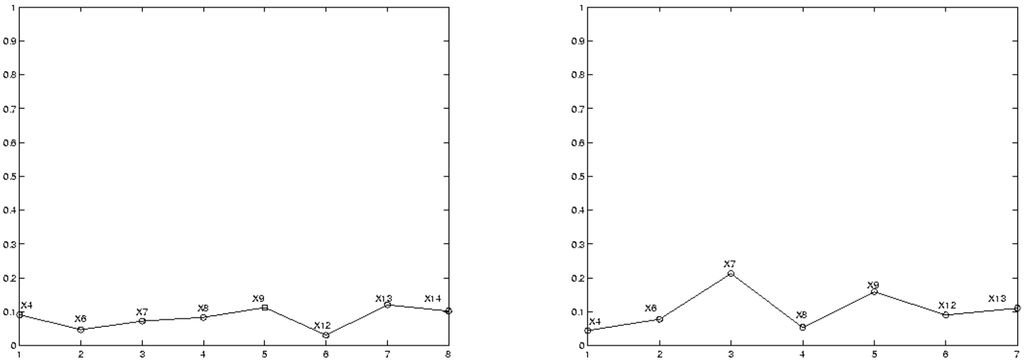


Figure 5-10 Loss of quality for models of complexity 5 (left) and 6 (right), when each of the incinerator system variables is modelled from other variables with which they are related through one or more subsystems.

The loss of quality is almost always below 15%. Only few models exhibit a loss that is slightly greater; for example, the complexity 6 model for x_7 has a loss of the 21.2%, and the complexity 4 model for x_{13} shows a loss of 17%. Only one model exhibits a very high loss: the complexity 3 model for x_{10} with a loss of quality of 82%. These results show that, with the obtained decompositions, there may indeed occur a loss of quality when modelling a system from its subsystems, but as shown later, the computational reduction achieved makes up for the quality lost.

To reinforce the evaluation of the results, another experiment has been performed. The qualities of the FIR models that are obtained using candidate masks with the complementary sets of variables have been computed. The results from this experiment have been compared with those stemming from the use of a full candidate mask in terms of percentage of quality lost.

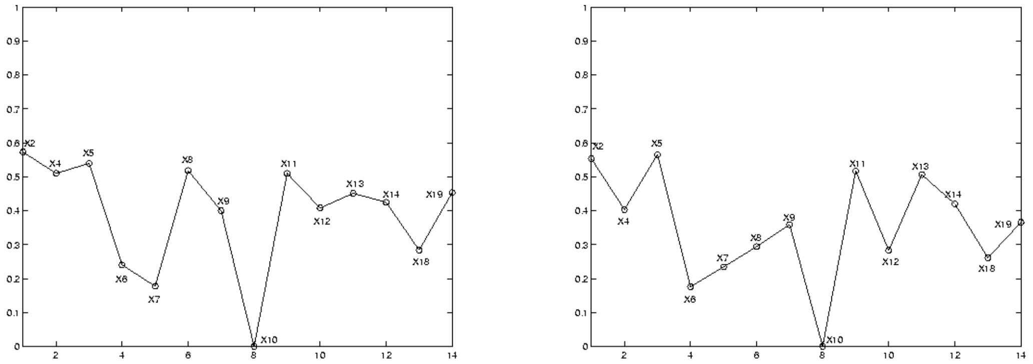


Figure 5-11 Loss of quality for models of complexity 3 (left) and 4 (right), when each of the incinerator system variables is modelled from the complementary sets of variables to those used in Figure 5-9.

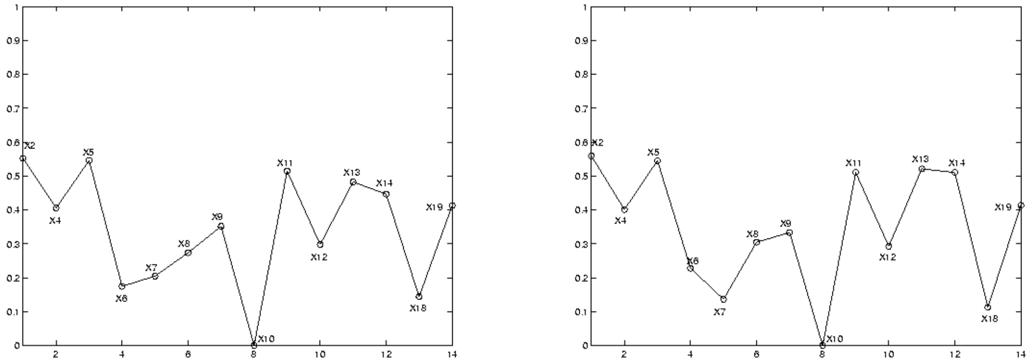


Figure 5-12 Loss of quality for models of complexity 5 (left) and 6 (right), when each of the incinerator system variables is modelled from the complementary sets of variables to those used in Figure 5-10.

Figures 5-11 and 5-12 are labelled as before. This time it is possible to compute FIR models for all considered complexities of all variables. These four figures show that the loss of quality of FIR models, using the complementary sets, is much higher than that obtained in Figures 5-9 and 5-10, with the exception of variable x_{10}, which in this case does not ever exhibit a loss of quality. It is an encouraging result indicating that the information regarding the whole system contained in the obtained subsystems is indeed generally much higher than that contained in any other subgroup of variables.

To offer a more quantitative view of the results obtained, Figure 5-13 shows the loss of quality found in the two performed experiments for FIR model complexities 3 and 4. The dotted line represents the quality lost when using a candidate mask based on the subsystems obtained by the advocated methodology. The continuous line shows the results obtained with the complementary sets of variables.

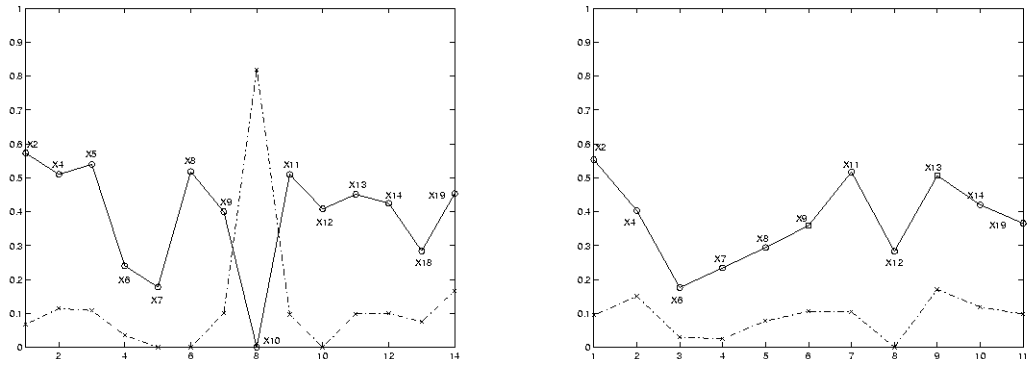


Figure 5-13 Comparative loss of quality, when using the two different presented sets of the incinerator variables for models of complexities 3 and 4.

Now looking at the computational complexity involved in the FIR methodology, it can be seen that a substantial reduction in the model computing time is achieved when the whole system is modelled from the obtained subsystems.

Time dependency of a variable on the past values of the others has not been considered in the proposed methodology for the identification of subsystems, but FIR copes with this issue. Let us suppose that a FIR model of the incinerator process is to be built using exhaustive search. The mask depth is assumed to be 16, and all 20 variables are being considered. Consequently, the mask candidate matrix contains 319 '-1' elements. Table 5-XV lists the number of models (masks) that must be computed for each allowed complexity if an exhaustive search is performed.

Complexity	No. of models to compute
C2	319
C3	50.721
C4	5.359.519
C5	423.402.001

Table 5-XV. Number of FIR models to compute considering a full candidate matrix.

We wish to evaluate the computational complexity of the proposed suboptimal search strategy. The output to be modelled is variable *y*. A problem is encountered, because precisely that variable does not form part of any of the submodels. In order to be able to apply the proposed methodology to the task at hand, the output *y* of the system is added to each one of the found subsystems, listed in Table 5-XII. The cardinality of the subsets will be 4,7,5,5 and 4 respectively. Table 5-XVI gives the number of FIR models that need to be calculated for systems with 4, 5, and 7 variables. It also includes the total sum of models (two times column 1 plus two times column 2 plus column 3) to be computed in this example when the system is modelled from its subsystems. The last column of Table 5-XVI shows the relative computational reduction achieved with the proposed method.

If column 2 of Table 5-XV is named *NMC*, and column 4 of Table 5-XVI is called *T*, the percent of reduction is computed as *100-(T/NMC)*100*. Note, that a consistent reduction in computational time is achieved, except for those few models to be computed with complexity 2.

	4 Var.	5 Var.	7 Var.	Total	% reduction
C2	63	79	111	395	-23.82
C3	1.953	3.081	6.105	16.173	68.11
C4	39.711	79.079	221.815	459.395	91.42
C5	595.665	1.502.501	5.989.005	10.185.337	97.59

Table 5-XVI Reduction of computation effort in the garbage incinerator system, when using the proposed subsystem decomposition method.

5.3 Methodologies leading to a dynamic model

As it was explained in Chapter 3, the term ‘dynamic’ is used in the sense that relations between variables and their delayed versions are considered in the model, i.e., time is explicitly taken into account in the analysis. A static model has been previously defined as one which models the considered output from inputs and other system outputs at zero time delay, i.e., all variables are sampled simultaneously. A dynamic model takes also into consideration past values of all the possible variables involved in the modelling process.

In this section, two different dynamic methods are presented that allow simplifying the mask search space of the FIR methodology. The first technique is an extended version of the subsystem decomposition method discussed in section 5.2.2.1. The second algorithm is based on energy considerations of the available signals or variable trajectories.

5.3.1 Subsystem decomposition method extended with time

The subsystem decomposition method previously discussed in section 5.2.2.1, applied to the garbage incinerator system, can be extended with the explicit inclusion of time in the analysis. The approach taken is the one already outlined in the introduction to this chapter: the raw data model is duplicated and triplicated; each new copy is concatenated to the previous model from the right, after shifting it up by one row; and the so enlarged data model is now considered in a static analysis.

The method will be demonstrated using the same example, i.e., the garbage incinerator system described in Appendix I.3. In accordance with the previous analysis, the considered mask depth is chosen to be 16, i.e., fifteen copies of the original raw data matrix are created and concatenated to the original data matrix from the right.

Hence, the enlarged raw data model is a very big matrix of dimensions 43186 × 320. Thus, a system of 320 variables now needs to be decomposed into subsystems.

The analysis begins by applying the correlation analysis to all the variables of the enlarged garbage incinerator system, corresponding to those variables listed in Table I-III in Appendix I.3 as well as their delayed versions up to $(t\text{-}15\delta t)$. Note that in this first variable selection step, only redundancy among input variables is being searched, as it was done in the case of the static models; thus we should not include in the analysis the considered output variable $y(t)$, but we must consider its delayed versions, from $y(t\text{-}1)$ to $y(t\text{-}15)$ since now, those are considered new possible input variables to model the output. The sample correlation matrix for those 319 input variables is not reported here. The value chosen for the discriminator r_0 is $r_0 = 0.75$, the same as in the previous analysis. In Table 5-XVII, a list of the variables chosen by this first selection step is given. Note that in this first variable selection step, variables x_3, x_{15}, x_{16}, and x_{17}, are already discarded entirely from the set of variables. This reduces enormously the mask search space of FIR, since (20*16)-1 = 319 possible relations among variables (number of '-1' elements of a candidate mask of a 20-variable system of depth 16), are reduced to 319-(4*20) = 239 allowed relations among the remaining variables. Yet, the actual savings are even larger, as shown in Table 5-XVII.

Selected variables
$x_1(t)$, $x_1(t\text{-}9)$
$x_2(t)$
$x_4(t\text{-}\delta t)$
$x_5(t\text{-}14\delta t)$
$x_6(t\text{-}14\delta t)$
$x_7(t\text{-}14\delta t)$
$x_8(t)$
$x_9(t\text{-}14\delta t)$
$x_{10}(t\text{-}14\delta t)$
$x_{11}(t\text{-}2\delta t)$
$x_{12}(t)$
$x_{13}(t\text{-}13\delta t)$
$x_{14}(t)$
$x_{18}(t)$, $x_{18}(t\text{-}\delta t)$, $x_{18}(t\text{-}2\delta t)$, $x_{18}(t\text{-}3\delta t)$, $x_{18}(t\text{-}4\delta t)$, $x_{18}(t\text{-}5\delta t)$, $x_{18}(t\text{-}7\delta t)$, $x_{18}(t\text{-}8\delta t)$, $x_{18}(t\text{-}10\delta t)$, $x_{18}(t\text{-}11\delta t)$, $x_{18}(t\text{-}13\delta t)$, $x_{18}(t\text{-}14\delta t)$
$x_{19}(t\text{-}14\delta t)$
$y(t\text{-}\delta t)$, $y(t\text{-}4\delta t)$, $y(t\text{-}10\delta t)$

Table 5-XVII Garbage incinerator system selected variables, when including time in the subsystem decomposition method.

Therefore, in this very crude first variable selection step, plenty of redundant input variables are already dropped out of the posterior modelling process. Only 30 of the 319 possible input variables are retained. Now with the remaining input variables together with the output, that is a total of 31 variables, subsets (subsystems) of variables are formed that are linearly correlated among each other. The process used to this aim is based on the first steps of a principal component analysis.

The correlation matrix of the remaining variables together with the output can be derived from the entire correlation matrix (with dimensions 320 × 320) previously computed (although for the first variable selection step it was used without the column and row corresponding to the $y(t)$ variable) by eliminating each row and column that reflect a dropped variable. In this way, a new $n \times n$ matrix, $n = 31$, of correlation coefficients is obtained. Then, a singular value decomposition of this matrix is performed, and the obtained eigenvectors are projected onto the principal axes.

In the case at hand, information from only two projections was necessary to derive the groups of linearly related variables. The used projections were first versus second, and first versus third axes. Other projections, such as first versus fourth, second versus third, and second versus fourth axes, were analysed, yet the subsystems derived from these projections were already included in the two main projections. Table 5-XVIII lists the obtained subsystems that only include linear relations among variables.

	Formed subsystems							
S1	x_{18}(t-5)	x_{18}(t-7)	x_{18}(t-8)	x_{18}(t-10)	y(t-10)			
S2	x_4(t-1)	x_6(t-14)						
S3	x_{18}(t-2)	x_{18}(t-3)	x_{18}(t-4)					
S4	x_1	x_2	x_6(t-14)	x_7(t-14)	x_8	x_{11}(t-2)	x_{12}	x_{14}
S5	x_{18}(t-11)	y(t-10)						

Table 5-XVIII Formed subsystems for the garbage incinerator when only linear relations are considered and time is included.

Therefore at this stage of the subsystem decomposition method, we already have identified 5 subsystems with varying numbers of variables. It may be noted that not all of the variables under consideration have been included in one or more of the obtained subsystems. Concretely, variables $x_1(t-9\delta t)$, $x_5(t-14\delta t)$, $x_9(t-14\delta t)$, $x_{10}(t-14\delta t)$, $x_{13}(t-13\delta t)$, $x_{18}(t)$, $x_{18}(t-\delta t)$, $x_{18}(t-14\delta t)$, $x_{19}(t-14\delta t)$, $y(t)$, $y(t-\delta t)$ and $y(t-4\delta t)$ do not form part of any subsystem. This is to be expected, because up to now, only linear relations have been investigated among the system variables to group them into possible subsystems[5].

Now, for those 12 variables that were not included in any of the 5 subsystems, possible non-linear relations between them and the formed subsets should be considered. The previously explained algorithm of Section 4.5 is used for this task.

After computing all the necessary linear combinations, splines and correlations among variables, the subsystem decomposition of the garbage incinerator system remains as listed in Table 5-XIX. In this table, the variables that joined the subsystems in the final non-linear step are marked in bold. Eleven of the twelve previously excluded variables joined the subsystems at this time. Only $x_{13}(t-13\delta t)$ is permanently removed from the list of remaining variables.

[5] Of course, the formed subsystems may also contain non-linear relations among the included variables.

	Final obtained subsystems											
S1	**x_{10}(t-14)**	x_{18}(t-5)	x_{18}(t-7)	x_{18}(t-8)	x_{18}(t-10)	**x_{18}(t-14)**	**x_{19}(t-14)**	y(t-10)				
S2	x_4(t-1)	x_6(t-14)	**x_{18}(t-13)**	**y(t)**	**y(t-1)**	**y(t-4)**						
S3	**x_5(t-14)**	**x_9(t-14)**	**x_{10}(t-14)**	**x_{18}(t-1)**	x_{18}(t-2)	x_{18}(t-3)	x_{18}(t-4)	**x_{18}(t-14)**	**x_{19}(t-14)**			
S4	x_1	**x_1(t-9)**	x_2	x_6(t-14)	x_7(t-14)	x_8	x_{11}(t-2)	x_{12}	x_{14}	**y(t)**	**y(t-1)**	**y(t-4)**
S5	**x_5(t-14)**	**x_{18}**	**x_{18}(t-1)**	x_{18}(t-11)	y(t-10)							

Table 5-XIX Final subsystem decomposition when time is included for the garbage incinerator system.

Note that two of the final obtained subsystems include the output variable at time *t* within them, namely subsystems S2 and S4. If none of the subsystems would contain *y(t)*, this variable would have to be added to all of the subsystems. Now, those subsystems that include the output variable at time *t* are used to postulate candidate masks to FIR and hence serve to derive models of the considered output[6]. One of the subsystems, S2, is by itself a complexity six model of the output:

$$y_{S2\_C6}(t) = f\{\ x_4(t-1),\ x_6(t-14),\ x_{18}(t-13),\ y(t-1),\ y(t-4)\ \}$$

The quality of this model is found to be $Q = 0.4153$.

As we have been using FIR models of complexity 5 all throughout the dissertation, FIR is presented with the candidate matrix that can be derived from the S2 subsystem:

$$mcan_{S2} = \begin{array}{c} \begin{array}{cccc} x_4 & x_6 & x_{18} & y \end{array} \\ \begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

The number of models to compute are 5 of complexity 2, 10 of complexities 3 and 4, and 5 of complexity 5, a total 30 models, a very low number if one takes into consideration the number of models we have to compute with FIR in a typical application.

[6] In fact, subsystems containing the output variable at time *t*, i.e., containing *y(t)*, could be considered by themselves models of the output variable. In section 5.2.2.1 it was shown that when a variable is FIR-modelled from other variables with which it is related in the found subsystems, the loss of quality with respect to those FIR models derived from all the variables is small. Here, the criterion of proposing to FIR candidate masks extracted from those subsystems, has been adopted in order to obtain suboptimal FIR models and then compare the results with the rest of the models derived in the dissertation.

When the optimal mask search algorithm is used with this candidate mask, the following optimal models are obtained, one for each of the allowed complexities, together with their corresponding qualities:

$$\begin{array}{ll}
y_{S2\_C2}(t) = f\{\, y(t\text{-}1) \,\} & Q = 0.6137 \\
y_{S2\_C3}(t) = f\{y(t\text{-}1),\ x_4(t\text{-}1) \,\} & Q = 0.5834 \\
y_{S2\_C4}(t) = f\{\, y(t\text{-}1),\ x_4(t\text{-}1),\ x_6(t\text{-}14) \,\} & Q = 0.5781 \\
y_{S2\_C5}(t) = f\,\{\, y(t\text{-}1),\ x_4(t\text{-}1),\ x_6(t\text{-}14),\ x_{18}(t\text{-}13\delta t) \,\} & Q = 0.5775
\end{array}$$

For the other subsystem, S4, we find that it contains 12 variables, which is clearly not optimal in terms of FIR. Accordingly, the following candidate mask is proposed to FIR:

$$mcan_{S4} = \begin{array}{c} \begin{array}{ccccccccc} x_1 & x_2 & x_6 & x_7 & x_8 & x_{11} & x_{12} & x_{14} & y \end{array} \\ \begin{pmatrix}
0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
-1 & -1 & 0 & 0 & -1 & 0 & -1 & -1 & 1
\end{pmatrix} \end{array}$$

In this case, 11 models of complexity 2, 55 of complexity 3, 165 of complexity 4, and 330 of complexity 5, a total of 561 models, must be computed, also a very low number of models with FIR applications. Hence, when the optimal mask search algorithm is used with this candidate mask, the following optimal models are obtained, one for each of the allowed complexities, together with their corresponding qualities:

$$\begin{array}{ll}
y_{S4\_C2}(t) = f\{\, y(t\text{-}1) \,\} & Q = 0.6137 \\
y_{S4\_C3}(t) = f\{y(t\text{-}1),\ x_2(t) \,\} & Q = 0.5983 \\
y_{S4\_C4}(t) = f\{\, y(t\text{-}1),\ x_1(t\text{-}9\delta t),\ x_2(t) \,\} & Q = 0.5809 \\
y_{S4\_C5}(t) = f\,\{\, y(t\text{-}1),\ x_1(t\text{-}9\delta t),\ x_2(t),\ x_7(t\text{-}14\delta t) \,\} & Q = 0.5897
\end{array}$$

The result of simulating the output variable with the complexity 5 $y_{S2}(t)$ model is presented in Figure 5-14. As was done with all other simulations, 85% of the available data were used to derive the model, whereas the remaining 15% were considered to be the validation data set, in this case 36.720 and 6.480 data points respectively. Figure 5-14 shows the last 500 points of the simulated output. In this figure, the real output trajectory is depicted as a continuous line, whereas the simulated output is plotted as a dotted line. It can be appreciated that the simulated output follows the real trajectory rather accurately. The reader may appreciate the quality of predictions achievable by a carefully chosen suboptimal dynamic FIR model. It should be noted that this model is capable of following the spurious changes on the NOx gas emission present in the data. The MSE error when using this model is 0.1135.

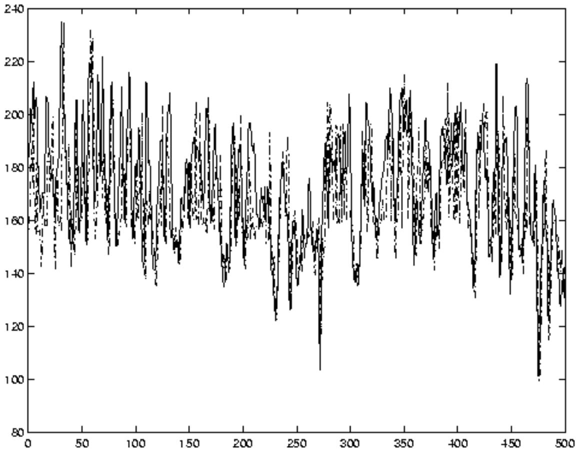


Figure 5-14 Garbage incinerator output variable simulated with the complexity 5 model derived from S2.

There is another simulation that ought to be evaluated: the one obtained using the corresponding model of equal complexity derived from the S4 subsystem. Figure 5-15 shows it. Also these simulation results are excellent, with a slight increase of quality. The MSE for this model is 0.1108.

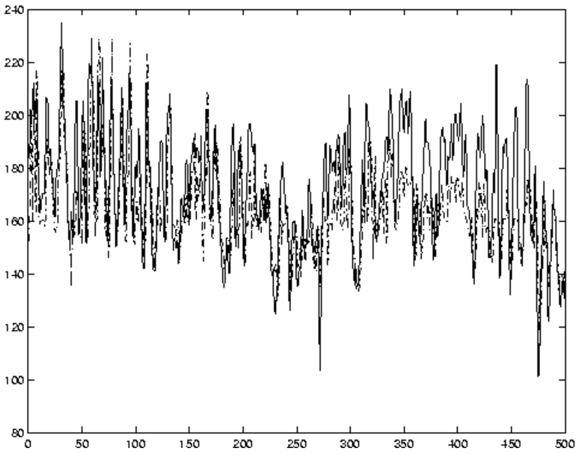
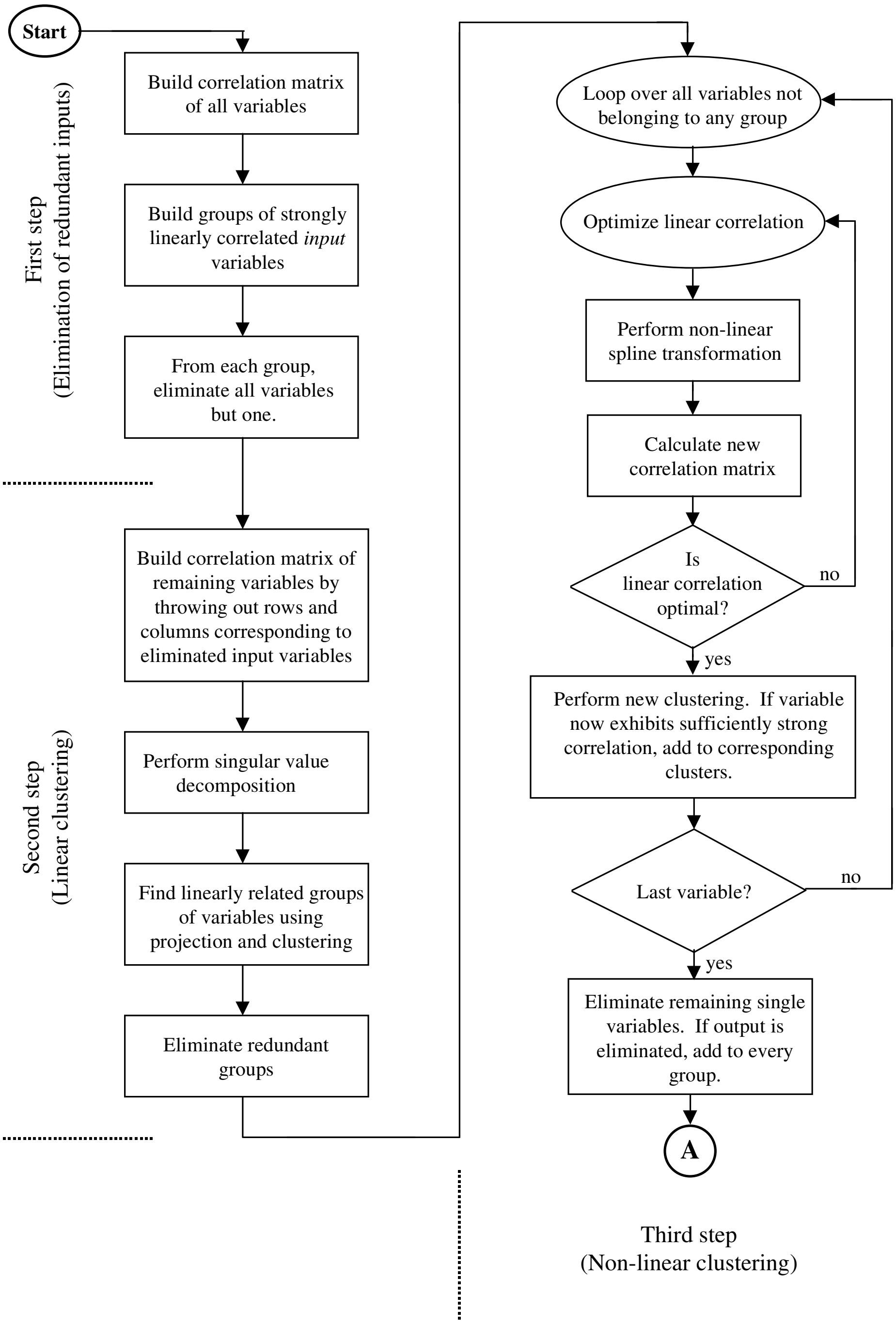


Figure 5-15 Garbage incinerator output variable simulated with the complexity 5 system derived from subsystem S4.

As can be seen from the presented figures, the results obtained with the proposed algorithm are quite satisfactory. Note that probably none of the obtained models is truly optimal. We cannot be sure because solving the exhaustive search problem of finding the true optimal mask of a system with 319 '-1' elements is by far outside the capabilities of our current day computers. The importance of optimality in a FIR model is relative; this in turns is logical since we are computing qualitative models. Hence it is possible to obtain a good prediction without using the truly optimal model, which in a FIR sense, is defined by the mask of highest quality.

The subsystem decomposition method extended with the inclusion of time allows obtaining a decent FIR model of a large-scale system acceptably fast, i.e., while keeping the computational burden within acceptable limits. Finally, in this section, a flowchart is added so as to summarize the four principal stages of the presented method.


Start
First step
(Elimination of redundant inputs)
Build correlation matrix of all variables
Build groups of strongly linearly correlated *input* variables
From each group, eliminate all variables but one.
Second step
(Linear clustering)
Build correlation matrix of remaining variables by throwing out rows and columns corresponding to eliminated input variables
Perform singular value decomposition
Find linearly related groups of variables using projection and clustering
Eliminate redundant groups
Loop over all variables not belonging to any group
Optimize linear correlation
Perform non-linear spline transformation
Calculate new correlation matrix
Is linear correlation optimal?
no
yes
Perform new clustering. If variable now exhibits sufficiently strong correlation, add to corresponding clusters.
Last variable?
no
yes
Eliminate remaining single variables. If output is eliminated, add to every group.
A
Third step
(Non-linear clustering)

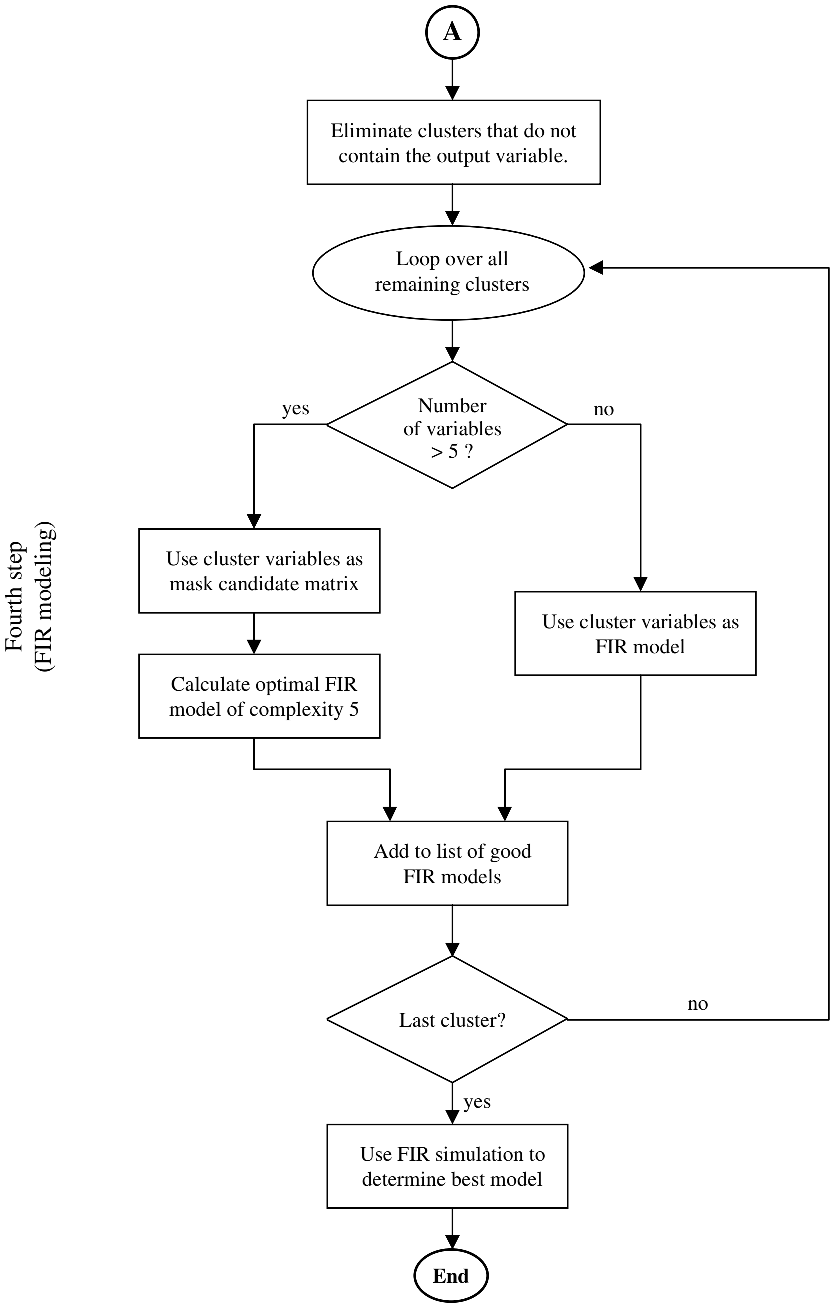

A
Eliminate clusters that do not contain the output variable.
Loop over all remaining clusters
Number of variables > 5 ?
yes
no
Use cluster variables as mask candidate matrix
Calculate optimal FIR model of complexity 5
Use cluster variables as FIR model
Add to list of good FIR models
Last cluster?
no
yes
Use FIR simulation to determine best model
End
Fourth step (FIR modeling)

Figure 5-16 Flowchart summarizing the subgroup decomposition method.

Four main steps form the advocated method. In the first variable reduction step, the output is not considered. Input variables that, in their recorded history, contain very similar information, i.e., can be considered redundant, are grouped together. At the end of this step, the following variables survive:

- one from each of the formed groups,
- all variables not belonging to any group, and
- the output.

In the second linear clustering step, the output is considered and groups of linearly related variables are formed. A different clustering algorithm is used than in step one, an algorithm that is more sensitive than the one previously used, and in this way, new groups of related variables are formed. No variables are eliminated in this second step.

In the third non-linear clustering step, the variables not forming part of any group yet may join groups. To this end, the algorithm used here investigates possible non-linear relations between each one of these variables separately and any one of the clusters formed. Those variables that do not join any groups are eliminated now, except for the output, which may never be eliminated.

In the fourth FIR-modelling step, those groups not containing the output are eliminated first. Groups containing less than 6 variables lead directly to a FIR model. On the other hand, for groups of 6 or more variables, a candidate mask is proposed to FIR using as *m*-inputs and *m*-output the variables of those groups.

By using this method, a FIR-model of a complex system can be obtained in a reasonable amount of time.

5.3.2 Delay estimation using energy information

Up to now, each variable trajectory has been analysed from a deterministic point of view, i.e., at each time interval, each measured value has been considered to match the real value of the observed physical variable. At this point, a more realistic interpretation of the data vector should be introduced.

Each variable trajectory can be seen as the collection of values measuring a desired physical characteristic, such as the fuel flow through a pipe, plus an added noise, representing e.g. measurement noise or thermal noise. So each of the obtained trajectories can be interpreted as a realisation of a stochastic process, i.e., their exists a deterministic as well as a random part in the available trajectories. With this interpretation, the energy of the signals can be computed and used to determine, at which delays each input variable has more energy related to the output, i.e., obtaining the most probable delays for which the input variables are to appear in a FIR qualitative model.

Considering data episodes as stochastic signals is, of course, not a new concept in the engineering world, but it is in the mark of the FIR methodology. The concept of obtaining those input variable delays that are more *energetically* related to the output variable is different from other approaches previously taken in the context of FIR-related research.

In the open literature, delay estimation is usually understood to mean the time delay between an emitted signal and a delayed (possibly distorted) version of the same signal when it is being received. A number of approaches have been treated in the literature that relate to the processing of radar or acoustics signals. A first group of techniques are those based on correlation or spectral analysis. These approaches have been broadly used in acoustics and involve calculating the correlation between the signals before and after the time delay. A comparison of different delay estimation correlation-based methods is offered in [Fertner and Sjölund, 1986]. In [Jacovitti and Scarano, 1993], the direct cross-correlation method is analysed and compared to other delay estimators when two differently positioned sensors receive the same signal at different time instants:

$$x_1(t) = s(t) + n_1(t)$$
$$x_2(t) = A\cdot s(t-D) + n_2(t)$$

where $n_1(t)$ and $n_2(t)$ are different random processes representing the measurement noise of the two sensors. In [Marple, 1999], cross-correlation algorithms are provided for estimating the group and phase delay between two finite *N*-point real-valued signals. A different perspective is given in [Readle and Henry, 1994], where a high-order recursive least squares estimator together with a fuzzy-logic-based reasoning engine are used to estimate the time delay. In [Ettaleb, *et al.*, 1998], an off-line extended least squares method is used for time delay estimation purposes. In [Händel, 1999], a frequency-selective algorithm is applied to estimate the time delay between a signal $x_1(k)$ and its delayed version $x_2(k-D)$.

All these techniques compute a unique, real- or integer-valued, time delay and need the original as well as the delayed versions of the signal to estimate it. In this dissertation, the concept of delay estimation is different. It is desired to identify those (possibly multiple) time delays, for which each input variable is most strongly related to the selected output. This information can then be used to propose a sparse mask candidate matrix, so leading to a substantial FIR model search-space reduction. There are two main differences between the previously described techniques and the one developed here: more than one time delay may be considered, and there are no original and delayed versions of the same signal, but rather input/output trajectories of a system.

A brief review of stochastic signal theory is offered previous to explaining the developed approach. The notation employed here is that used in [Kalouptsidis, 1997].

Stochastic signals provide a successful approach for modelling those signals that cannot be completely determined. For example when transmitting a signal several times by the same source through a communication channel, the received signal is not always exactly the same. Measuring the same signal under the same environmental conditions yields slightly different waveforms due to random noise in the system. The value $x(n)$ of a stochastic or random discrete signal at time instant *n* can be expressed as a random variable $x(n) : \Omega \rightarrow R$, $\varsigma \rightarrow x(n)\ (\varsigma)$. So it depends on the outcome ς of a probabilistic experiment specified by the space Ω, the set of events, and a probability law *P*. The behaviour of the signal at time *n* is given by the distribution function of the random variable $x(n)$,

$$F_n(x) = P\{\varsigma \in \Omega : x(n)\ (\varsigma) \leq x\} = P[x(n) \leq x],\ \ x \in R$$

where *x* is the independent variable of the distribution function and should not be confused with the signal *x*. If the distribution function is differentiable with respect to *x*, the behaviour of the signal at time *n* is determined by the probability density function

$$f_n(x) = \frac{dF_n(x)}{dx}$$

This function, $f_n(x)$, expresses the probability that the value of the signal at time n lies inside an infinitesimal interval given by

$$f_n(x)dx = P[x \leq x(n) \leq x + dx]$$

At this point an important question arises: Which is the necessary information to fully describe a stochastic process? It can be thought that the family of density functions for each $n \in [0, N-1]$ is enough, but this family does not suffice to characterise the statistical evolution of the signal. Consider, for example, that the signal at time n_2 is to be predicted from information gathered at a past time instant n_1. In such application, the conditional density function $f_{n1|n2}(x_1,x_2)$ would be involved, which in turn relates to the second order density $f_{n1,n2}(x_1,x_2)$. In a more general case when trying to infer conclusions on the behaviour of the signal at time n_k from information gathered at times n_1, n_2, ..., n_{k-1}, the family of k^{th} order distribution functions is necessary:

$$F_{n1,\, n2,\, ...,\, nk-1}(x_1,\, x_2,\, ...,\, x_k) = P[x(n_1) \leq x_1,\, x(n_2) \leq x_2,\, ...,\, x(n_k) \leq n_k]$$

It can be shown that the family of distributions

$$\{F_{n1,\, n2,\, ...,\, nk-1}(x_1,\, x_2,\, ...,\, x_k) : k \in Z^+,\, n_1,\, n_2,\, ...,\, n_{k-1} \in Z\}$$

characterises the probabilistic behaviour of the signal $x(n)$. A deeper mathematical development for stochastic processes is offered in [Papoulis, 1991; Wong, 1971; Jazwinski, 1970].

In practice, however, determination of the above families is an unmanageable task. Hence alternative considerations should be found that involve only a few parameters of the mentioned family of functions. To this end, the stochastic signals to be considered need to be limited in appropriate ways. We shall now briefly discuss the class of stochastic signals referred to as stationary signals.

A random signal $x(n)$, $n \in Z$, is called stationary, if for every integer I, the $x(n + I)$ shifted signal has the same family of distributions, meaning that every distribution remains invariant under a time shift. For purely random processes, it is a necessary as well as sufficient condition for stationarity that first-order densities are identical, i.e., the mean $m(n)$ and all moments $m_k(n)$ are constant, independent of n.

$$m(n) = E[x(n)] = \int_{-\infty}^{\infty} x f_n(x)\,dx$$
$$m_k(n) = E[x^k(n)] = \int_{-\infty}^{\infty} x^k f_n(x)\,dx$$

This is insufficient for processes with a non-random component, for which, to ensure strict stationarity, shift invariance of the entire family of densities must be accomplished. Yet, as this property is hard to establish in practice, the class of considered stochastic processes is widened to include so-called quasi-stationary signals. To this end, the concept of a *Wide Sense Stationary Process (WSSP)* is defined. A random process is said to be a WSSP when the family of second-order densities depends on the time difference n_1 - n_2

only. Thus a signal *x(n)* is called wide sense stationary, if its mean is constant and its autocorrelation function, *R(n1,n2)*, depends only on the difference |n_1 - n_2|.

$$R(n_1, n_2) = E[x(n_1)\,x(n_2)]$$

Hence, for a WSSP, the following expressions hold:

$$R(n_1, n_2) = R(n_1\text{-}n_2\text{+}n_2,\ 0\text{+}n_2) = R(n_1\text{-}n_2,0) = r(n_1\text{-}n_2)$$
$$r(n) = E[x(n+k)\ x(k)]$$

For this concrete class of stochastic processes, it is possible to extend the harmonic analysis of deterministic signals so as to perform a spectral analysis. The Fourier transform of the autocorrelation sequence *r(n)*

$$S_x(w) = \sum_{n=-\infty}^{\infty} r(n)\,e^{-jwn}$$

is the so-called power spectral density of the signal. *S(w)* is a 2π-periodic function. The autocorrelation can be recovered by means of the inverse Fourier transform of *S(w)*.

In the ongoing research, there only is information available about the trajectories of the measured variables. So rather than computing the second-order statistics of the process and then its spectral density from its probability density function, it is necessary to proceed directly from the observed data. An obvious approach is to first estimate the autocorrelation *r(n)* from the available data, and then use the discrete Fourier transform (DFT) to compute the power spectral density *S(w)*. Two possible estimators of the *r(n)* sequence are

$$\hat{r}(n) = \frac{1}{N}\sum_{k=0}^{N-|n|-1} x(k)\,x(k+|n|), \qquad |n| \le N-1$$
$$\hat{r}_u(n) = \frac{1}{N-|n|}\sum_{k=0}^{N-|n|-1} x(k)\,x(k+|n|), \qquad |n| \le N-1$$

The first of these two estimators is biased, whereas the second one is an unbiased estimator of the autocorrelation *r(n)*. Once the *r(n)* sequence has been determined, the power spectral density *S(w)* can be estimated by means of the FFT (Fast Fourier Transform) algorithm. This approach is usually called indirect, because it computes the autocorrelation prior to computing the power spectral density. Another possibility to determine *S(w)* is to use the so-called direct approach, which computes the power spectral density directly from the available data:

$$S(w) = \frac{1}{N}|X(w)|^2 \qquad X(w) = \sum_{n=0}^{N-1} x(n)\,w(n)\,e^{-jwn}$$

where *w(n)* is a window in the time domain, normally real and even, that minimises the effects of the finite data length. This estimator is also called periodogram, or modified periodogram when windowing is used. A detailed description of these and other methods to estimate the power spectral density, as well as their implementation in the Matlab software package can be found in [Stearns and David, 1996; Marple, 1987].

Another comment should be made about the mathematical properties of stationary stochastic processes. Each time the researcher gathers data from the system, the variable trajectories have slightly different waveforms due to the random noise associated with the signal and the measurement process. One interesting property would be to ensure that the statistics of the stochastic process are invariant with respect to their realisation; this property is called ergodicity. Strictly, a stationary process is said to be ergodic if every invariant random variable of the process is almost surely equal to a constant. This would guarantee that it is possible to derive a valid model of the system from a single data set of the input/output variables forming the system, supposing that the data are gathered in the normal operational mode of the plant to be modelled.

The interest of this section is, given a matrix of discrete real-valued trajectories of a MISO system, to obtain the time delays, for which each input variable is most strongly, in energy terms, related to the selected output variable. The computation of those delays is based on the determination of the power spectral densities of the signals. In order to compute the signal power spectra, it is necessary to work with stationary or quasi-stationary stochastic processes. Moreover, if only one realisation of the process is available, ergodicity is also required.

The data sets analysed in this research effort stem from large-scale industrial systems. In this kind of systems, supposing a normal operation status of the plant, the involved variables lay inside a given margin of values. Hence it is reasonable to interpret their trajectories as realisations of quasi-stationary stochastic processes when the observation time is significantly longer than the largest time constant of the system. Ergodicity is also a reasonable hypothesis. Suppose for example, that a model is to be constructed of a system from which data have been gathered during an entire hour. If the system is in its normal operational mode and supposing that a time span of an hour suffices to cover the system dynamics, the resulting model should be identical to the one obtained if the data had been gathered during the following hour. The information that the variables contain about the system under study is the same, although the shape of the trajectories may be slightly different due to noise. Of course, this implies that the signal-to-noise ratio (SNR) of the signals to be measured is sufficiently high, but this depends on the system design and the method used to gather data from the system, both beyond the scope of this work.

So under the hypothesis of quasi-stationarity and ergodicity of the available signal trajectories, a power-spectral-density-based method for estimating the delays for which the (measured) input variables are most strongly related to the (measured) output is presented. This approach is directly based on computing the so-called *spectral coherence function*. Using this method, only delays from 2 to ∞ can be computed. No information can be obtained in this way about delays 0 and 1. Consequently, the method does not provide any information about the bottom two rows of the FIR mask candidate matrix. There are two possibilities to deal with this problem. The first and most obvious solution is to fill those two rows entirely with -1 elements. The second solution is to use another technique, such as the methodology proposed in Section 2.4.1, to obtain a depth-two candidate mask and then combine it with the results obtained here.

The proposed algorithm finds a single sparsely populated mask candidate matrix by computing the delays of each variable that are most relevant to the considered output. This is accomplished by means of a spectral analysis performed on the observed trajectories. To this end, Matlab's *spectrum* function is applied separately to each input/output pair after detrending all variables individually. The spectrum function returns the frequency range, F, over which the power spectra are being sampled. Assuming the time signals were sampled at time intervals of length t_s, the spectra then range from $-f_s/2$ until $+f_s/2$, where $f_s = 1/t_s$. The spectrum lines are equidistantly spaced over the spectral range. The number of spectral lines, n_s, to be chosen is an input parameter to the *spectrum* function. The output parameter F is a vector of length n_s providing the frequency values of the spectral lines.

The system has n observed variables, one of which is the selected output, the remaining $(n\text{-}1)$ variables are potential inputs. The input variables are called x_1, x_2, …, x_{n-1}, and the output variable is called y. The *spectrum* function, applied separately to each pair $<x_i,y>$, returns, beside from the frequency vector, F, the following power spectra: P_{xxi}, the power spectral density of the input variable x_i; P_{yy}, the power spectral density of the output variable y; and P_{xyi}, the cross-spectral density of the input/output pair. It also returns the *coherence function*, C_{xyi}, which is defined as follows [Marple, 1987]:

$$C_{xyi}(f_i) = \frac{\left|P_{xyi}(f_i)\right|^2}{P_{xxi}(f_i)P_{yy}(f_i)}$$

where f_i are the frequency values over which the spectra were sampled. The C_{xyi} values are positive real values in the range [0,1]. They are relative measures of the cross-energy density that exists between the input variable x_i and the output variable y at different frequencies f_i.

After applying the algorithm, $(n\text{-}1)$ coherence functions have been calculated, one for each of the potential inputs. These coherence functions are now used to identify the significant inputs. A *significant input* exists, where its corresponding coherence function exhibits a *significant peak.* The time delays, t_i, associated with these significant peaks are simply the inverses of the frequency values, f_i, at which these peaks occur.

The following algorithm is proposed to identify the significant peaks and significant inputs of the coherence functions:

1. Each coherence function is detrended separately. Since the coherence functions are quasi-stationary, detrending essentially means removing the mean. Matlab's *detrend* function is used to accomplish the task. After detrending the coherence function, the negative values of the detrended coherence function are set to zero, as those values correspond to the smallest of the peaks.

2. If desired, the same process can be repeated several times. Each time, the smallest of the remaining peaks will be cancelled, and only the larger peaks remain.

3. The *significant peaks* are defined as those peaks that are larger than 2.5 times the standard deviation of the detrended coherence function:

   *sig_peak = F(find(Cxy > 2.5*std(Cxy)))*;

4. The *significant inputs* are those delays that correspond to the significant peaks:

   sig_inp = round(fs ./ sig_peak);

The significant inputs denote the positions within the mask candidate matrix that need to be filled with '-1' elements. Due to the limited resolution of the method, the smallest delay obtainable is 2, i.e., no information can be obtained about delays 0 or 1. The algorithm can be completed by either filling the two bottom rows of the mask candidate matrix with '-1' elements, since the algorithm does not provide any information for that case, or alternatively, by using the first two steps of the algorithm discussed in Section 2.4.1 to obtain the missing information.

This approach has been applied to study the garbage incinerator system reported in Appendix I.3. For incineration systems, turbulent combustion with the interaction of fluid mechanics, chemical kinetics, and heat transfer makes the process highly non-linear, where it is difficult to apply linear statistical techniques to perform variable selection. In the example presented here, 19 input variables and one output variable are taken into account, i.e., a MISO system is considered. It is desired to simplify the system as well as obtain a good predictive model; the system under study has 20 variables, leading to a huge space of possible masks when no criterion is applied to find information about the important delays between variables. The approach can also be extended to MIMO systems analysing each output separately so as to obtain a different FIR model for each of the outputs.

In the stream of the previous experiments, 85% of the gathered data (in this concrete case, 36720 data points) have been used to perform the model search. The sampling period of this data is 1 minute. Table 5-XX shows the most important delays found between the input and output variables when applying the power-spectral-density-based proposed method.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	y
15	10	7	3	20	4	18	3	15	7	8	5	5	3	5	18	18	_	_	24
12	9	6		19	3	17	2	12	6	7	4	4		2	17	7			23
9	6	5		18	2	8		11	3	5		3			4	2			20
4	5	4		12		7		9	2	4		2			3				18
3	4	3		11		5		8		2					2				8
2	3	2		9		3		7											6
	2			7		2		3											5
				3				2											4
				2															3
																			2

Table 5-XX Delays to consider between inputs and output variable for the garbage incinerator system.

With this information, a candidate mask up to depth 25 could be proposed. In order to compare the newly obtained models to those obtained in the previous section, where a subgroup decomposition method has been applied to the garbage incinerator system, a candidate mask of depth 16 is to be proposed to FIR. Since the energy approach does not provide information about delays 0 and 1, the first two rows of the candidate mask should be filled with -1 elements. This would lead to a mask candidate matrix with 113 -1 elements out of 319. Although the simplification achieved is significant, there still are lots of models to compute. Yet, a more economic candidate mask could be proposed for delays 0 and 1 using the information given by a previously presented approach, namely the mask search algorithm discussed in Section 2.4.1 of this dissertation. Using this approach, the following depth-16 candidate mask can be proposed[7]:

$$mcan = \begin{pmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\
-1 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 & 0 & -1 & 0 & 0 & 0 & -1 \\
-1 & -1 & -1 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & -1 \\
-1 & -1 & 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & -1 & -1 \\
-1 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1
\end{pmatrix}$$

Note that, in this candidate mask, variable 18 has already been discarded, i.e., a lateral effect of the given approach is variable selection. The proposed mask has 94 -1 elements, so 94 masks of complexity 2, 4.371 of complexity 3, 134.044 of complexity 4, and 3.049.501 of complexity 5 must be computed. If a full candidate mask had been proposed, the number of masks, models, to compute would have been 319, 50.721, 5.359.519 and 423.402.001 for complexities 2, 3, 4, and 5 respectively, so an important reduction in the number of models to be computed has been achieved. Results of presenting the previous reduced candidate mask to the FIR optimal search engine are given in Table 5-XXI.

In order to better understand this table, we recall the sub-optimal search algorithm discussed in Section 2.4.1. That algorithm was based on proposing to FIR candidate masks of increasing depth. The optimal (exhaustive) mask search algorithm was then employed to evaluate the quality of each mask of complexity c □ m. Masks were grouped in sets of equal complexity. For each complexity, c, the mask of highest quality was found. Its value is Q_{best}. The relative quality of any one of these masks is defined as $Q_{rel} = Q / Q_{best}$. All masks with a relative quality of $Q_{rel} > s$ were considered to be *good masks*[8]. All good masks of a given complexity were then investigated. If a given input was being used by at

[7] Results of applying the algorithm of Section 2.4.1 to the garbage incinerator system are presented in Appendix II.1. The depth–2 mask used here for rows 0 and 1 of the presented candidate mask is borrowed from this Appendix.

[8] In the current implementation of the algorithm, $s = 0.975$.

least *t* % of all good masks of a given complexity, it was considered a *significant input* to the next step of the algorithm.

Here, the same post-analysis that we used in the algorithm described in Section 2.4.1 has been used since, as it has been mentioned all throughout the dissertation, there may be more than one good qualitative model to simulate the system. All of the masks that the FIR optimal search engine computed from the given depth-16 candidate mask were analysed so as to find, which variables were used for those masks with a relative quality higher than 0.975. Table 5-XXI shows the inputs used by the *good masks* for each of the allowed mask complexities. Since every mask includes the output variable, the last row for delay 0 shows how many good masks were found for each of the allowed complexities. Due to space requirements, this table only shows the results for mask complexities 4 and 5 and only lists those delays, from 0 to 15, that were used in the selected models. For those masks with complexities 2 and 3, the results of this simulation are the same as for the simulations presented in Appendix II.1.

	Complex. 4 (Q_{max}= 0.6340) $Q>0.975Q_{max}$ -- 37 masks										Complex. 5 (Q_{max}= 0.6274) $Q>0.975Q_{max}$ -- 180 masks										
Delay	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	12
x_1	—	—	—	—	—	—	—	—	—	—	3	3	3	3	3	—	—	—	—	69	1
x_2	3	2	3	3	3	3	3	0	0	3	3	3	3	3	3	3	3	—	—	102	—
x_6	—	—	—	—	—	—	—	—	—	—	6	6	6	5	7	—	—	—	—	—	—
x_7	1	1	1	1	1	1	1	1	1	1	28	7	7	7	—	7	—	6	6	—	—
x_{12}	—	—	—	—	—	—	—	—	—	—	3	3	—	—	3	3	—	—	—	—	—
x_{13}	—	—	—	—	—	—	—	—	—	—	—	3	4	3	3	4	—	—	—	—	—
x_{19}	—	—	—	—	—	—	—	—	—	—	5	5	—	—	—	—	—	—	—	—	—
y	37	37	4	4	5	13	13	—	12	—	180	180	—	9	19	51	59	—	66	—	—

Table 5-XXI Results obtained when determining the most important delays in energy terms.

Using this approach, good qualitative models are obtained while achieving an important reduction of the overall computational cost. The qualities of the obtained models are very similar to those obtained in the analysis provided in Appendix II.1. In this study, the input variables 1, 2, 6, 7, 12, 13, and 19 are selected to model the output while all other input variables are discarded. In Table 5-XXII, the percentage of computational savings for complexity 5 models is shown, achieved with respect to the case of proposing a full candidate mask to FIR.

	Number of visited models	Number of models to visit using a depth 10 full candidate mask	Percent of computation alleviation
C5	3.049.501	428.812.560	99,29%

Table 5-XXII Reduction on the computational cost for the garbage incinerator system.

Just to see the quality of the obtained models, the result of simulating the output variable with the model of complexity 5 with highest quality is presented in Figure 5-17. This model is given by the equation:

$$y(t) = f\{\ y(t-\delta t),\ y(t-8\delta t),\ x_2(t-9\delta t),\ x_7(t)\ \} \qquad Q = 0.6274$$

As was done with all other simulations, 85% of the available data were used to derive the model, whereas the remaining 15% were considered to be the validation data set, in this case 36.720 and 6.480 data points respectively. Figure 5-17 shows the last 500 points of the simulated output. In this figure, the real output trajectory is depicted as a continuous line, whereas the simulated output is plotted as a dotted line. It can be appreciated that the simulated output follows the real trajectory rather accurately. The MSE error when using this model is 0.1096.

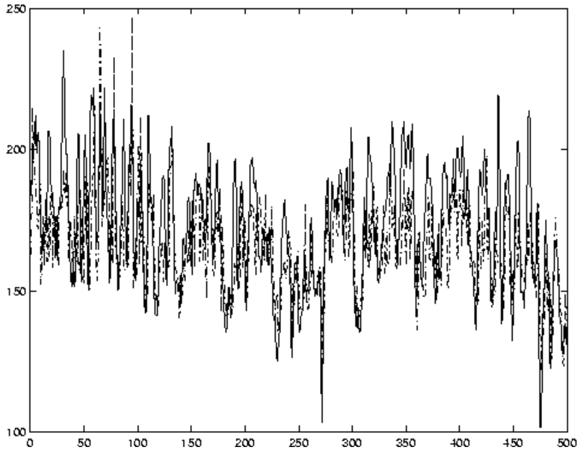


Figure 5-17 Garbage incinerator output variable simulated with the highest quality complexity 5 model when the candidate mask given to FIR is derived from the energy method.

Information about good models in Table 5-XXI was given in order to manifest that a good prediction can be achieved not only with the qualitative model of highest quality, precisely because we are dealing with qualitative models. So another model, the one that used as inputs those four variables that show the highest frequencies among all the complexity-5 masks, is used to simulate the last 500 points of the output trajectory. This model is given by the equation:

$$y(t) = f\{\ y(t-\delta t),\ y(t-8\delta t),\ x_1(t-9\delta t),\ x_2(t-9\delta t)\ \} \qquad Q = 0.6249$$

Note the similarity with the optimal model for this study, in which three of the four input variables of the model coincide. Note also the slightly inferior quality of the non-optimal model. Results of this simulation can be appreciated in Figure 5-18. The MSE error when using this model is 0.1108.

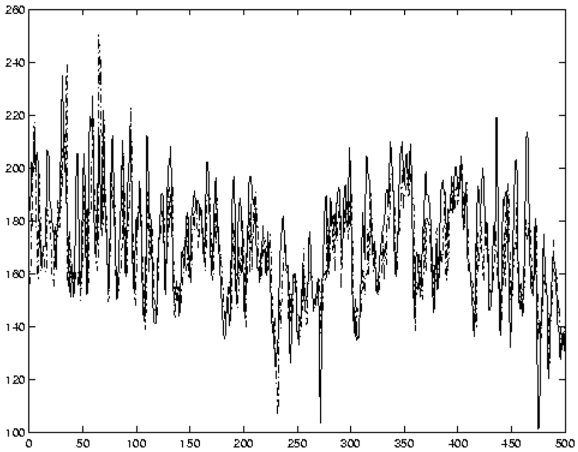


Figure 5-18 Garbage incinerator output variable simulated using a complexity-5 model, the inputs of which are those four variables that show the highest frequencies among all the complexity-5 masks. The candidate mask given to FIR is derived from the energy method

This section has shown a second method of deriving a FIR-model of a complex system by reducing the mask search space, in this case, by means of energy considerations. The following flowchart summarizes the primary aspects of the advocated method.

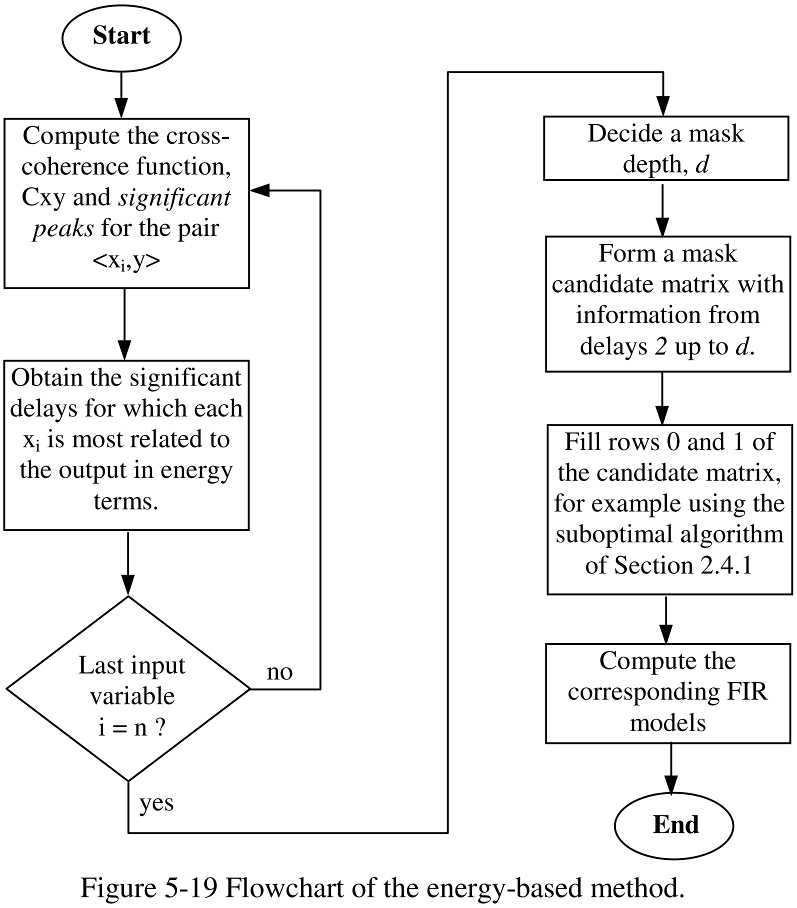

Start
Compute the cross-coherence function, Cxy and *significant peaks* for the pair <xi,y>
Obtain the significant delays for which each xi is most related to the output in energy terms.
Last input variable i = n ?
no
yes
Decide a mask depth, *d*
Form a mask candidate matrix with information from delays *2* up to *d*.
Fill rows 0 and 1 of the candidate matrix, for example using the suboptimal algorithm of Section 2.4.1
Compute the corresponding FIR models
End

Figure 5-19 Flowchart of the energy-based method.

5.4 Conclusions

The behaviour of systems can be predicted using either *a priori* knowledge (deductive techniques) or observations (inductive techniques). Some inductive prediction techniques were analysed in this chapter. All but the simplest of those techniques make predictions in two steps. In a first step, an input/output model is being created based on the observations made; in the second step, a simulation of the previously made model is then performed with the purpose of making predictions. All of the techniques surveyed in Section 5.2.1, first make a model that is then being used in a simulation.

When creating a model, the observations can either be used directly (quantitative modelling techniques), or they can first be discretised or at least fuzzified (qualitative modelling techniques). All of the techniques studied in this chapter make use of quantitative modelling techniques, except for FIR, which embraces a qualitative modelling approach.

The modelling process, be it quantitative or qualitative in nature, is usually performed in two steps. In a first step, the laws or equations that govern the system are being identified. In a second step, the model *parameters* are being estimated. Most of the techniques discussed in this chapter operate in such a fashion. The equations determine the set of variables to be used by the model. In the case of the techniques discussed in Sections 5.2.1.2 to 5.2.1.5 of the chapter, as well as those previously discussed in Sections 3.2 and 3.3, these variables are then being used, either directly or indirectly, in a linear regression model. The parameter estimation step determines the regression coefficients. FIR also starts out by determining the laws governing the system, i.e., by selecting the set of variables to be used in the simulation. However, no parameter estimation takes place, as FIR is a non-parametric technique. During its qualitative simulation, FIR refers directly back to the training data, rather than capturing the knowledge contained in the training data in a set of parameter values.

Models can be either *dynamic* or *static*, as defined in Chapter 3. In a dynamic model, the current value of the output may depend on its own past, as well as on current and past values of the inputs. In a static model, the current value of the output only depends on the current values of the inputs. All of the techniques advocated in this chapter may be used to create either static or dynamic models, though the majority of them were only demonstrated in the context of static modelling.

Since all of the discussed techniques first select a set of variables to be used, they can be arbitrarily combined with each other, i.e., any one of the techniques can be used to select the set of variables, which can then be used by either the same or any other technique to make predictions.

Sections 5.2.1.2 to 5.2.1.4 of the chapter analyse a set of statistical modelling and simulation techniques. All of these techniques were used exclusively for the creation of static models. In each subsection, a technique was used to select a subset of variables to be subsequently used in a linear regression analysis for making predictions. The simulation results obtained in this way were compared against simulations making use of the variables proposed by FIR, i.e., the first step of each technique was replaced by a FIR model selection, whereas the subsequent parameter identification and regression techniques were preserved from the methods discussed. It turned out that none of these *linear* modelling techniques did a very good job at choosing a pertinent subset of variables of the *non-linear* plant used as an example. The variables proposed by FIR were usually superior, even for the purpose of being used in *linear* regression models.

Section 5.2.1.5 discussed a set of clustering techniques for the purpose of variable selection. These are pure modelling techniques that can be combined with any of the previously discussed simulation approaches. No simulations were performed in that section. All of the techniques discussed in this section were used for static modelling only. It turned out that the techniques advocated in Section 5.2.1.5 were excellently suited for the purpose of variable selection.

Section 5.2.1.6 made use of the subsets of variables proposed by the different techniques presented in the earlier sections for the purpose of creating dynamic FIR models to be used in subsequent FIR simulations.

Of all the techniques discussed in this chapter, FIR is by far the best both in terms of its modelling capabilities as well as the power of its simulation engine. Hence FIR can be used as a gauge against which the other techniques can be measured. Yet, FIR is deplorably *slow* both during modelling and during simulation. FIR's modelling engine is of exponential computational complexity, at least if an exhaustive mask search is being used, and consequently, FIR is unsuited for dealing with large-scale models. Only Fuzzy Reconstruction Analysis (FRA) and Artificial Neural Networks (ANN) are yet slower in terms of creating models from observations. Hence FIR needs a booster technique. Some of the discussed approaches revealed themselves as excellently suited for such purpose.

In order to deal with the exponential complexity of the FIR modelling engine and simplify the mask search space of FIR, while using its entire modelling capabilities as well as the power of its simulation engine, two methodologies were proposed in Section 5.3 based on the work developed in this dissertation. Both methods use FIR in order to obtain a qualitative model of the system; yet, their computational effort is alleviated, by proposing to FIR a sparse mask candidate matrix obtained by means of a booster algorithm.

The first of these approaches makes use of a refined set of statistical techniques to minimise the number of '-1' elements in the mask candidate matrix. It is based on a decomposition of the system into subsystems. The second technique is based on energy considerations. It calculates the energy content from any input signal to the output signal at different frequencies, corresponding to different time delays. The former approach is more refined than the latter. However, it suffers from the fact that statistical techniques are inherently linear in nature, i.e., special considerations are needed to take into account the strongly non-linear nature of most engineering systems. The latter approach is based on engineering knowledge, as energy considerations are the true cornerstones of physics in general and engineering in particular. This approach is inherently non-linear, and therefore, does not require special techniques to be applicable to engineering systems.

Whereas the former approach is more computationally intensive than the latter, it results in a set of mask candidate matrices of fairly low complexity, leaving less work to FIR.

The energy-based method is very fast, but the resulting mask candidate matrix is of higher complexity, which slows down the subsequent FIR modelling step, yet may result in a final suboptimal model of slightly superior quality. Yet, both approaches are very competitive, and since both of them lead to suboptimal models, it may be justified to use both of them in parallel.

The reader may have gotten the impression that the two techniques introduced in Section 5.3 are *closed* algorithms that should always be executed as proposed. Yet, the author prefers to view these techniques as *open* methods, because any one of their individual steps may be performed independently and may be combined with other algorithms as needed. Thus, whereas the occasional user may prefer to use these algorithms as two black boxes, the more knowledgeable user may get more mileage out of analysing their internal steps separately and combining them in the most suitable ways to get the best performance in any given situation.

Chapter 6 presents a truly large-scale industrial application, and applies the two techniques introduced in Section 5.3 to it. Yet, rather than applying the two algorithms blindly as proposed, they are modified in suitable ways, employing the available engineering knowledge, in order to further increase their efficiency. The example demonstrates clearly the suitability of either technique to the task at hand, i.e., the efficient qualitative modelling of large-scale industrial processes for the purpose of making accurate predictions about their future behaviour.

6 Application of the proposed methodologies to an industrial system: a gas turbine for electric power generation.

6.0 **Abstract**

This chapter presents a real industrial application of the methods that have been discussed in the previous chapters. Concretely, a gas turbine for electric power generation is analysed and modelled. In the first two sections of the chapter, a brief description of the turbine principles as well as some remarks about its uses along time are given. In Section 6.3.2, a decomposition of the gas turbine system using the subsystem decomposition method offered in Section 5.3.1 is performed for the purpose of determining a high-quality FIR model of the turbine within acceptable computational time. In Section 6.3.3, the energy-based method proposed in Section 5.3.2 is applied to the turbine system.

6.1 **Introduction**

Gas turbines are central to many industrial plants and provide one of the primary means for electric power generation. Turbines are highly complex devices, with a large number of measurement variables to monitor, and with extremely high maintenance cost. This is particularly true for very large turbines that generate over 200 megawatts of electricity. It is of interest, as for any other industrial complex system, to find a way of reducing the maintenance cost while increasing the availability of the gas turbine. This could be done by means of computing an accurate model of the device that allows, if desired, to rapidly diagnose the cause of a trip or failure so that power can be restored as quickly as possible. Although fault detection and/or identification is not an objective of the current research, it is a direct consequence of it, because almost all fault detection systems are model based and thus depend heavily on technologies such as the one analysed in this research effort. The aim of this chapter is to demonstrate the feasibility of using the introduced methodologies in the analysis of an industrial large-scale system such as a modern gas turbine.

It has been mentioned that modern turbines are today frequently used for electric power generation, yet the concept of a turbine is very old. The earliest example of jet propulsion can be traced as far back as 150 BC to an Egyptian by the name of Hero. Hero invented a toy that rotated on top of a boiling pot due to the reaction effect of hot air or steam exiting several nozzles arranged radially around a wheel. He called this invention an *aeolipile*. It is drawn in Figure 6-1.

Many centuries later, in 1232, the Chinese used rockets to frighten enemy soldiers, so using what can be called the first missiles.

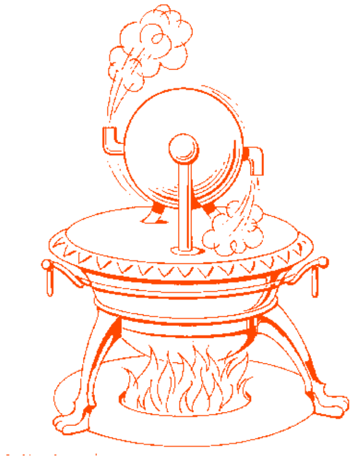


Figure 6-1 The Hero's *aeolipile*.

Around 1500 AD, Leonardo da Vinci drew a sketch of a device that rotated due to the effect of hot gasses flowing up a chimney. The device was intended for rotating meat being roasted. In 1629, another Italian by the name of Giovanni Branca actually developed a device that used jets of steam to rotate a turbine that in turn was used to operate machinery. This was the first practical application of a steam turbine. In 1678, Ferdinand Verbiest, a Jesuit in China, built a model carriage that used a steam jet for power.

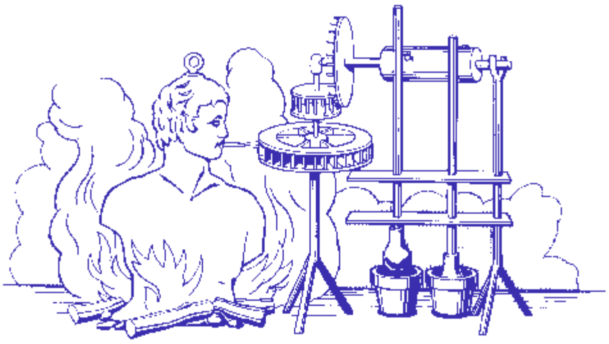


Figure 6-2 Steam turbine to operate machinery.

The first patent for a turbine engine was granted in 1791 to an Englishman by the name of John Barber. It incorporated many of the elements of a modern gas turbine, but used a reciprocating compressor. There are many more early examples of turbine engines designed by various inventors, but none were considered to be true gas turbines because they incorporated steam at some point in the process. In 1872, a man by the name of Stolze designed the first true gas turbine. His engine incorporated a multistage turbine section and a multi-stage axial flow compressor. He tested working models of his invention in the early 1900’s.

Charles Curtis was the inventor of the Curtis steam engine. He filed the first patent application for a gas turbine engine in the U.S. His patent was granted in 1914, but not without some controversy.

The General Electric Company started their gas turbine division in 1903. An engineer by the name of Standford Moss led most of the project. His most outstanding development was the General Electric turbo-supercharger during World War 1, although credit for the concept is given to Rateau of France. It used hot exhaust gasses from a reciprocating engine to drive a turbine wheel that in turn drove a centrifugal compressor used for supercharging. The revolutionary design of the turbo-supercharger made it possible to construct the first reliable gas turbine engines.

Sir Frank Whittle of Great Britain patented a design for a jet aircraft engine in 1930. He first proposed using the gas turbine engine for propulsion in 1928 while a student at the Royal Air Force College in Cramwell, England. In 1941, an engine designed by Whittle was the first successful turbojet aeroplane flown in Great Britain.

Concurrently with Whittle's development efforts, Hans von Ohain and Max Hahn, two students at Göttingen in Germany, developed and patented their own engine design in 1936. Their design was adopted by the Ernst Heinkel Aircraft Company. The German Heinkel Aircraft Company is credited with the first flight of a gas turbine-powered jet-propelled aircraft on August 27th, 1939. The HE178 was the first jet aeroplane to fly. The Heinkel HE178 developed 1100 lbs. of thrust and flew over 400 mph. Later came the ME262, a 500-mph fighter plane. By the end of the Second World War, more than 1600 of these planes had been built. The German jet engines were more advanced than their British counterparts and offered such features as blade cooling and variable-area exhaust nozzles.

In 1941, Frank Whittle began flight tests of a turbojet engine of his own design in England. Eventually, The General Electric Company manufactured engines in the U.S. that were based on Whittle's design.

Modern turbines implement many improvements with respect to the early designs; for example, modern sensors to monitor many different variables that participate in the turbine functioning cycle, or an expert system capable of identifying anomalous situations. Yet, they are based on the same principles, and there is nothing conceptually new about them.

The turbine that is going to be analysed in this chapter is a General Electric MARK 5 Frame 6 gas turbine. These kinds of turbines are single-shaft machines designed to generate electric power. A MARK 5 Frame 6 turbine can generate up to 40 megawatts of power. The analysed turbine is presently at use in an electric power generation plant located at Aylesford in the south of England. Data was obtained from this turbine thanks to a European project (ESPRIT project, nº 27548).

The aim of this chapter is to apply the different methods, separately or in combination, developed in the previous chapters that allow reducing the computation time of a FIR qualitative model. Up to now, computing a FIR model of such a large system had not been feasible due to the amount of computation time required for such purpose.

6.2 **Turbine principles**

A simple gas turbine is comprised of three main sections: a compressor, a combustor, and a power turbine. The gas turbine operates on the principle of the Brayton cycle, where compressed air is mixed with fuel and burned under constant pressure conditions. The resulting hot gas is allowed to expand through a turbine to perform work. In a 33% efficient gas turbine, 67% of this work is spent compressing the air, driving auxiliary devices, and overcoming inefficiency (mechanical friction); the rest is available for other work, such as a mechanical drive or electrical power generation.

Hence the gas turbine, like any other heat engine, is a device designed to convert a part of the chemical energy contained in a fuel into useful available mechanical power (later convertible to electric power). It does this in a manner similar in many ways to the cycle used by a four-stroke internal combustion engine. The main apparent difference is that work is accomplished in a continuous manner in the turbine power process, whilst in the reciprocating engine, this is an intermittent process. Figure 6-3 illustrates the basic cycle of a gas turbine.

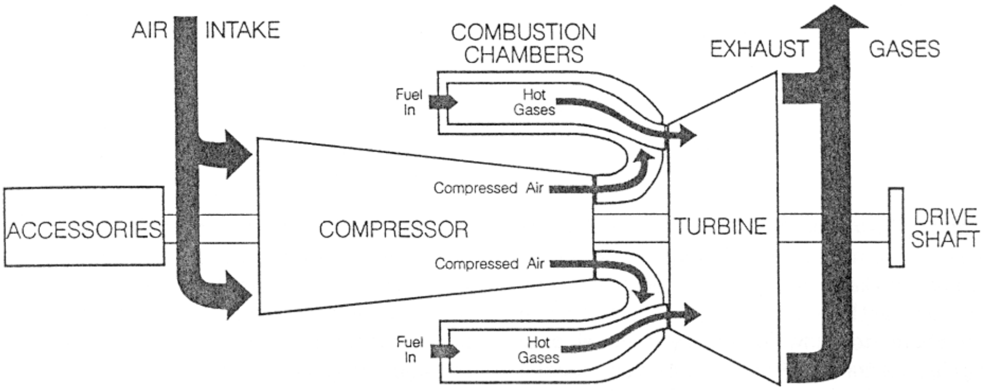


Figure 6-3 The basic cycle of a gas turbine.

As indicated by this illustration, air is drawn into the compressor through an air filter situated in the so-called filter house. This filter removes any harmful solid particles from the air stream before it enters the compressor. Although the air is filtered before being drawn into the compressor, small particles can still get into the compressor, which can gradually build up dirt on the compressor blades and hence reduce the efficiency of the compressor. The efficiency of the compressor is one of the key factors in the efficiency of the whole turbine. After the filtration stage, the air is compressed as it passes through the axial compressor thanks to multiple sets of fixed and rotating blades. In this compressor stage, the air, as well as being raised to a higher pressure, also becomes hotter. This hot compressed air is then fed to the combustion system, where it is mixed with injected fuel.

Several types of compressors are available for gas turbine applications, for example, the so-called *Centrifugal*, *Axial Flow,* and *Intermeshing Lobe* types. The turbine analysed in this research uses an axial compressor. It has the ability to pump large volumes of air at better efficiency than either the centrifugal or lobe types. Axial flow compressors are so designed that the air moves axially through the blading with essentially no radial travel.

This type of compressor is made of rows of air-foiled shape blades with each set of rotating blades followed by a set of similar stationary blades.

In the combustion chamber, the fuel burns and adds its energy to the air. The combustion zone of a turbine is the space required for the actual burning of the fuel and the subsequent dilution by secondary air from flame temperatures to usable values between 1650 and 1750 ^{o}F. The combustion chamber comprises an outer casing, an inner casing (or 'liner') and the necessary air and gas passages. The combustion process raises the air temperature to a flame zone value of between 2500 and 3200 ^{o}F which is immediately reduced to usable values by mixing secondary air that enters the combustion chamber through specifically placed holes.

The hot, high-pressure gas is then delivered to the turbine section from the combustion chamber at a temperature and flow required by the load. During its flow through nozzles and buckets (turbine blades), the gas loses both heat and pressure, until it is discharged from the final stage at exhaust stack pressure and temperature. It is within the turbine section of a gas turbine that a part of the thermal energy contained in the hot gas, provided by the combustion system, is converted to mechanical energy. In the turbine process, sufficient mechanical energy must be taken out of the gas stream to supply the necessary power to drive the axial flow main compressor and the driven auxiliaries, such as the accessory gear-box, fuel pump, cooling water pumps, lube oil pumps etc., provide for bearing frictional losses, and have enough excess power to do a reasonable amount of external work, such as drive an alternator, for electric power generation, a load compressor, or some other type of load equipment. In Figure 6-4, a general scheme of a gas turbine is shown that is used together with an alternator for electric power generation.

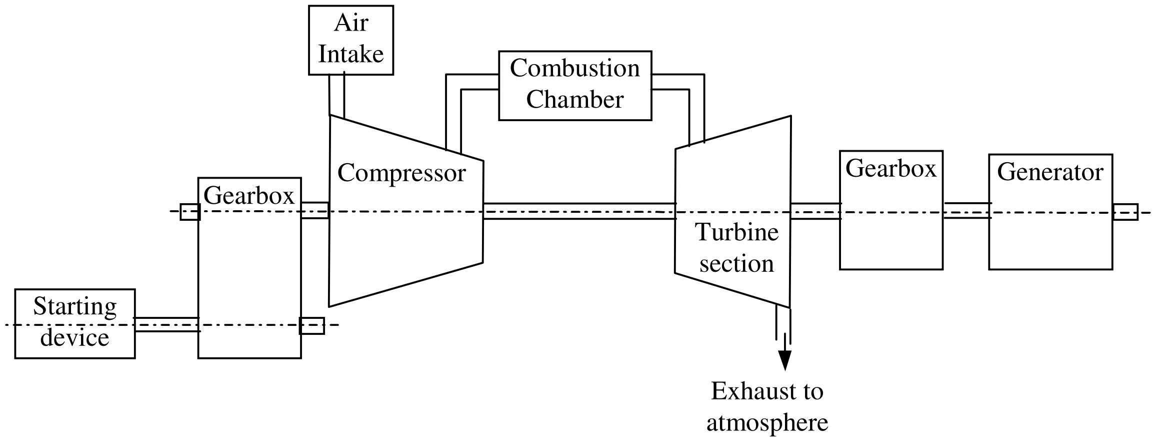


Figure 6-4 Simplified schematic of a gas turbine with a load generator for electric power generation.

Two main types of turbine blade designs are used for energy conversion, the so-called *Reaction* and *Impulse* types. In the first of these two types of blade designs, the hot gas is allowed to expand in both the rotating and stationary blading. This is an efficient method of extracting work from a gas stream, but since not much pressure drop can be used per stage, many stages are required. In the other type, the so-called Impulse type, most of the pressure drop occurs in the stationary elements with only a small percentage taking place in the rotating parts. This type has the advantage of being able to do more work per stage, hence fewer stages are required than with the Reaction type. It also permits larger pressure and temperature drops to occur in the stationary parts, rather than in the more highly stressed and difficult to cool rotating elements.

After the turbine section, the used gas is allowed to flow to the exhaust stack system, also called exhaust pipe. This air can either be released to the atmosphere or be used for other purposes, such as heating up water or as hot feed supply to a separately fired boiler. It is of interest to monitor the exhaust temperature and use it to model the system since it provides a means of measuring the energy left in the exhaust air, and, hence, calculating the turbine efficiency. Any heat recovery method employed will help to increase the overall thermal efficiency of the turbine cycle.

So far, descriptions of the turbine cycle have primarily been made for a single-shaft non-regenerative unit. Whilst this type of machine is simple, powerful, and reasonably good on thermal efficiency, some applications require more flexibility of operation and thermal efficiencies, than this basic machine configuration is capable of providing.

It is for this reason that the two-shaft machine was developed. This type of engine has the load turbine and the compressor turbine on separate shafts with a controllable angle nozzle in between. The angle nozzle is effectively a variable area orifice. This other type of turbine has different advantages, such as a better thermal efficiency, the ability to run compressor and load shafts at different speeds, so that best results can be achieved and a lower starting power is required. With wide-open nozzles, the ΔP over the high-pressure turbine is maximised, thus the turbine develops more power in the starting moment, which reduces the starting power required.

Higher overall thermal efficiencies may be achieved by the addition of a regenerator. In this component, the turbine exhaust gas is allowed to give up some of its heat to the compressor discharge air. In this way, heat that would be otherwise wasted is returned to the cycle, thereby reducing the required amount of fuel and increasing the output power.

As a general observation and in order to avoid confusion, the ‘gas’ in the gas turbine refers to the products of combustion when liquid or gaseous fuels are burnt in pressurised air in a closed chamber. The gas is at high temperature and pressure, which, when channelled through fixed nozzles, impinges on buckets mounted on the circumference of a rotor, causing the rotor to rotate.

In this section, it is not intended to provide a deep insight into the operation of gas turbines, but rather to offer a general description of its functioning cycle. We will not enter in the description of the thermo-dynamic equations that govern the turbine work. This is not necessary since it is intended to obtain a qualitative model of a gas-turbine using the data gathered from different sensors situated all along the engine. More information about gas turbines may be found in [Bathie, 1996].

6.3 **FIR model of a General Electric MARK 5 Frame 6 gas turbine**

As was mentioned in the introduction section of the present chapter, data was obtained from a General Electric MARK 5 Frame 6 turbine type, actually used in an electric power generation plant in the south of England, thanks to the European project ESPRIT nº 27548. Data have been gathered since the year 1997 using a sampling time of one second. The company that generates the electric power uses these data to feed an expert system that is capable of informing the plant operator when an anomalous situation occurs within the plant. In this research, a period of 2 hours worth of data on August 21, 1999 has been used to obtain a qualitative model of the gas turbine plant. 85% of those data (a total of 7200 data points are available) is used to compute the qualitative models whilst the remaining 15% will be used to validate the model.

Abnormal as well as normal situations can be found in this short period of considered time. More concretely, data corresponding to an emergency stop followed by a start up and the coupling of the turbine shaft with the generator are found beside from data representing normal operating conditions. The trajectory of the considered output variable, the generated electric power, is shown in Figure 6-5. In this way, the obtained model will be capable of simulating and predicting the system under normal operating conditions as well as detecting that an anomalous situation has occurred, although fault detection and identification is beyond the scope of the present research. The aim of the present investigation is limited to reducing the computation time of a FIR model, so that qualitative FIR models for large-scale systems, that were impossible to compute previously, can now be obtained.

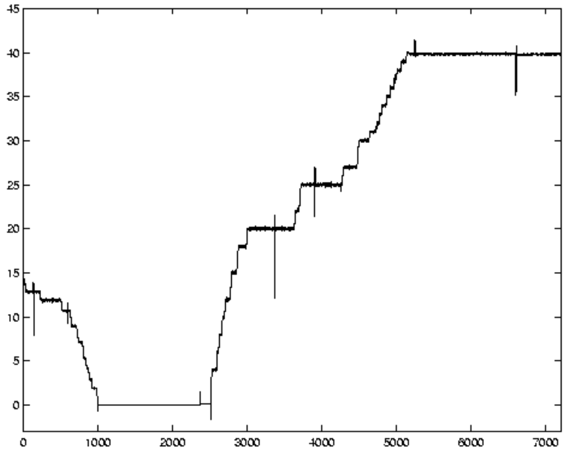


Figure 6-5 Trajectory of the considered gas turbine output (7200 data points).

6.3.1 System description

The General Electric MARK 5 Frame 6 turbine analysed in this research effort follows the general description of gas turbines offered in previous sections of this chapter. The turbine works along a single shaft, and has been designed to generate a maximum of 40 megawatts of electric power. The only particularity of this turbine with respect to the previously provided general description is that it can use either gas or liquid fuel in order to make the turbine rotate and generate electric power. Figure 6-6 provides a general schematic of the used turbine as well as the location in the system of some of the variables (only some temperatures, fluxes, and pressures have been depicted out of hundreds of variables) that can be measured.

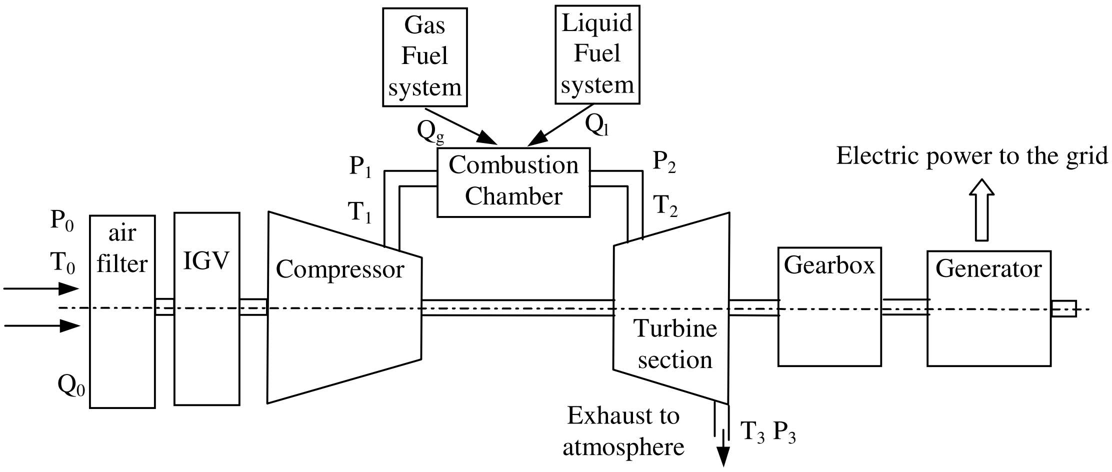


Figure 6-6 Schematic of the General Electric MARK 5 Frame 6 turbine.

A full list of the measured variables, including their names, measurement units, and a brief description, is given in the Appendix I.4. Not all of the variables described in this list have been used for the computation of the FIR qualitative model. Although the first idea was to apply the methods previously studied with all the system variables, it was rapidly decided that it made little sense to proceed in this way. This first idea originated because of the nature of the studied inquiry: we want to compute a qualitative model (particularly of a large-scale system) using only information contained in the system data, with, theoretically, no knowledge about the system being analysed. Yet, such an approach would have slowed down the analysis in unnecessary ways, and furthermore, it can be expected that a user possesses a basic understanding of the system that he or she is about to analyse. It seems wasteful, not to make use of such knowledge. The variable pre-selection, as used here, shall be fully explained and does not require any knowledge that the average user would not possess. Furthermore, the methodology explained in the sequel is not dependent on such a pre-selection. It would have worked without it, albeit a bit slower.

The first criterion was to discard all those variables that were logic signals, used by the expert system to analyse the status of the plant. This kind of output variables, although useful to the expert system as well as to the plant engineers, does not provide us with any useful information about the behaviour of the gas turbine, so only provoking an increase of the computational cost of the qualitative model. If left in the raw data, FIR would certainly not select any of these variables as inputs to the model of the generated electric power. Hence they can be safely discarded.

Second, we may wish to review here what it is understood by the word *model*. In Chapter 2, it was said that Fuzzy Inductive Reasoning is a modelling and simulation methodology capable of generating a qualitative input/output model of a system from real-valued trajectories of its physical variables. Hence it is wanted to find a causal relation between the input variables and the output variable (assuming a MISO system). Note that an input to the model, in the FIR context, can be either a physical input to the system, an internal variable, or even the same output delayed some sampling interval times. Therefore, only those variables measuring physical quantities that can directly influence the studied output must be considered as potential input variables. But how can we decide whether a variable may or may not influence the considered output? There will be variables that evidently influence the output, and others, for which a relation with the output is less obvious. In the given case, there are a number of local control loops used to regulate some physical properties of the system, such as flow rates, pressures, and temperatures. It may be acceptable to only keep the controlled quantities in the model, while discarding the local control variables, as those contain essentially redundant information, assuming that the time constants of the local control loops are considerably smaller than those of the overall plant to be modelled. Consider, for example, the control scheme depicted in Figure 6-7.

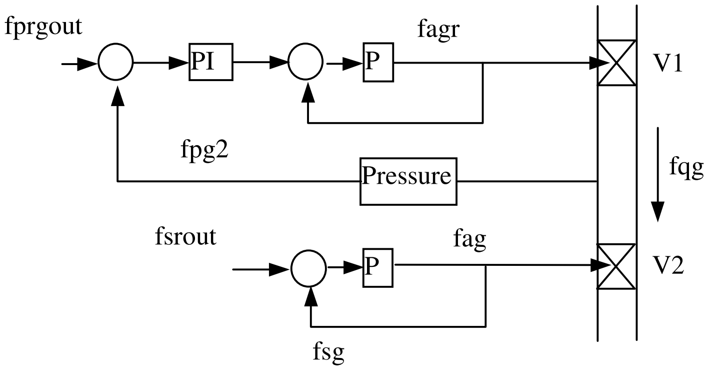


Figure 6-7 Control loops for the fuel gas input pipe.

This is a simplified scheme of the control module of the gas fuel system [Escobet *et. al*, 1999]. The main components of this control unit are two series-connected valves that control the flow of gas fuel that enters the combustion chambers. The first of these valves, V_1, is controlled by a feedback loop that maintains constant gas pressure at its output (named *fpg2* in Figure 6-7). The second valve, V_2, is a position-controlled valve. The meaning of the other variables appearing in Figure 6-7 is given in Table 6-I.

Variable name	Description
fpgrout	inter-valve pressure reference
fagr	V_1 servo current
fsgr	V_1 position
fpg2	inter-valve gas pressure
fqg	gas fuel flow
fsrout	V_2 position reference
fag	V_2 fag servo current
fsg	V_2 position

Table 6-I Variables of the gas fuel system control module.

From these eight variables, only one, the gas fuel flow, labelled *fqg*, may influence the considered output, i.e., the generated electric power, in a direct way. The other seven variables would only be useful if a model of the control gas fuel module were to be computed. For the purpose of modelling the generated electric power, the information contained in these other seven variables is essentially redundant. Therefore, the seven variables involved in the control loop may be safely discarded from the list to be offered to the FIR qualitative modelling engine. If left in the raw data, these redundant variables would be filtered out during the variable reduction stage.

Last, a third obvious criterion can be used to discard some more variables in this first pre-analysis stage. Those variables with a constant value, due either to a sensor malfunction or because the variable is indeed constant in the considered period of time of gathered data, can also be discarded. Variables with constant values do not carry any useful information about the dynamic behaviour of the system. Therefore, it is safe to discard them, so further reducing the initial computational complexity for a FIR qualitative model search.

Using the three techniques described above, only 64 variables out of a total of 209 measured variables were selected in this first pre-analysis stage to later apply to them some of the methods developed in the mark of this doctoral thesis. Table 6-II lists the names and offers a one-line description of the chosen variables. As previously mentioned, a full list of all measured variables is given in the Appendix I.4.

Variable name	Description
x_1, AFPBD	Compressor differential pressure
x_2, AFPCD	Compressor discharge pressure
x_3, AFQ	Compressor inlet air flow
x_4, AFQD	Compressor inlet dry airflow
x_5, ATID	Inlet air heating thermocouple
x_6, BB1	Vibration transducer #1
x_7, BB10	Vibration transducer #10
x_8, BB11	Vibration transducer #11
x_9, BB12	Vibration transducer #12
x_{10}, BB2	Vibration transducer #2
x_{11}, BB4	Vibration transducer #4
x_{12}, BB5	Vibration transducer #5
x_{13}, BTGJ1	Bearing metal temperature generator journal #1
x_{14}, BTGJ2	Bearing metal temperature generator journal #2
x_{15}, CMHUM	Specific humidity

x_{16}, CPD	Compressor discharge pressure
x_{17}, CTD	Compressor discharge temperature
x_{18}, CTDA1	Compressor temperature thermocouple #1
x_{19}, CTDA2	Compressor temperature thermocouple #2
x_{20}, CTIF1	Compressor inlet flange temperature – thermocouple #1
x_{21}, CTIF2	Compressor inlet flange temperature – thermocouple #2
x_{22}, DF	Generator frequency
x_{23}, DTGGC10	Generator cold gas rtd
x_{24}, DTGGC11	Generator cold gas rtd
x_{25}, DTGGH18	Generator hot gas rtd
x_{26}, DTGGH19	Generator hot gas rtd
x_{27}, DV	Generator line voltage
x_{28}, FQG	Gas fuel flow
x_{29}, FQLM1	Liquid fuel mass flow, LFS
x_{30}, SDSJ1	Steam injection low-range differential pressure
x_{31}, SDSJ2	Steam injection high-range differential pressure
x_{32}, SPSJ1	Steam injection supply pressure
x_{33}, STSJ	Steam injection temperature
x_{34}, SVL	System line voltage
x_{35}, TGSDIFU1	Generator U phase winding temperature. #1, main input exciter
x_{36}, TGSDIFU2	Generator U phase winding temperature. #2, centre
x_{37}, TGSDIFV1	Generator V phase winding temperature. #1, exciter end
x_{38}, TGSDIFV2	Generator V phase winding temperature. #2, centre
x_{39}, TGSDIFW1	Generator W phase winding temperature. #1, exciter end
x_{40}, TGSDIFW2	Generator W phase winding temperature. #1, centre
x_{41}, TIFDP1	Gas turbine inlet filter differential pressure, #2
x_{42}, TIFDP2	Gas turbine inlet filter differential pressure, #3
x_{43}, TIFDP3	Gas turbine inlet filter differential pressure, #4
x_{44}, TNH	Turbine rotation speed
x_{45}, TTWS1AO1	Turbine wheelspace temperature 1st stage aft. outer, #1
x_{46}, TTWS1AO2	Turbine wheelspace temperature 1st stage aft. outer, #2
x_{47}, TTWS1FI1	Turbine wheelspace temperature 1st stage fwd. inner, #1
x_{48}, TTWS1FI2	Turbine wheelspace temperature 1st stage fwd. inner, #2
x_{49}, TTWS1FO1	Turbine wheelspace temperature 1st stage fwd. outer, #1
x_{50}, TTWS1FO2	Turbine wheelspace temperature 1st stage fwd. outer, #2
x_{51}, TTWS2AO1	Turbine wheelspace temperature 2st stage aft. outer, #1
x_{52}, TTWS2AO2	Turbine wheelspace temperature 2st stage aft. outer, #2
x_{53}, TTWS2FO1	Turbine wheelspace temperature 2st stage fwd. outer, #1
x_{54}, TTWS2FO2	Turbine wheelspace temperature 2st stage fwd. outer, #2
x_{55}, TTWS3AO1	Turbine wheelspace temperature 3st stage aft. outer, #1
x_{56}, TTWS3AO2	Turbine wheelspace temperature 3st stage aft. outer, #2
x_{57}, TTWS3FO1	Turbine wheelspace temperature 3st stage fwd. outer, #1
x_{58}, TTWS3FO2	Turbine wheelspace temperature 3st stage fwd. outer, #2
x_{59}, TTXM	Exhaust temperature
x_{60}, TTXSP1	Combustion actual spread, #1
x_{61}, TTXSP2	Combustion actual spread, #2
x_{62}, TTXSP3	Combustion actual spread, #3
x_{63}, WQJ	Wet low NOX injection flow
y, DWATT	Generator load watts

Table 6-II Turbine variables used to compute the FIR qualitative model.

The last variable of Table 6-II is the considered output variable, i.e., the variable to be modelled, whereas the other 63 variables are considered potential inputs to the system. The turbine plant could be considered as a MIMO system instead of a MISO system, if other variables such as the *DV* or *DF* variables were considered as additional outputs. Here, only the generator load power has been considered as output. If a MIMO system were to be analysed, the steps presented in the next sections would have to be applied separately for every output, taking into account that every output variable may be considered as a potential input variable for each of the other output variables.

6.3.2 Turbine FIR model from its subsystem decomposition

In chapter 5, two different methodologies were presented that enable a user to identify a qualitative model of a large-scale system within reasonable time limits. One of these methodologies is based in obtaining subsystems of the whole system using a statistical approach.

The first step of the developed methodology is to perform a first rough variable selection via a correlation analysis of the data. It is well known that correlation and covariance measures provide measures of linear association, or, in other words, association along a line. When using this kind of indices to study possible relations among variables, one must consider that possible non-linear relations may exist that cannot be revealed by these descriptive statistics. These non-linear relations are investigated in a posterior step of the methodology. Also, these statistics are very sensitive to outlier observations that may indicate a relation when in fact there exists very little relation if any. In spite of its shortcomings, the correlation index is a very useful tool as it is easy and quick to compute and may offer a first insight about what variables are related to which others when the structure of a system is being investigated.

The first step of the proposed methodology is thus based on a simple idea: to identify those variables that contain redundant information about the system and eliminate them from the set of variables to be used in the subsequent analysis, so simplifying a posterior model computation. A correlation analysis of the input data is performed, obtaining the data correlation matrix (Equation 3.21), so variables with a strong linear relationship can be detected. Afterwards, those variables with an absolute linear correlation coefficient equal to or greater than an absolute upper limit r_0 are identified. The variables in these groups contain almost identical information about the studied process, and therefore, any one of them may be chosen to represent such information. The criterion used to select this variable, as described in Section 3.4.1, is to maximise the variation coefficient (Equation 3.22). This index is a normalisation of each variance variable using its mean as normalisation factor.

Then, as explained in Chapter 5, a singular value decomposition of the newly reduced correlation matrix is performed, so its eigenvalues and eigenvectors are obtained. Then, the orthogonal eigenvectors are projected onto the principal axes. Theoretically, the projections onto all of subspaces spanned by each pair of eigenvectors have to be examined, but in practice, it suffices to take into account only those projections that account primarily for the system variance. Then for each projection, the axes are divided into sectors of 30º, and the variables with larger projection in the obtained sections are joined into the same subsets. In this way subgroups of linearly related variables are obtained.

Finally, there exists the possibility that some of the system variables do not form part of any of the found subsystems due to the fact that, since now, only linear relations among variables have been considered. Possible non-linear relations between each of those non-included variables and the obtained subsets should be investigated, in a way that, if a significant non-linear correlation is found between a variable and any subsystem, the variable should be included in it.

Two decompositions of the gas turbine system have been performed. The first decomposition does not include time in the analysis, so the obtained model is static. The second subsystem decomposition extends the analysis with the inclusion of time. Time is included enlarging the data matrix to the right with the trajectories of the variables delayed one sampling interval each time.

6.3.2.1 Static system decomposition

The analysis starts by applying correlation analysis to the chosen variables of the gas turbine system as given in Table 6-II. The sample correlation matrix for the 63 input variables and one output of the turbine system is not reported here. It is not useful to look at a matrix of 64 x 64 = 4096 real elements and analyse it by hand. The described variable selection step is performed automatically using Matlab. For this example, the value chosen for r_0 has been $r_0 = 0.75$, exactly the same as it was used in the garbage incinerator example of Chapter 3. This value has been considered sufficiently high to discriminate whether a linear correlation exists or not. Table 6-III lists the variables chosen by this first variable selection step. 20 of the 63 input variables survive this first variable reduction stage.

Now with those remaining input variables plus the added output, subsets (subsystems) of variables are formed that are linearly correlated among each other. The process used to this aim, as explained in Chapter 5, is based on the first steps of a principal component analysis. First, the correlation matrix of the remaining variables has to be computed. Actually it is not necessary to re-compute this matrix because the full correlation matrix for the system under consideration has been computed in the previous variable selection step. Since the linear correlation between variables is independent of the presence or absence of other variables, it is only necessary to eliminate those rows and columns pertaining to the removed variables. In this way, a new $n \times n$ matrix, $n = 21$, of correlation coefficients is obtained. Then a singular value decomposition of the newly reduced correlation matrix is performed. The obtained eigenvectors are projected onto the principal axes. In the sequel, the axes are divided into sectors of 30º, and the variables with larger projection in the obtained sections are joined into the same subsets.

Variable	Description
x_2, AFPCD	Compressor discharge pressure
x_5, ATID	Inlet air heating thermocouple
x_6, BB1	Vibration transducer #1
x_8, BB11	Vibration transducer #11
$x_{9,}$ BB12	Vibration transducer #12
x_{11}, BB4	Vibration transducer #4
x_{16}, CPD	Compressor discharge pressure
x_{17}, CTD	Compressor discharge temperature
x_{20}, CTIF1	Compressor inlet flange temperature – thermocouple #1
x_{27}, DV	Generator line voltage
x_{28}, FQG	Gas fuel flow
x_{29}, FQLM1	Liquid fuel mass flow, LFS
x_{31}, SDSJ2	Steam injection high-range differential pressure
x_{34}, SVL	System line voltage
x_{42}, TIFDP2	Gas turbine inlet filter differential pressure, #3
x_{48}, TTWS1FI2	Turbine wheelspace temperature 1st stage fwd. inner, #2
x_{56}, TTWS3AO2	Turbine wheelspace temperature 3rd stage aft. outer, #2
x_{59}, TTXM	Exhaust temperature
x_{60}, TTXSP1	Combustion actual spread, #1
x_{63}, WQJ	Wet low NOX injection flow
y, DWATT	Generator load watts

Table 6-III Turbine variables chosen after the correlation analysis of the data has been performed

In Figure 6-8, the projections of the principal components onto the first and second principal axes are shown. The projection space is divided into 30° sectors, and those variables with large projection (here, as in Chapters 4 and 5, a large projection means a modulus of the projection $p > r_0$) within these sectors are considered to form subsets of linearly related variables.

The subsystems generated from Figure 6-8 for the turbine system are listed in Table 6-IV.

	Formed subsystems								
S1	x_2	x_{11}	x_{16}	x_{17}	x_{20}	x_{27}	x_{34}	x_{42}	x_{48}
S2	x_6	x_{11}							
S3	x_1	x_8	x_{28}	x_{48}	x_{56}	x_{59}	y		
S4	x_{31}	x_{63}							

Table 6-IV Subsystems formed from the projection of the first versus the second principal axes.

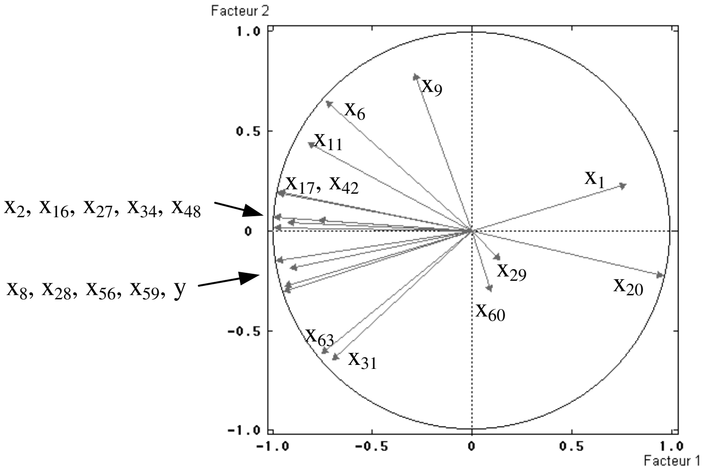


Figure 6-8 Projection onto the first and second principal axes.

As dividing the figure into 30º sections is a heuristic criterion, those variables located at the boundaries between neighbouring division are taken into account in both of them. This is why variable X_{11} is in the first, S1, and second, S2, subsystems listed in Table 6-IV.

The algorithm proceeds by looking at other axes projections. The next one to analyse is the projection of the first versus the third axes. Figure 6-9 shows this projection.

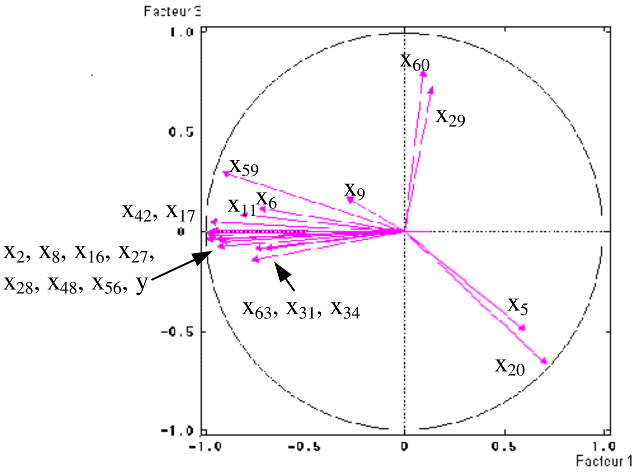


Figure 6-9 Projection onto the first and third principal axes.

From this new projection, different subsystems may be formed from those previously constructed using the information contained in the projection of the first and second principal axes. They are listed in Table 6-V.

	Formed subsystems								
S5	x_2	x_8	x_{16}	x_{27}	x_{28}	x_{42}	x_{48}	x_{56}	y
S6	x_{17}	x_{42}	x_{59}						
S7	x_5	x_{20}							
S8	x_{29}	x_{60}							

Table 6-V Subsystems formed from the first versus second principal axes projection.

Again, variable X_{42} is in the S5 and S6 subsystems listed in Table 6-V.

As further projections are analysed, no new subsets of variables are obtained, so this analysis can be terminated at this point. Note that, conversely to what happened with the garbage incinerator example given in Chapter 4, no redundant subsets of variables were encountered for the gas turbine system. Therefore, at this stage of the subsystem decomposition, we have 8 subsystems with varying numbers of variables. It is also important to note than not all of the 21 variables that survived the variable reduction stage were included in at least one of the obtained subsystems. Concretely, variable x_9 does not form part of any subsystem.

Now, for variable x_9, which was not included in any of the subsystems, a possible non-linear relation between it and the formed subsets should be considered. The variable is found to exhibit high non-linear correlation[1] with all of the formed subsystems except for S2. So this variable is added to every one of those subsystems. Finally in Table 6-VI, the obtained subsystem decomposition is listed.

	Final formed subsystems									
S1	x_2	x_9	x_{11}	x_{16}	x_{17}	x_{20}	x_{27}	x_{34}	x_{42}	x_{48}
S2	x_6	x_{11}								
S3	x_1	x_8	x_9	x_{28}	x_{48}	x_{56}	x_{59}	y		
S4	x_9	x_{31}	x_{63}							
S5	x_2	x_8	x_9	x_{16}	x_{27}	x_{28}	x_{42}	x_{48}	x_{56}	y
S6	x_9	x_{17}	x_{42}	x_{59}						
S7	x_5	x_9	x_{20}							
S8	x_9	x_{29}	x_{60}							

Table 6-VI Final formed subsystems for the gas turbine system.

From this subsystem decomposition, it can be noted that two of the subsystems include the output variable within them, namely subsystems S2 and S5 in Table 6-VI. These two subsystems can be considered as models of the output. This would be equivalent to having found, in this case, two different FIR masks, the first with complexity 8 and the second with complexity 10. Yet, those models have been obtained without taking into account the possible delays between the signals. As these models are not very useful by themselves, no simulations are added at this point.

[1] Here, 'high non-linear correlation' refers to the same limit used when computing the linear correlation, i.e., if a variable has a non-linear correlation $R_{nl} > 0.75$ with respect to a subset of variables, it is considered to form part of it.

6.3.2.2 Extended system decomp osition including time

The subsystem decomposition method presented in Chapter 5, which has been applied in the previous section to the gas turbine system for obtaining a static model, can be extended with the inclusion of time to obtain a dynamic model. Time can be introduced in the analysis by means of duplicating several times the raw data model, shifting each copy up by one row relative to the previous one, and concatenating the shifted data from the right to the already existing data set. The method had been introduced already in Chapter 5. Contrary to the algorithm proposed in Chapter 2, the statistical approach advocated in Chapter 5 does not offer a recipe for how to choose the number of delays, i.e., the mask depth, to be considered. In the given study, a mask depth of 6 was chosen, i.e., 5 copies of the original data set are made and concatenated to the original data from the right. If the models obtained in this way are unsatisfactory, the user may need to increase the mask depth and repeat the analysis. Of course, it is always possible in such a case to only preserve those columns of the already enlarged data set that were previously selected when concatenating more copies of the original data set from the right. In this way, the modelling effort that was made before will not be lost.

When 6 time delays are considered, the raw data model turns into a huge matrix of dimensions 7195 × 384. Now a system of 384 variables, comprising the original and the delayed ones, is to be decomposed into subsystems. The analysis begins again by applying correlation analysis to all input variables of the enlarged gas turbine system, corresponding to those variables listed in Table 6-II as well as their delayed versions up to (t-$5\delta t$). Of course, the output variable does not take part in this first step of the methodology. The sample correlation matrix for those 383 possible input variables is not reported here. A value of $r_0 = 0.75$ was chosen for the analysis just like before. In Table 6-VII, a list of the variables surviving this first variable reduction step is given. Only 28 of the original 383 input variables survived the selection. Hence this first variable reduction step is very important. It reduces the number of variables to be retained drastically. In fact, the number of variables to be analysed henceforth is not much larger than in the case of the static model, where 20 out of 63 possible input variables had survived the variable reduction step.

In this case, it happened that the output one time step delayed, $y(t\text{-}\delta t)$, was eliminated from the set of variables to be retained. This is understandable under the given circumstances. From each group of variables, only the one with the largest variability index survives. Since $y(t)$ is a controlled variable, its values do not change much except during transition periods between different set values of the control inputs. Although $y(t\text{-}\delta t)$ formed part of a number of different groups, each group contained at least one other variable with a larger variability index. Hence $y(t\text{-}\delta t)$ was eliminated. Since $y(t\text{-}\delta t)$, in any system, is the variable that contains most information about the output to be predicted, $y(t)$, it was decided to add this variable to the set of retained variables.

With this rough variable selection, those variables that are redundant, except for $y(t\text{-}\delta t)$, are eliminated from the posterior modelling process. Now with the remaining input variables and the added output, subsets (subsystems) of variables are formed that are linearly correlated among each other. The process used to this aim is based on the first steps of a principal component analysis.

The correlation matrix of the remaining input variables and the output can be derived from the total correlation matrix (with dimensions 384 × 384) previously computed by throwing out those rows and columns that correspond to the eliminated variables. In this way, a new $n \times n$ matrix, $n = 30$, of correlation coefficients is obtained. Then, a singular value decomposition of this matrix is performed, and the obtained eigenvectors are projected onto the principal axes. Then for each relevant projection, the axes are divided into sectors of 30º, and the variables with larger projection in the obtained sections are joined into the same subsets.

Variable	Description
x_5, ATID	Inlet air heating thermocouple
$x_{9,}$ BB12	Vibration transducer #12
x_{11}, BB4	Vibration transducer #4
x_{20}, CTIF1	Compressor inlet flange temperature – thermocouple #1
x_{27}, DV	Generator line voltage
x_{28}, FQG	Gas fuel flow
x_{29}, FQLM1	Liquid fuel mass flow, LFS
x_{44}, TNH	Turbine rotation speed
x_{60}, TTXSP1	Combustion actual spread, #1
y, DWATT	Generator load watts
y(t-δt), DWATT	One sampling time delayed version of y
x_5(t-δt), ATID	One sampling time delayed version of x_5
x_{28}(t-δt), FQG	One sampling time delayed version of x_{28}
x_5(t-2δt), ATID	Two sampling times delayed version of x_5
x_{28}(t-2δt), FQG	Two sampling times delayed version of x_{28}
x_{29}(t-2δt), FQLM1	Two sampling times delayed version of x_{29}
x_{44}(t-2δt), TNH	Two sampling times delayed version of x_{44}
x_5(t-3δt), ATID	Three sampling times delayed version of x_5
x_9(t-3δt), BB12	Three sampling times delayed version of x_9
x_{28}(t-3δt), FQG	Three sampling times delayed version of x_{28}
x_{31}(t-3δt), SDSJ2	Three sampling times delayed version of x_{31}
x_{42}(t-3δt), TIFDP2	Three sampling times delayed version of x_{42}
x_5(t-4δt), ATID	Four sampling times delayed version of x_5
x_9(t-4δt), BB12	Four sampling times delayed version of x_9
x_{20}(t-4δt), CTIF1	Four sampling times delayed version of x_{20}
x_{34}(t-4δt), SVL	Four sampling times delayed version of x_{34}
x_{56}(t-4δt), TTWS3AO2	Four sampling times delayed version of x_{56}
x_{60}(t-4δt), TTXSP1	Four sampling times delayed version of x_{60}
x_5(t-5δt), ATID	Five sampling times delayed version of x_5
x_{28}(t-5δt), FQG	Five sampling times delayed version of x_{28}

Table 6-VII Turbine variables chosen after the correlation analysis of the data including time

Figure 6-10 shows the projection of the principal components onto the first and second principal axes. The figure is divided into 30° sectors, and those variables with high projection, that is a modulus of the projection $p > r_0$ in these sectors, are considered to form subsets of linearly related variables. Subsystems obtained from Figure 6-10 are listed in Table 6-VIII.

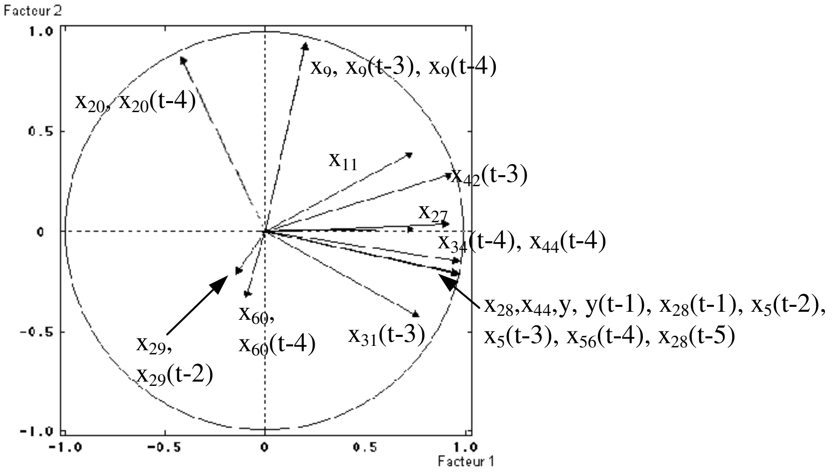


Figure 6-10 Projection onto the first and second principal axes when time is considered.

	Formed subsystems											
S1	x_9	$x_9(t\text{-}3)$	$x_9(t\text{-}4)$									
S2	x_{27}	$x_{42}(t\text{-}3)$										
S3	$x_5(t\text{-}2)$	$x_5(t\text{-}3)$	x_{27}	x_{28}	$x_{28}(t\text{-}1)$	$x_{28}(t\text{-}5)$	$x_{31}(t\text{-}3)$	x_{44}	$x_{44}(t\text{-}2)$	$x_{56}(t\text{-}4)$	$y(t\text{-}1)$	y
S4	x_{20}	$x_{20}(t\text{-}4)$										

Table 6-VIII Subsystems formed from the first versus second principal axes projection when time is considered.

As dividing the figure into 30° sectors is a heuristic criterion, those variables pertaining to the neighbourhood of each division are taken into account in both of the neighbouring sectors. This is why variable x_{27} is in the second, S2, and third, S3, subsystems listed in Table 6-VIII.

The method then proceeds by looking at other axes projections. The next one to analyse is the projection of the first versus the third axes, depicted in Figure 6-11. In this way projections of decreasing explained variance are analysed.

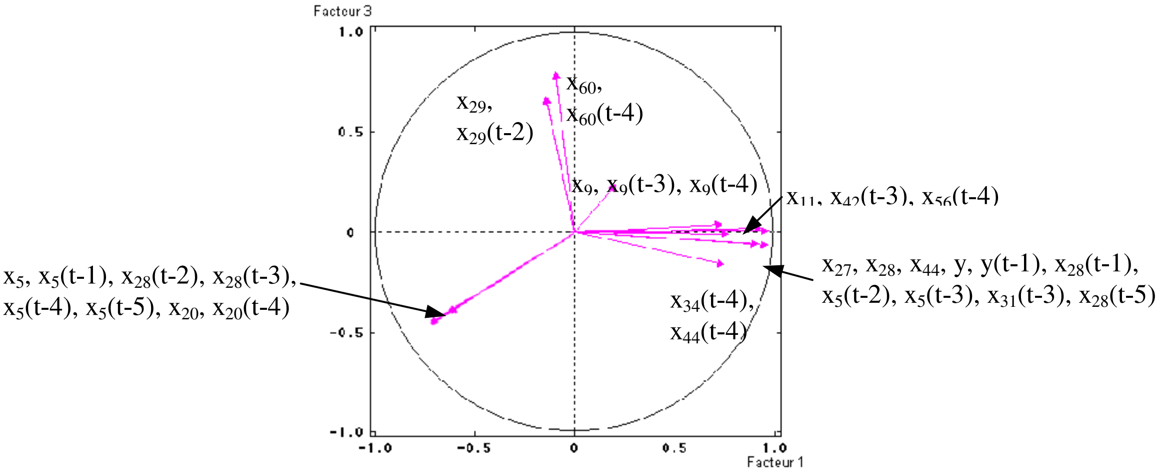


Figure 6-11 Projection onto the first and third principal axes when time is included.

From this new projection different subsystems may be formed from those previously constructed using the information contained in the projection of the first and second principal axes. The newly obtained subsystems are listed in Table 6-IX.

	Formed subsystems								
S5	x_5	x_5(t-1)	x_5(t-4)	x_5(t-5)	x_{20}	x_{20}(t-4)	x_{28}(t-2)	x_{28}(t-3)		
S6	x_{60}	x_{60}(t-4)								
S7	x_{11}	x_{31}(t-3)	x_{42}(t-3)	x_{56}(t-4)						
S8	x_5(t-2)	x_5(t-3)	x_{27}	x_{28}	x_{28}(t-1)	x_{28}(t-5)	x_{31}(t-3)	x_{44}	y(t-1)	y

Table 6-IX Subsystems formed from the first versus third principal axes projection when time is included in the analysis.

Again, variable *x_{31}(t-3)* is in the S7 and S8 subsystems listed in Table 6-IX.

Other projections can still be considered. In the case at hand, the projection of the first versus the fourth axes gives subsystem information that is not included in the previous analysed projections. This projection is shown in Figure 6-12.

From this third considered projection different subsystems may be formed from those previously constructed. The newly obtained subsystems are listed in Table 6-X.

	Formed subsystems									
S9	x_{20}	x_{20}(t-4)								
S10	x_5	x_5(t-1)	x_5(t-4)	x_5(t-5)	x_{28}(t-2)	x_{28}(t-3)				
S11	x_{31}(t-3)	x_{56}(t-4)								
S12	x_5(t-2)	x_5(t-3)	x_{27}	x_{28}	x_{28}(t-1)	x_{28}(t-5)	x_{42}(t-3)	x_{44}	y(t-1)	y
S13	x_{11}	x_{34}(t-4)	x_{44}(t-2)							

Table 6-X Subsystems formed from the first versus fourth principal axes projection when time is included in the analysis.

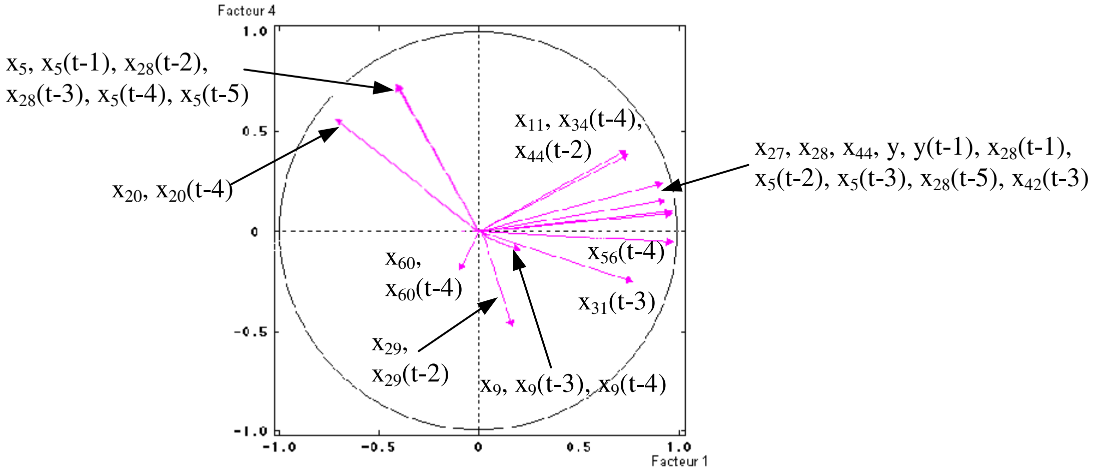


Figure 6-12 Projection onto the first and fourth principal axes when time is included.

Theoretically all the projections between different axes should be taken into account. In practice, if other projections are analysed, no new subsets of variables are obtained, so the algorithm can be terminated at this point.

In this analysis, i.e., the gas turbine system including time information, redundant subsets of variables have been encountered, as it happened in Chapter 4 with the garbage incinerator example. Note that subsystem S2 is included in S12, subsystems S4, S9 and S10 are included in S5, and subsystems S8 and S11 are included in S3. Those redundant subsystems do not add new information to the analysis of the system, as they form part of other bigger subsystems. Therefore, they are eliminated from the set of subsystems. The remaining subsystems are listed in Table 6-XI, in which they have been relabelled from S1 to S7.

	Formed subsystems											
S1	x_9	$x_9(t-3)$	$x_9(t-4)$									
S2	$x_5(t-2)$	$x_5(t-3)$	x_{27}	x_{28}	$x_{28}(t-1)$	$x_{28}(t-5)$	$x_{31}(t-3)$	x_{44}	$x_{44}(t-2)$	$x_{56}(t-4)$	$y(t-1)$	y
S3	x_5	$x_5(t-1)$	$x_{28}(t-2)$	$x_{28}(t-3)$	$x_5(t-4)$	$x_5(t-5)$	x_{20}	$x_{20}(t-4)$				
S4	x_{60}	$x_{60}(t-4)$										
S5	x_{11}	$x_{31}(t-3)$	$x_{42}(t-3)$	$x_{56}(t-4)$								
S6	$x_5(t-2)$	$x_5(t-3)$	x_{27}	x_{28}	$x_{28}(t-1)$	$x_{28}(t-5)$	$x_{42}(t-3)$	x_{44}	$y(t-1)$	y		
S7	x_{11}	$x_{34}(t-4)$	$x_{44}(t-2)$									

Table 6-XI Considered subsystems formed from linear relations between variables when time is included in the gas turbine analysis.

At this stage of the subsystem decomposition, we have 7 subsystems with varying numbers of variables. It is important to note than in the previous analysis not all the variables that have been considered have been included in at least one of the obtained subsystems. Concretely, variables x_{29} and $x_{29}(t-2\delta t)$ do not form part of any subsystem.

Now, for those variables that have not been included in any of the subsystems, a possible non-linear relation between them and the formed subsets is being considered.

Both, x_{29} and $x_{29}(t\text{-}2\delta t)$ are found to exhibit high non-linear correlation ($R_{nl} > 0.75$) with the formed subsystem S4. Variable $x_{29}(t\text{-}2\delta t)$ is found to also exhibit non-linear correlation with subsystem S7. So the S4 and S7 subsystems are augmented with the corresponding variables. In Table 6-XII, the final subsystem decomposition is provided.

	Final formed subsystems											
S1	x_9	x_9(t-3)	x_9(t-4)									
S2	x_5(t-2)	x_5(t-3)	x_{27}	x_{28}	x_{28}(t-1)	x_{28}(t-5)	x_{31}(t-3)	x_{44}	x_{44}(t-2)	x_{56}(t-4)	y(t-1)	y
S3	x_5	x_5(t-1)	x_{28}(t-2)	x_{28}(t-3)	x_5(t-4)	x_5(t-5)	x_{20}	x_{20}(t-4)				
S4	x_{29}	x_{29}(t-2)	x_{60}	x_{60}(t-4)								
S5	x_{11}	x_{31}(t-3)	x_{42}(t-3)	x_{56}(t-4)								
S6	x_5(t-2)	x_5(t-3)	x_{27}	x_{28}	x_{28}(t-1)	x_{28}(t-5)	x_{42}(t-3)	x_{44}	y(t-1)	y		
S7	x_{11}	x_{29}(t-2)	x_{34}(t-4)	x_{44}(t-2)								

Table 6-XII Final susbsystem decomposition when time is included in the gas turbine system.

Note that two of the final obtained subsystems include the output variable within them, those are subsystems S2 and S6. Those subsystems will be used to postulate candidate masks to FIR and hence derive a model of the considered output[2]. The following candidate masks are separately proposed to FIR:

$$mcan_{S2} = \begin{array}{c} \begin{array}{ccccccc} x_5 & x_{27} & x_{28} & x_{31} & x_{44} & x_{56} & y \end{array} \\ \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 0 & 1 \end{pmatrix} \end{array} \qquad mcan_{S6} = \begin{array}{c} \begin{array}{cccccc} x_5 & x_{27} & x_{28} & x_{42} & x_{44} & y \end{array} \\ \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 1 \end{pmatrix} \end{array}$$

When the optimal mask search algorithm is used with these candidate masks, the following optimal models are obtained, one for each of the allowed complexities. They are listed together with their respective qualities:

$$y_{S2}(t) = f\{y(t\text{-}\delta t)\}\quad Q = 0.9959$$
$$y_{S2}(t) = f\{\,x_{28}(t),\, y(t\text{-}\delta t)\}\quad Q = 0.9741$$
$$y_{S2}(t) = f\{\,x_5(t\text{-}2\delta t),\, x_{28}(t),\, y(t\text{-}\delta t)\}\quad Q = 0.8262$$
$$y_{S2}(t) = f\{\,x_5(t\text{-}2\delta t),\, x_{28}(t),\, x_{31}(t\text{-}3\delta t),\, y(t\text{-}\delta t)\}Q = 0.6603$$

[2] In fact, subsystems containing the output variable form by themselves models of the studied system. Yet, those subsystems are composed of more than 5 variables; in the given case, 9 variables for S2 and 8 variables for S6. This determines the complexity of the FIR model. In Chapter 2, it was discussed that the higher the complexity is, the more deterministic the model becomes; yet, the predictiveness of such a model may be very poor since, probably, already the next predicted state has never been observed before. Much more data points than those that are available here would be needed so as to justify the use of complexity 8 or 9 FIR models. This is the reason for proposing the search of complexity 5 models. Moreover, FIR models can only be truly compared for equal complexity, and complexity 5 models have been used all throughout the dissertation.

$$y_{S6}(t) = f\{y(t\text{-}\delta t)\} \quad Q = 0.9959$$
$$y_{S6}(t) = f\{ x_{28}(t), y(t\text{-}\delta t)\} \quad Q = 0.9741$$
$$y_{S6}(t) = f\{ x_5(t\text{-}2\delta t), x_{28}(t), y(t\text{-}\delta t)\} \quad Q = 0.8262$$
$$y_{S6}(t) = f\{ x_5(t\text{-}2\delta t), x_{28}(t), x_{42}(t\text{-}3\delta t), y(t\text{-}\delta t)\} Q = 0.6165$$

The result of simulating the output variable with the $y_{S2}(t)$ model of complexity 5 is presented in Figure 6-13. As it has been done with other simulations, 85% of the available data were used to derive the model, while the remaining 15% were used as validation data set. The left graph of Figure 6-13 shows 1080 points, corresponding to the full validation data set, of the output variable. The measurement data are shown as a continuous line, whereas the simulation results are presented as a dash-dotted line. The simulated output seems to follow the real data quite accurately, yet there are so many points on that graph that it is difficult to appreciate the difference between the two curves. The right graph of Figure 6-13 zooms in on the validation data set, limiting the shown data points to the range from 450 to 520. Here it is possible to better judge the quality of the prediction obtained with the FIR model. The model treats the small variations during the controlled periods as noise, i.e., essentially models their mean value. The spike representing a transient is simulated correctly, albeit with a short delay and with a reduced amplitude. The MSE error associated with this simulation is 0.1152.

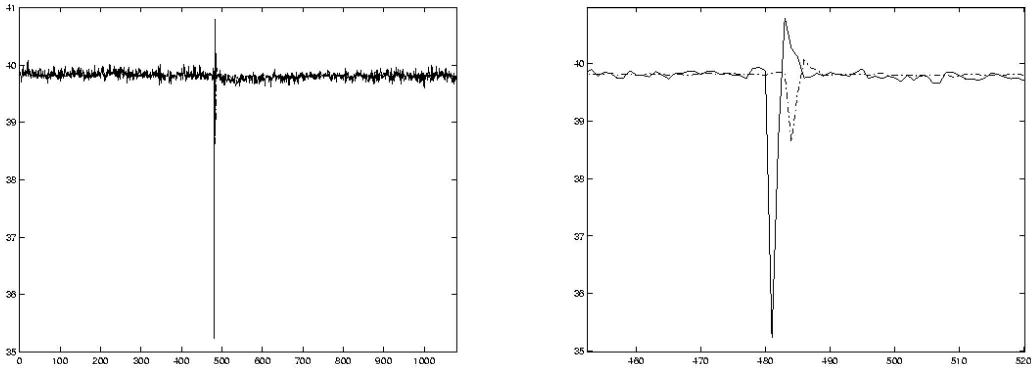


Figure 6-13 Gas turbine output simulation results using a complexity-5 FIR model derived from subsystem S2 taking into account the time variable.

If the model of complexity 4, derived from the information of the S2 subsystem, is used to simulate the output, a slightly inferior result from that provided in Figure 6-13 is found. The simulation results are shown in Figure 6-14. When comparing the two figures, the reader notices that the FIR predictions are almost identical. However, the new simulation shows a second smaller peak that does not correspond to a transient of the measurement data. Consequently, the MSE value for this new simulation is 0.1348, slightly higher than in the previous case.

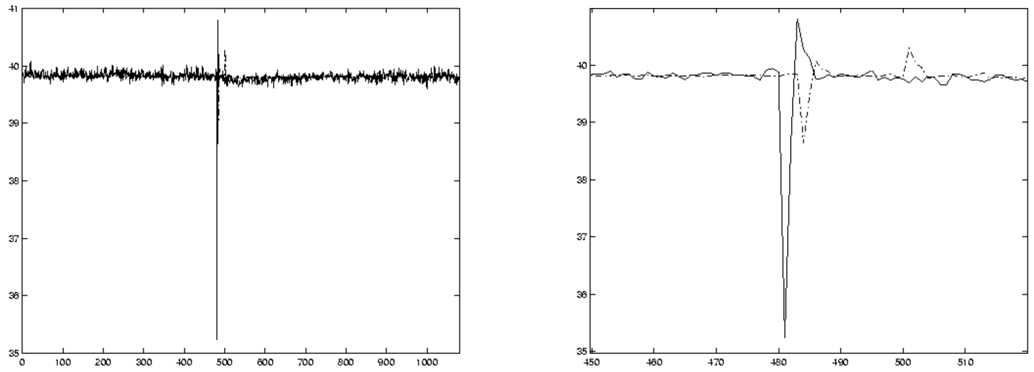


Figure 6-14 Gas turbine output simulation results using a complexity-4 FIR model derived from subsystem S2 taking into account the time variable.

There are two more simulations that should be evaluated: those obtained from the complexity-4 and complexity-5 models derived from the S6 subsystem. The complexity-4 model obtained from the S6 subsystem is exactly the same as that obtained from the S2 subsystem. Hence this simulation has already been presented in Figure 6-14.

For the complexity-5 model, the obtained simulation is depicted in Figure 6-15. As can be seen from this figure, the simulated output again follows well the behaviour of the real system. However, the delay of the simulated peak is slightly larger. The MSE value found in this case is 0.1271.

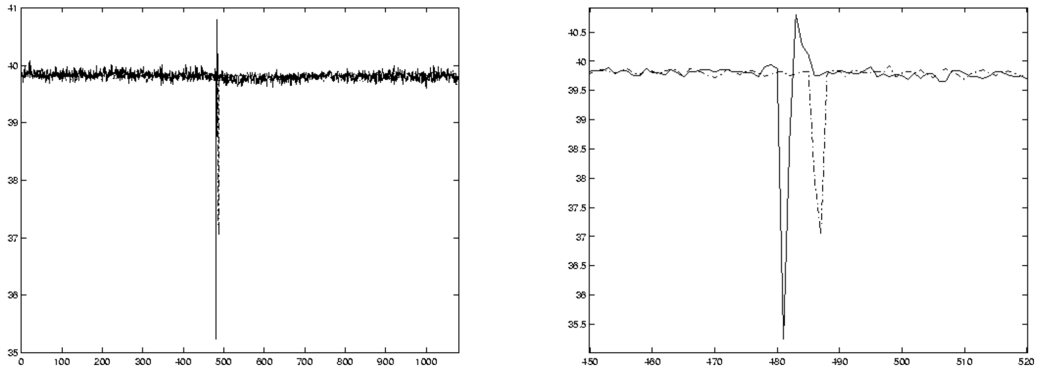


Figure 6-15 Gas turbine output simulation results using a complexity 5 FIR model derived from subsystem S6 having into account the time variable.

All of the FIR models that were simulated gave similar results. As can be seen from the presented figures, these results are quite satisfactory. Note that the used models are not necessary optimal; in fact, it is very likely that they are suboptimal. It would be very much of a coincidence if the optimal model had been found. This helps to understand the concepts underlying the envisaged suboptimal search strategy. The importance of optimality in a FIR model is relative. This in turns is logical, since we are dealing with qualitative models. Hence it is possible to obtain very good predictions without using the truly optimal (in a FIR sense: the one with the highest mask quality) model.

6.3.3 Dynamic FIR model of the gas turbine using energy information

A different point of view is taken when considering the variable trajectories as stochastic signals. As was described in Chapter 5, each observed variable trajectory can be seen as a superposition of two values, one measuring a desired physical characteristic, such as the fuel flow through a pipe, the other denoting a noise term, representing e.g. measurement noise or thermal noise. So each of these trajectories can be interpreted as a realisation of a stochastic process, i.e., there exists a deterministic as well as a random component in the observed trajectory. With this interpretation, the energy of the signals can be computed and used to determine, at which delays each input variable exhibits more energy relating to the output. This interpretation enables the modeller to obtain the most probable delays for which the input variables are to appear in a FIR qualitative model.

This method of obtaining the positions in which to place '-1' elements in the mask candidate matrix has been applied to the 64 variables chosen for the gas turbine system. The results obtained are summarised in Table 6-XIII, where the delays at which each input variable exhibits more energy relating to the output are shown. The integer numbers in the second column represent the number of sampling interval times that the signal is delayed with respect to time reference 0, that is, for example in row one, the numbers 9, 6, 3, 2 represent *$x_1(t-9)$*, *$x_1(t-6)$*, *$x_1(t-3)$*, and *$x_1(t-2)$*.

Variable	Delays with maximum energy related to the output variable
x_1, AFPBD	9, 6, 3, 2
x_2, AFPCD	9, 8, 7, 5, 3, 2
x_3, AFQ	9, 6, 5, 3
x_4, AFQD	11
x_5, ATID	85, 8, 7, 4, 3, 2
x_6, BB1	3
x_7, BB10	28, 26, 23, 21, 20, 16, 14, 13
x_8, BB11	43, 32, 28, 23, 20, 17, 16, 3, 2
x_9, BB12	43, 28, 23, 21, 16, 12, 3, 2
x_{10}, BB2	6, 4, 3, 2
x_{11}, BB4	10, 6, 5, 3, 2
x_{12}, BB5	5, 3, 2
x_{13}, BTGJ1	10, 9, 5, 4, 3, 2
x_{14}, BTGJ2	10, 6
x_{15}, CMHUM	14, 12, 10, 6, 4, 3
x_{16}, CPD	64, 37, 6, 4, 3
x_{17}, CTD	64, 37, 6, 5, 3, 2
x_{18}, CTDA1	64, 37, 4, 3
x_{19}, CTDA2	64, 37, 3
x_{20}, CTIF1	32, 28, 9, 7, 3
x_{21}, CTIF2	8, 5, 4, 3
x_{22}, DF	13, 12, 11, 5, 3
x_{23}, DTGGC10	7, 6, 5, 4, 3, 2
x_{24}, DTGGC11	37, 6, 5, 3
x_{25}, DTGGH18	6, 5, 3, 2
x_{26}, DTGGH19	10, 9, 7, 5, 4, 3, 2
x_{27}, DV	28, 26, 23, 22, 16, 14, 13, 12, 11, 2
x_{28}, FQG	64, 43, 37, 28, 23, 14, 13, 5
x_{29}, FQLM1	3
x_{30}, SDSJ1	3, 2

x_{31}, SDSJ2	6, 4, 3, 2
x_{32}, SPSJ1	7, 4, 3, 2
x_{33}, STSJ	9, 8, 7, 6, 4, 3
x_{34}, SVL	21, 20, 18
x_{35}, TGSDIFU1	10, 5, 2
x_{36}, TGSDIFU2	10, 5, 4, 2
x_{37}, TGSDIFV1	5, 4, 3, 2
x_{38}, TGSDIFV2	10, 4, 2
x_{39}, TGSDIFW1	11, 9, 2
x_{40}, TGSDIFW2	12, 11, 10, 5
x_{41}, TIFDP1	16, 13, 6, 4, 3
x_{42}, TIFDP2	6, 4, 3, 2
x_{43}, TIFDP3	32, 16, 6, 4, 3, 2
x_{44}, TNH	13, 12, 11, 5
x_{45}, TTWS1AO1	8, 6, 4, 3, 2
x_{46}, TTWS1AO2	6, 5, 4, 3, 2
x_{47}, TTWS1FI1	6, 4, 2
x_{48}, TTWS1FI2	8, 6, 5, 3, 2
x_{49}, TTWS1FO1	13, 5, 4, 2
x_{50}, TTWS1FO2	9, 5, 4, 3, 2
x_{51}, TTWS2AO1	64, 51, 37, 20, 3, 2
x_{52}, TTWS2AO2	64, 51, 37, 11, 10, 9, 2
x_{53}, TTWS2FO1	7, 6, 4, 3
x_{54}, TTWS2FO2	6, 5, 4, 2
x_{55}, TTWS3AO1	11, 9, 5, 4, 2
x_{56}, TTWS3AO2	11, 9, 5, 3, 2
x_{57}, TTWS3FO1	13, 12, 11, 10, 9, 4
x_{58}, TTWS3FO2	9, 5, 4, 2
x_{59}, TTXM	37, 4, 3
x_{60}, TTXSP1	4, 3, 2
x_{61}, TTXSP2	4, 3, 2
x_{62}, TTXSP3	3, 2
x_{63}, WQJ	8, 4, 3, 2
y, DWATT	85, 64, 51, 43, 37, 32, 28, 26, 23, 21, 20, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2

Table 6-XIII Delays with maximum energy related to the output variable.

Notice that delays 0 and 1 do not show up in Table 6-XIII, because the method employed cannot resolve these delays. They need to be treated separately.

Prior to using this delay information to generate a mask candidate matrix, it is necessary to decide the number of delays to allow, i.e., to decide, which mask depth is necessary to cover the system dynamics. People who worked with the gas turbine plant within the European project mentioned in the introduction advised to use a number of delays equal to 30. If variable information up to delay 30, i.e., $x_i(t\text{-}30)$, is to be used, a depth-31 candidate mask would need to be proposed to FIR in order to find a model of the system.

Let us discuss the case in which 30 delays are considered. A 31 x 64 matrix is filled with '-1' elements in the positions corresponding to the delays (up to 30) listed in Table 6-XIII so as to form a mask candidate matrix. The element (31, 64) of this candidate mask is set to '+1' denoting the variable considered as the output of the system, i.e., the variable that is to be modelled[3].

Using this information, the resulting mask candidate matrix contains a total of 287 out of 1984 possible '-1' elements. Moreover, it must be taken into account that the energy method does not say anything about delays 0 and 1, so the total number of '-1' entries of the candidate mask is raised to a maximum of 287 + (64 + 63) = 414 '-1' elements. An estimation of the time needed to compute the FIR model of a 414 and a 1984 '-1' element mask candidate matrix can be made[4]. Table 6-XIV lists the number of models to compute for both masks and for each allowed model complexity. The total number of models that have to be computed when a mask candidate matrix with *n* elements set to '-1' and a maximum allowed complexity of *c*, is given by the formula:

$$\begin{bmatrix} n \\ 1 \end{bmatrix} + \begin{bmatrix} n \\ 2 \end{bmatrix} + \cdots + \begin{bmatrix} n \\ c-1 \end{bmatrix}$$

For example, if a candidate mask is used that has 191 '–1' elements and the maximum allowed complexity is 5, the number of models to be computed is:

$$\begin{bmatrix} 191 \\ 1 \end{bmatrix} + \begin{bmatrix} 191 \\ 2 \end{bmatrix} + \begin{bmatrix} 191 \\ 3 \end{bmatrix} + \begin{bmatrix} 191 \\ 4 \end{bmatrix} = 191 + 18'145 + 1'143'135 + 53'727'345 = 54'888'816$$

There are 191 models of complexity 2, 18'145 models of complexity 3, and so on, leading to a total number of models of almost 55 millions.

FIR model complexity	Number of models for a 414 '-1' elements candidate mask	Number of models for a 1984 '-1' elements candidate mask
2	414	1984
3	85'491	1'967'136
4	11'740'764	1'299'621'184
5	1'206'363'501	643'637'391'376

Table 6-XIV Number of models to compute for a candidate mask containing 414 and 1984 '-1' elements respectively.

In Table 6-XV, the total number of models to be computed (considering up to complexity 5), an estimation of the model computation time, and the reduction of computation achieved with the energy method are provided.

[3] In the referenced European project, a model using only 5 time delays was used. In Section 6.3.2.2, a depth-6 FIR model was derived using a subsystem decomposition technique. In order to provide a fair comparison between the two techniques, a depth-6 FIR model shall be adopted here also. For the suggested depth-31 model, considering 30 delay times, a discussion of how a corresponding mask candidate matrix could be derived is provided as well, since the author considers this discussion to be of interest.

[4] This estimation has been made using information gained by computing FIR models with the gas turbine system later in the same section. The SUN ULTRASPARC II workstation used for the calculations was able to compute 1727 models per minute.

Number of '-1' elements	Total number of models to be computed	Time estimation for computing these models	Achieved time reduction
1984	644'938'981'680	711 years	-
414	1'218'190'170	1 year and 125 days	99.9811%

Table 6-XV Estimation of the computing time needed for candidate masks with 414 and 1984 '-1' elements, respectively.

From Table 6-XV, we can see the estimation of time needed to obtain (sub)optimal FIR depth-30 models for the gas turbine. If the FIR model of the studied system were to be computed without any pre-analysis at all, about 711 years would be needed to accomplish this, which is impractical. About thirty generations of people would be needed to find such a model, and, after such a long time, surely the gas turbine would have become inoperable. Using the energy method presented in Chapter 5, a dramatic reduction on the FIR model computation time is achieved. Yet, this reduction it is insufficient, as it is also unrealistic to assume that the workstation used all along the dissertation would have its electric power uninterrupted during almost 1.5 years. Precisely, while the author was writing these lines, a strong autumn storm over Barcelona interrupted the electric power of the laboratory for a few minutes, which caused a modelling effort requiring two hours of computing time to be lost.

One way to further reduce the computation time of the gas turbine FIR model is to use, for the first row of the proposed candidate mask, i.e., the row corresponding to the zero delay, information gained by a correlation analysis. This correlation analysis has already been performed for the gas turbine system in the previous subsection. Using this information, the delay-0 row of the depth-30 candidate mask contains only 20 '-1' elements out of 63, which reduces the number of mask inputs to the proposed candidate mask from 414 to 371. This reduction in the number of possible *m*-inputs to the model reduces also the estimation of the computation time from 1 year and 125 days to 315 days, almost half a year less. Yet, this reduction is still insufficient to compute a model for the gas turbine system within an acceptable amount of time.

A further reduction on the computation time of this model could be obtained if the algorithm described in Chapter 2, Section 2.4.1, were combined with the results obtained with the energy method. With this procedure, and in a first step, the most important qualitative relations for the first delay, i.e., the second row of the candidate matrix, now all filled with -1 elements, can be reduced.

This algorithm, as explained in Chapter 2, permits to propose candidate masks of increasing depths. *N* candidate masks are proposed such that its total computation time is much less than if a unique depth-*N* candidate mask were proposed to the FIR exhaustive search model algorithm. Let us resume the functioning of the suboptimal search algorithm proposed in Chapter 2. This algorithm reduces the total number of models to compute by reducing the number of '-1' elements in the mask candidate matrix. In the situation at hand, since information about delay-0 has already been obtained, it can start with a mask of depth *d=2*, in which the entire second row of the candidate mask is set to '–1'. In this way, a depth-2 candidate mask is proposed that has a total of 84 '-1' elements. For this candidate mask, the number of models to be calculated is offered in Table 6-XVI.

Model complexity	C2	C3	C4	C5	Total
Number of models	84	3.486	95.284	1.929.501	2.028.355

Table 6-XVI Number of models to be computed for each allowed complexity with a 84 '-1' element candidate mask.

This computation took exactly 19.5 hours on the SUN workstation. After an analysis[5] of all the computed masks using the algorithm provided in Chapter 2 with the specified parameters, those qualitative relations that are most strongly related to the output are identified. In Table 6-XVII such relations are shown. In this table, only those variables used by the selected masks are listed, in order not to generate large tables listing all 64 variables each time. The complexity-2 models are not listed since they do not have crucial importance for the study; yet, the maximum quality found for those kinds of models was $Q_{best} = 0.9959$, and the number of selected models, complying with the restrictions of the algorithm, is 23.

For each complexity separately, only those inputs are considered *significant inputs* that are present in at least the 10% of the good masks of that complexity. In Table 6-XVII, those significant inputs are enhanced in bold. Note that of all the possible *m*-inputs in the previously proposed candidate mask, corresponding to all of the 64 physical variables, the algorithm only chooses 10 *m*-inputs (of only 6 physical variables) that will form part of the next depth-3 candidate mask. For this new candidate mask, the elements that must be set to '-1' in the first two rows, i.e., delays 0 and 1, are determined by the previous experiment. Those are columns (variables) 5, 9, 22, 27, 44, and 60 for the delay-1 row, and columns (variables) 5, 9, 27, and 60 for the delay-0 row.

This technique could be combined with the depth-30 energy model search, providing the '-1' entries in the bottom two rows of the mask candidate matrix. Hence the number of '-1' entries in the subsequent depth-30 mask candidate matrix is now 287 + 10 = 297. This reduces the computing time of the depth-30 suboptimal mask to 129.5 days, which is still too long, taking into consideration the reliability of the electric circuits at this university.

What if the energy method were not used at all, and instead, the hill-climbing technique proposed in Chapter 2 would be continued all the way to depth 30? At each step of the algorithm, a similar number of important inputs, in the order of 10 to 25 are retained. Thus, each step is of similar computational complexity, i.e., takes about one day of computing time. Thus, the entire algorithm could be completed within roughly 30 days, which is still unpleasant, but maybe feasible.

Of course, a better approach would be to increase the threshold parameter that distinguishes a significant power peak from an insignificant one. In this way, the number of '-1' elements of the energy method can be reduced easily and quickly, and a suboptimal mask can be obtained within a few hours of computing time.

[5] This analysis means to look for masks with a quality equal or greater than 97.5 % of the best found quality, and then identify those variables that have been used by at least 10% of those masks. These operations are performed using user-programmed codes in Matlab. The analysis time needed for this operation adds up to about 0.5 hours for each performed run, which must be taken into account in the total used model computation time.

	Complex. 3 (Q_{best}= 0.9912) $Q>0.975Q_{best}$ masks – 58		Complex. 4 (Q_{best}= 0.8342) $Q>0.975Q_{best}$ masks – 117		Complex. 5 (Q_{best}= 0.5934) $Q>0.975Q_{best}$ masks - 4	
Delay	0	1	0	1	0	1
x_5	5	5	**59**	**77**	**4**	**4**
x_9	—	—	4	4	**2**	**2**
x_{17}	2	2	—	—	—	—
x_{18}	—	2	—	—	—	—
x_{22}	—	—	—	**36**	—	—
x_{23}	—	2	—	—	—	—
x_{24}	—	2	—	1	—	—
x_{27}	—	—	4	4	**2**	**2**
x_{28}	4	2	6	6	—	—
x_{44}	—	—	—	**36**	—	—
x_{45}	—	2	—	6	—	—
x_{46}	—	2	—	6	—	—
x_{47}	—	2	—	6	—	—
x_{48}	2	2	6	6	—	—
x_{49}	—	2	—	6	—	—
x_{50}	—	2	—	6	—	—
x_{51}	—	2	—	1	—	—
x_{52}	—	2	—	5	—	—
x_{53}	—	2	—	6	—	—
x_{54}	—	4	—	6	—	—
x_{55}	—	2	—	6	—	—
x_{56}	4	4	6	6	—	—
x_{57}	—	2	—	6	—	—
x_{58}	—	2	—	6	—	—
x_{60}	**24**	**24**	**18**	—	—	—
y	58	4	117	6	4	—

Table 6-XVII Results from the analysis of all the computed masks, when a depth-2 candidate mask with 84 '-1' elements is proposed to FIR. Enhanced in bold are the *m*-inputs to the next depth-3 candidate mask.

Instead of continuing with this hypothetical (and rather depressing) computational analysis, the discussion shall now focus on the computation of a depth-6 model using the energy-based method. The bottom two rows are, of course, computed as outlined above.

For the third row of the new candidate mask, the $(t\text{-}2\delta t)$ row, the elements that must be set to '-1' are given by the energy-based information shown in Table 6-XIII, so combining the hill-climbing approach with the energy method in order to simplify the model search problem. Those elements are columns (variables) 1, 2, 5, 8, 9, 10, 11, 12, 13, 17, 23, 25, 26, 27, 30, 31, 32, 35, 36, 37, 38, 39, 42, 43, 45, 46, 47, 48, 49, 50, 51, 52, 54, 55, 56, 58, 60, 61, 62, 63 and 64. In this list, column 64 stands for the considered output variable two time steps in the past.

The new depth-3 mask candidate matrix contains 51 out of 191 possible '-1' elements. An optimal model search is now performed, proposing this mask candidate matrix to FIR. This time, the total number of models to be computed (considering models up to complexity 5) was 272.051, and it took 2.5 hours to compute them all. Table 6-XVIII shows the results of this model search. In this table, only those variables used in the selected masks are shown. This is the reason why this new table does not contain exactly the same variables as the previous one. In the given case, variables 18, 23, 24, 28, 53 and 57 that were used in the depth-2 models are no longer selected in the depth-3 models. The qualities of the best masks of each complexity are only slightly higher than in the case of the depth-2 models. 65 good masks of complexity 3, 66 good masks of complexity 4, and 27 good masks of complexity 5 were found. At this point, FIR shows preferences for some variables one time-step back over others. The reason is that FIR now has more choices, and often prefers the same variable two time-steps back.

	C3 (Q_{best}= 0.9912) selected masks 65			C4 (Q_{best}= 0.8457) selected masks 66			C5 (Q_{best}= 0.6142) selected masks 27		
Delay	0	1	2	0	1	2	0	1	2
x_5	**7**	**7**	6	**21**	**22**	**31**	**18**	**18**	**18**
x_9	—	—	—	—	—	—	**9**	**9**	**9**
x_{17}	—	—	3	—	—	2	—	—	—
x_{22}	—	—	—	—	**7**	—	—	—	—
x_{27}	—	—	—	—	—	—	**9**	**9**	**9**
x_{44}	—	—	6	—	6	—	—	—	—
x_{45}	—	—	6	—	—	4	—	—	—
x_{46}	—	—	5	—	—	5	—	—	—
x_{47}	—	—	6	—	—	4	—	—	—
x_{48}	—	—	3	—	—	4	—	—	—
x_{49}	—	—	3	—	—	4	—	—	—
x_{50}	—	—	3	—	—	4	—	—	—
x_{51}	—	—	3	—	—	—	—	—	—
x_{52}	—	—	3	—	—	1	—	—	—
x_{54}	—	—	6	—	—	**12**	—	—	—
x_{55}	—	—	3	—	—	4	—	—	—
x_{56}	—	—	6	—	—	**12**	—	—	—
x_{58}	—	—	6	—	—	4	—	—	—
x_{60}	**15**	**15**	**15**	**23**	**11**	**12**	—	—	—
y	65	—	6	66	—	5	27	—	—

Table 6-XVIII Results from the analysis of all computed masks when a depth-3 candidate mask with 51 '-1' elements is proposed to FIR. Enhanced in bold are the *m*-inputs to the next depth-4 candidate mask.

The way the algorithm is implemented, once an input variable at a certain time delay has been eliminated from one of the mask candidate matrices, it will never show up again in any of the subsequent mask candidate matrices. Therefore, the *discriminator* value (in the turbine gas system analysed here it is set to the 10%) must be chosen carefully, in order not to eliminate potentially useful inputs too early. A smaller discriminator might generate a better suboptimal mask at the end, but this goes at the expense of having to evaluate more masks in the process. A higher discriminator value leads to a faster search, but may result in a suboptimal mask of lower quality.

Again, those inputs that are selected to inhabit the next depth-4 candidate mask are enhanced in bold in Table 6-XVIII. These inputs will form the first three rows of the new proposed candidate mask. This mask candidate matrix will have, in the row corresponding to the 0 delay, columns (variables) 5, 9, 27, and 60 set to '–1'. For the row (t-δt), the columns 5, 9, 22, 27, and 60 should be set to '-1', so as to denote possible mask inputs to the next FIR models to be computed. Row (t-$2\delta t$) has '-1' elements in columns 5, 9, 27, 54, 56, and 60. Finally for the (t-$3\delta t$) row, the information which elements should be set to '-1' is gathered from Table 6-XIII. Those elements are columns (variables): 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 29, 30, 31, 32, 33, 37, 41, 42, 43, 45, 46, 48, 50, 51, 53, 56, 59, 60, 61, 62, 63 and 64.

As in the list of the previous candidate mask, column 64 stands for the considered output variable, now three time steps back. Although this has not been explicitly mentioned so far, for a candidate mask with an actual depth of *N*, the element (*N*, 64) is always set to '+1', denoting the position of the *m*-output within the FIR qualitative model.

The new depth-4 mask candidate matrix contains 61 out of 255 possible '-1' elements. In each step of the algorithm, a candidate mask that has its depth increased by 1 with respect to the previous step is proposed. Each time, as can be stated with the number of '-1' elements that inhabit the candidate mask, the relative reduction in the number of models to compute is increased.

An optimal model search is now performed, proposing the newly computed mask candidate matrix to FIR. This time, the total number of models to compute (considering models up to complexity 5) was 559.736, and it took 4 hours and 58 minutes to calculate them all. Table 6-XIX shows the results of this new model search.

	C3 (Q_{best}= 0.9912) selected masks 82				C4 (Q_{best}= 0.8557) selected masks 103				C5 (Q_{best}= 0.6425) selected masks 26			
Delay	0	1	2	3	0	1	2	3	0	1	2	3
x_5	6	7	5	6	**20**	**32**	**29**	**34**	**20**	**7**	**6**	**16**
x_9	—	—	—	—	—	—	—	—	**5**	**4**	**8**	**7**
x_{17}	—	—	—	4	—	—	—	3	—	—	—	—
x_{18}	—	—	—	4	—	—	—	—	—	—	—	—
x_{19}	—	—	—	4	—	—	—	—	—	—	—	—
x_{22}	—	—	—	—	—	**10**	—	**10**	—	—	—	—
x_{23}	—	—	—	4	—	—	—	—	—	—	—	—
x_{24}	—	—	—	4	—	—	—	1	—	—	—	—
x_{27}	—	—	—	—	—	—	—	—	**8**	**7**	**8**	—
x_{45}	—	—	—	5	—	—	—	6	—	—	—	—
x_{46}	—	—	—	7	—	—	—	8	—	—	—	—
x_{48}	—	—	—	4	—	—	—	8	—	—	—	—
x_{50}	—	—	—	4	—	—	—	8	—	—	—	—
x_{51}	—	—	—	4	—	—	—	—	—	—	—	—
x_{53}	—	—	—	7	—	—	—	8	—	—	—	—
x_{54}	—	—	7	—	—	—	**19**	—	—	—	—	—
x_{56}	—	—	**8**	**8**	—	—	**19**	**11**	—	—	—	—
x_{60}	**15**	**14**	**14**	**15**	**24**	**19**	**18**	**12**	—	—	—	—
y	82	—	—	**10**	103	—	—	9	26	—	—	**8**

Table 6-XIX Results from the analysis of all computed masks when a depth-4 candidate mask with 61 '-1' elements is proposed to FIR. Enhanced in bold are the *m*-inputs to the next depth-5 candidate mask.

The qualities of the best masks of each complexity increased a little more with respect to the case of the depth-3 models. 82 good masks of complexity 3, 103 good masks of complexity 4, and 26 good masks of complexity 5 were found. Again in Table 6-XIX, only those variables used in the depth-4 selected masks are shown, and so, the rows differ form those of the previously presented tables. In this particular case, variables 18, 19, 23, 24, and 53 appears (for example, variable 19 has not been used by either the depth-2 models or the depth-3 ones) or re-appears (as variables 23 and 24, not used in the previous simulation, but used in the depth-2 models). In the other hand, variables 44, 47, 49, 52, 55 and 58 that were used in the depth-3 models are no longer selected in the depth-4 models.

Note, also, that in this simulation the self-output at delay (*t-3δt*) is selected to form part of the next candidate mask, up to now, no autoregressive information had been selected.

In Table 6-XIX, those inputs that the algorithm selects to inhabit the next depth-5 candidate mask are enhanced in bold. This mask will have the columns (variables) 5, 9, 27, and 60 set to '-1' for the delay 0 row. In the row corresponding to (*t-δt*), columns (variables) 5, 9, 22, 27, and 60 should be set to '–1'. For the row (*t-2δt*), the columns 5, 9, 27, 54, 56, and 60 should be set to '-1'. Row (*t-3δt*) has '-1' elements in columns 5, 9, 22, 56, 60, and 64. Finally for the (*t-4δt*) row, the information which elements should be set to '-1' is taken from Table 6-XIII. Those elements are columns (variables) 5, 10, 13, 15, 16, 18, 21, 23, 26, 31, 32, 33, 36, 37, 38, 41, 42, 43, 45, 46, 47, 50, 53, 54, 55, 57, 58, 59, 60, 61, 63 and 64.

This new proposed mask candidate matrix, actually with a depth of 5, contains 54 out of 319 possible '-1' elements. In this step of the algorithm, the relative reduction in the number of models to compute is considerable. An optimal model search is now performed, proposing this mask candidate matrix to FIR. For this particular experiment, the total number of models to be computed (considering models up to complexity 5) was 342.540, and it took 3 hours and 18 minutes to calculate them. Results of this model search are provided in Table 6-XXI. Note that, once again, some of the variables that were encountered in the selected masks of the depth-4 models are not used in the depth-5 models. Those are variables 17, 19, 24, and 51. Conversely, variables 47, 49, 55, 57, and 58 are found in the depth-5 selected masks, variables that were not used by the depth-4 models.

Once again the overall qualities of the computed models increased slightly with respect to the last model computation. This suggests proposing a new depth-6 candidate mask using the information contained in Table 6-XXI and the information given by the cross-spectrum method. Those inputs that the algorithm selects to inhabit the next depth-6 candidate mask are enhanced in bold. This mask has the columns (variables) 5, 9, 27, and 60 set to '-1' for the delay 0 row. In the row corresponding to (*t-δt*), columns (variables) 5, 9, 27, and 60 should be set to '–1'. For the row (*t-2δt*), the columns 5, 9, 27, 54, 56, and 60 should be set to '-1'. Row (*t-3δt*) has '-1' elements in columns 5, 9, 60, and 64. Row (*t-4δt*) has '-1' elements in columns 5 and 60. Finally for the (*t-5δt*) row, the information which elements should be set to '-1' is taken from Table 6-XIII. Those elements are columns (variables) 2, 3, 4, 11, 12, 13, 17, 22, 23, 24, 25, 26, 28, 35, 36, 37, 40, 44, 46, 48, 49, 50, 54, 55, 56, 58 and 64.

	C3 (Q_{best}= 0.9912) selected masks 122					C4 (Q_{best}= 0.8657) selected masks 266					C5 (Q_{best}= 0.7025) selected masks 42				
Delay	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
x_5	7	7	7	9	7	**63**	**50**	**56**	**55**	**80**	**20**	**20**	**10**	**16**	**15**
x_9	—	—	—	—	—	—	—	—	—	—	**8**	**8**	**11**	**11**	—
x_{18}	—	—	—	—	5	—	—	—	—	—	—	—	—	—	—
x_{22}	—	—	—	—	—	—	12	—	12	—	—	—	—	—	—
x_{23}	—	—	—	—	5	—	—	—	—	1	—	—	—	—	—
x_{27}	—	—	—	—	—	—	—	—	—	—	**14**	**14**	**12**	—	—
x_{45}	—	—	—	—	5	—	—	—	—	14	—	—	—	—	—
x_{46}	—	—	—	—	5	—	—	—	—	14	—	—	—	—	—
x_{47}	—	—	—	—	5	—	—	—	—	14	—	—	—	—	—
x_{49}	—	—	—	—	5	—	—	—	—	13	—	—	—	—	—
x_{50}	—	—	—	—	5	—	—	—	—	13	—	—	—	—	—
x_{53}	—	—	—	—	10	—	—	—	—	13	—	—	—	—	—
x_{54}	—	—	10	—	10	—	—	**29**	—	14	—	—	—	—	—
x_{55}	—	—	—	—	6	—	—	—	—	13	—	—	—	—	—
x_{56}	—	—	10	10	—	—	—	**29**	18	—	—	—	—	—	—
x_{57}	—	—	—	—	5	—	—	—	—	15	—	—	—	—	—
x_{58}	—	—	—	—	10	—	—	—	—	17	—	—	—	—	—
x_{60}	**17**	**17**	**17**	**17**	**17**	**57**	**39**	**44**	**32**	**32**	—	—	—	—	—
y	122	—	—	1	1	122	—	—	**27**	—	42	—	—	**9**	—

Table 6-XXI Results from the analysis of the computed masks when a depth-5 candidate mask with 54 '-1' elements is proposed to FIR. Enhanced in bold are the *m*-inputs to the next depth-6 candidate mask.

The new depth-6 proposed mask candidate matrix contains 47 out of 383 possible '-1' elements. As outlined before, as the algorithm proceeds while increasing the mask depth, a higher relative reduction in the number of models to be computed is achieved. As with the other proposed candidate masks, the new depth-6 candidate mask is proposed to FIR, and an optimal model search is performed. Now, the total number of models to be computed (considering models up to complexity 5) was 195.713, and it took 1 hour and 53 minutes to calculate them. Results of this model search are provided in Table 6-XXII. Note that, once again, some of the variables that were encountered in the selected masks for the depth-5 models are not used in the depth-6 models. Those are variables 17, 19, 24, and 51. Conversely, variables 47, 49, 55, 57, and 58 have been found in the depth-6 selected masks that were not used by the depth-5 models.

Results from this last model computation are interesting. First, the overall quality of the masks incurred a smaller increase than in the previous cases, although it still grew somewhat. Second, FIR rather preferred to choose primarily variables with delays < 5, as can be seen by looking at Table 6-XXII. This fact may indicate that the maximum mask quality that the search algorithm can find has been reached. Yet, in order to better assess the results, a new depth-7 candidate mask should be proposed using the elements enhanced in bold in Table 6-XXII as well as the information provided in Table 6-XIII. This mask will have the columns (variables) 5, 9, 27, and 60 set to '-1' for rows corresponding to delay 0 and (*t-δt*). For the row (*t-2δt*), the columns 5, 9, 27, 54, 56, and 60 should be set to '-1'. Row (*t-3δt*) has '-1' elements in columns 5, 9, 60, and 64. Row (*t-4δt*) has '-1' elements in columns 5 and 60. Row (*t-5δt*) has a unique '-1' element in column 28. Finally for the (*t-6δt*) row, the energy-based method proposes to set '-1' elements in columns 1, 3, 4, 10, 11, 14, 15, 16, 17, 23, 24, 25, 31, 33, 41, 42, 43, 45, 46, 47, 48, 53, 54, and 64.

	C3 (Q_{best}= 0.9912) selected masks 92						C4 (Q_{best}= 0.8658) selected masks 215						C5 (Q_{best}= 0.7127) selected masks 42					
Delay	0	1	2	3	4	5	0	1	2	3	4	5	0	1	2	3	4	5
x_5	3	5	2	5	2	—	**51**	**40**	**48**	**45**	**66**	—	**20**	**22**	**11**	**19**	**15**	—
x_9	—	—	—	—	—	—	—	—	—	—	—	—	**9**	**9**	**11**	**10**	—	—
x_{17}	—	—	—	—	—	5	—	—	—	—	—	3	—	—	—	—	—	—
x_{22}	—	—	—	—	—	—	—	—	—	—	—	10	—	—	—	—	—	—
x_{23}	—	—	—	—	—	5	—	—	—	—	—	2	—	—	—	—	—	—
x_{24}	—	—	—	—	—	5	—	—	—	—	—	—	—	—	—	—	—	—
x_{27}	—	—	—	—	—	—	—	—	—	—	—	—	**12**	**13**	**11**	—	—	—
x_{28}	—	—	—	—	—	**9**	—	—	—	—	—	**22**	—	—	—	—	—	—
x_{44}	—	—	—	—	—	—	—	—	—	—	—	10	—	—	—	—	—	—
x_{46}	—	—	—	—	—	4	—	—	—	—	—	14	—	—	—	—	—	—
x_{48}	—	—	—	—	—	4	—	—	—	—	—	11	—	—	—	—	—	—
x_{49}	—	—	—	—	—	4	—	—	—	—	—	11	—	—	—	—	—	—
x_{50}	—	—	—	—	—	4	—	—	—	—	—	13	—	—	—	—	—	—
x_{54}	—	—	**10**	—	—	7	—	—	**29**	—	—	14	—	—	—	—	—	—
x_{55}	—	—	—	—	—	7	—	—	—	—	—	13	—	—	—	—	—	—
x_{56}	—	—	**10**	—	—	5	—	—	**24**	—	—	16	—	—	—	—	—	—
x_{58}	—	—	—	—	—	8	—	—	—	—	—	16	—	—	—	—	—	—
x_{60}	**15**	**15**	**15**	**15**	**15**	—	**42**	**35**	**33**	**24**	**26**	—	—	—	—	—	—	—
y	92	—	—	—	—	5	215	—	—	**22**	—	—	42	—	—	**11**	—	—

Table 6-XXII Results from the analysis of the computed masks when a depth-6 candidate mask with 47 '-1' elements is proposed to FIR. Enhanced in bold are the *m*-inputs to the next depth-7 candidate mask.

This new proposed candidate mask implies computing a total of 149.984 models (again considering models up to complexity five). The process took 1 hour and 12 minutes to complete. In this experiment, the overall quality of the masks did not increase from the quality values found in the depth-6 models, so the termination condition for the search algorithm is reached, and the model computation finishes here. Note that the used search algorithm suggests an optimal depth for the FIR models of 6, i.e., considering variables delayed up to five time steps, as some models of the referenced European project, from where the data were borrowed, suggested. If the search is not stopped here, and additional candidate masks of depth 8, 9, or 10 are proposed, the found mask qualities are seen to remain unchanged. Results from these last experiments are not shown here because they do not add any new or valuable information to the gas turbine model computation.

The computed models can now be used to simulate the turbine system. The highest quality is found for depth-6 models that are constructed using variables x_5, x_9, x_{17}, x_{22}, x_{23}, x_{24}, x_{27}, x_{28}, x_{44}, x_{46}, x_{48}, x_{49}, x_{50}, x_{54}, x_{55}, x_{56}, x_{58}, x_{60} and the output variable itself at various delays. From these variables, as can be seen from Table 6-XXII, only variables x_5, x_9, x_{27}, x_{28}, x_{54}, x_{56} and x_{60} are used in more of the 10% of the selected masks.

Let us simulate the system with the best complexity-5 model found, i.e., the one with quality 0.7127 and a mask depth of 6. was outlined in the beginning of this section, 85% of the available data have been used to generate the FIR qualitative model, while the remaining 15% are used to validate the model.

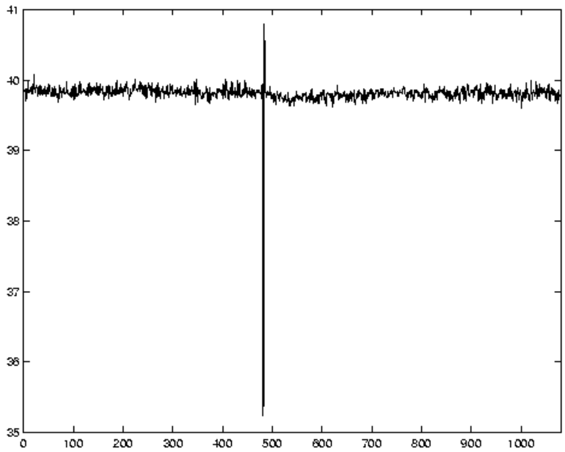


Figure 6-16 Real gas turbine output, 1080 points (15% of the available data).

In Figure 6-16, the entire validation data set of the considered output variable is depicted. These data correspond to a normal operating condition of the electric power generation plant. It can be seen that the generator load is almost constantly at 40 MW, except for one short oscillation.

The suboptimal model found by the energy-based algorithm is given by:

$$y(t) = f\{\, x_5(t-3\delta t),\, x_9(t-3\delta t),\, x_{27}(t-2\delta t),\, y(t-3\delta t) \,\}$$

which means that the generator load, y, can be modelled from the air temperature in the compressor, x_5, a measure of the vibration in the shaft that transmit the mechanical movement from the turbine to the electric generator, x_9, a measure of the generator voltage, x_{27}, and a delayed version of the output itself. Interestingly enough, $y(t-\delta t)$ is not used by the model. Instead, the model prefers to work with $y(t-3\delta t)$. The model found seems to be a quite plausible model, since three important factors are taken into account. First the temperature of the air in the compressor is directly related to the fuel consumption, which in turn is directly related to the electric power that the system is able to generate. Second, the shaft vibration measures the efficiency of the energy transmission between different parts of the whole system; if vibration in the shaft is important, a substantial amount of energy is being lost to produce such a vibration. Last, the voltage of the generator is directly related to the generated power. Therefore, using the combination of three methods, the energy based method introduced in Section 5.3.2, the hill-climbing algorithm discussed in Chapter 2, and the correlation analysis described in Chapter 3, a FIR model has been computed in a reasonable time. Table 6-XXIII summarizes the number of computed models as well as the computing time needed to obtain the final model. A time of 30 minutes, needed to compile the information related to the computed FIR models, has been added for each of the model computations, in order to provide a fair time calculation.

Depth	Number of computed models	Computing time	Model processing time
2	2.028.355	19h	30 min
3	272.051	2h 30min	30 min
4	559.736	4h 58min	30 min
5	342.540	3h 18min	30 min
6	195.713	1h 53min	30 min
7	149.984	1h 12min	-
Total	3.548.379	35h 21min	

Table 6-XXIII Total computation required to obtain a depth-6 gas turbine model using the energy-based method described in this research work.

The numbers given in this last table may be compared to those given in Table 6-XV. The achieved simplification is evident. Here, a model has been computed in approximately 35.5 hours, whereas the best option before was to spend 1 year and 125 days to get the submodel, or 711 years if a full candidate mask were to be used. The model obtained in this way is not necessary the best model that can be found. It is a suboptimal model, yet one that is based on engineering common sense, and that is expected to lead good simulation results.

In Figure 6-17, the measured validation data set, represented by a continuous line, is overlaid with the simulated output when using the proposed complexity-5 model, shown by a dash-dotted line. At a first glance, the model seems to mimic quite well the behaviour of the system, even in the oscillation present between data points 400 and 500 of the given figure. Only two important simulation errors occur between data points 500 and 600. In Figure 6-18, the previous graph is zoomed in, containing only the data points from 450 to 580, so as to show more clearly the details of the simulated data set relative to the measured data set.

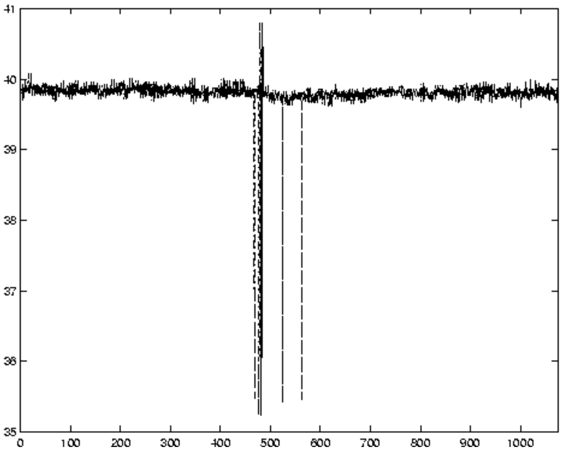


Figure 6-17 Real (continuous line) and simulated (dotted line) gas turbine output variable.

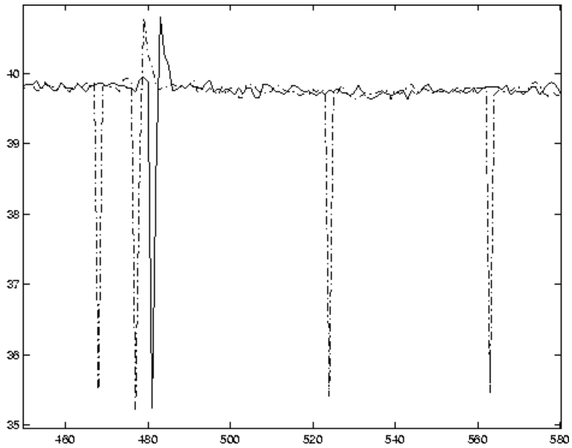


Figure 6-18 Real (continuous line) and simulated (dotted line) trajectory between data points 450 to 580.

It can be seen in these figures that the simulated data follows quite well the behaviour of the system. Just as in the case of the subsystem decomposition approach, the simulation essentially predicts the mean value of the output during the controlled regions. It also shows oscillations, this time even of the correct amplitude, though not exactly at the correct time instants. The MSE value for these data is 0.1873, which is slightly worse than using the model found by the subsystem decomposition method.

Another simulation has been performed using the best complexity-4 model found by means of the energy-based approach. This model is given by:

$$y(t) = f\{\ x_{28}(t\text{-}5),\ x_5(t\text{-}3),\ y(t\text{-}3\delta t)\ \}$$

which means that the output variable can be modelled from the fuel consumption, x_{28}, the air temperature in the compressor, x_5, and a delayed version of the output itself. The 1080 data points of the measured validation data set as well as the simulated data are depicted together in Figure 6-19.

Also this model mimics well the behaviour of the system. In Figure 6-20, two different windows of this simulation are shown; on the left, the data points from 450 to 520 are shown so as to exhibit more clearly the details of the modelled peak. On the right, the data points from 240 to 320 are shown so as to enlarge the erroneous peak that the complexity-4 model simulates.

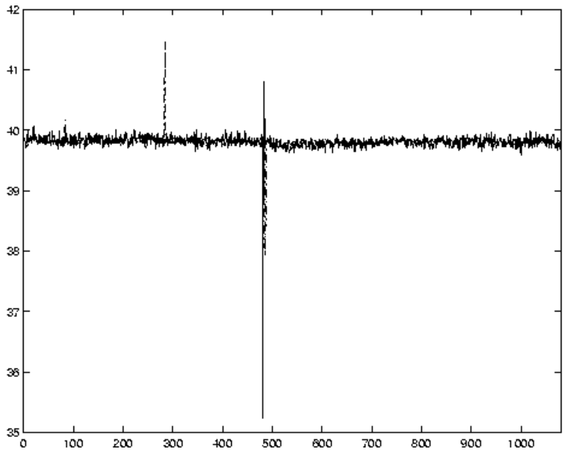


Figure 6-19 Real (continuous line) and simulated (dotted line) data when using the best complexity-4 model.

It can be seen that the simulated data follows the real data quite well. Yet, the used FIR qualitative model does not model the oscillation present in the validation data set as well as before. In despite of this detail, the MSE value of this simulation is 0.1734, slightly smaller than that of the complexity-5 model.

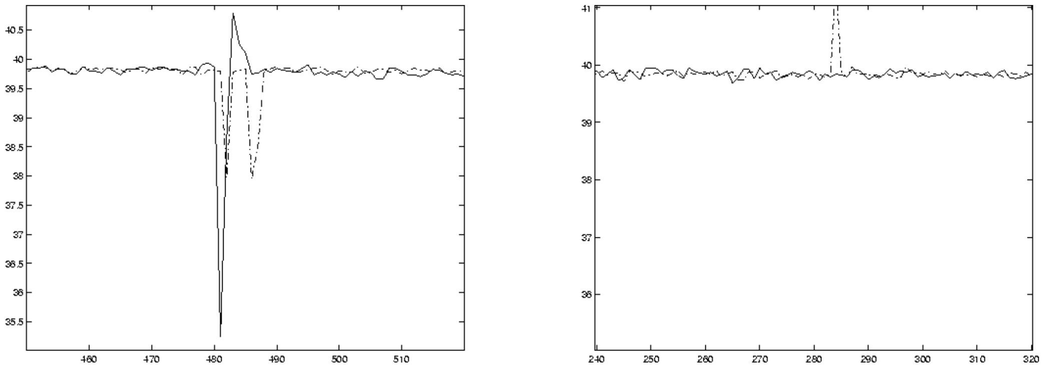


Figure 6-20 On the left, data points from 450 to 520 for the real and simulated data sets when using a complexity-4 model showing details of the real output peak simulation. On the right, data points from 240 to 320 showing an erroneous peak resulting present in the simulation only.

6.4 **Conclusions**

In previous chapters, some methods to reduce the FIR mask search space were presented so as to be able to compute a qualitative model of a large-scale system. Two subsets of these methods were combined in Chapter 5 to offer two different and independent approaches to dealing with large-scale systems in effective and efficient ways. Chapter 6 showed the application of these two methods to an industrial large-scale system, an industrial gas turbine used for electrical power generation.

The discussion offered in this chapter demonstrates the usefulness of having explained the stages of these algorithms individually, rather than having treated them as monolithic pieces of code. The subsystem decomposition method was slightly modified in order to ensure that the variable $y(t-\delta t)$ would not be eliminated from the data set. The energy-based method was modified in an even greater sense. Rather than computing one suboptimal model making use of the important power peaks up to the chosen mask depth, a series of suboptimal masks of increasing depths was computed. This was necessary in order to keep the number of '-1' elements inhabiting the mask candidate matrices sufficiently small.

Both the subsystem decomposition method and the energy-based approach found excellent suboptimal masks that can be used to qualitatively simulate the gas turbine system. Both techniques are acceptably fast considering the complexity of the modelling task at hand.

7 Final considerations

I got involved with Fuzzy Inductive Reasoning almost by chance. After obtaining my engineering degree, I went immediately to work in the private industry, yet the contact with the university was never lost. I was primarily interested in robotics, far from the methodologies developed in this dissertation, an area that I had worked on in my final project. Hence small studies in the mark of the university, parts of bigger projects, were still performed by me in those days.

It was a day, no matter which, in the spring of the year 1997, while working as a system and network administrator for a private company near Barcelona, that I went to the recently created Institute of Robotics. The purpose of that visit was to talk to a professor, for whom I was developing a system, part of a European project, to see if there was any possibility of getting a financial contract. While waiting in the hall of the Institute, a man, actually the director of that Institute, Rafael Huber, came out of his office and asked me whom I was waiting for. I was told that, unfortunately (or fortunately, we shall never know), this professor was not present as he had to attend an urgent interdepartmental meeting, so he could not meet with me that evening, and that, in turn, I could come into his office and talk to him.

During that talk, apart from other things, he told me about the Fuzzy Inductive Reasoning methodology. At first sight, it seemed like a fantastic methodology that could be used to predict a system without having any knowledge at all about it. Immediately it went through my head that this methodology could be implemented in the onboard computer of a mobile robot, and hence, be used to create models of the environment in which it lived. In this way, the robot would have an independent capacity to dynamically model its environment and to survive in it, just analysing information from its sensors, without any previous knowledge of the surrounding environment. So I began to study this methodology.

By the month of September within the same year, the possibility of obtaining a grant from the Spanish Government for performing FIR-related research arose. My application for this grant was approved, and so I began my research work on FIR.

In the first stages of this research, while learning about the FIR methodology, different bugs in the already implemented version of FIR (within the SAPS platform) were detected and fixed. This allowed me as well as other users of the tool to obtain improved accuracy in the predictions made with this methodology. Then, while researching different methods to evaluate the predictions that FIR makes, a new idea was conceived in the context of performing fault detection using FIR: the so-called envelopes. The envelope approach was compared with previous techniques published by the same group in *Simulation* in the years 1989 and 1994 simulating faults in a B-747 commercial aircraft, the same example that had been used in the previous publications. The formerly used crisp detection was replaced by the new approach, based on the computation of an acceptability interval for each predicted variable. The new approach resulted in improved capabilities of detecting faults earlier and in a more reliable manner.

It was not until some months of the year 1998 had passed that the work to be performed as part of my dissertation crystallised. The weather had become good enough in Barcelona so as to enjoy the sea. Dr. Cellier had just landed in Barcelona for one month, as he did frequently in order to provide guidance to the research direction of the group, and so, we had the opportunity to get acquainted and to talk about the FIR methodology.

I had realised that there existed a fundamental problem in the FIR methodology that had not yet been tackled: FIR was incapable to deal with large-scale systems effectively and efficiently due to the nature of its modelling engine. The model search engine of FIR is of exponential computational complexity, which makes it unsuited for dealing with large-scale systems (as in fact, most real industrial processes are). Hence it was agreed that my research should focus on studying methods so as to simplify the mask search space of FIR.

With this goal in mind, a first research stay of mine was organised to take place at the University of Texas during the fall semester of 1998, where I studied statistical methods under the supervision of Dr. S. J. Qin, who had previously published a number of interesting articles that let us to believe that the FIR methodology could benefit from that knowledge. Statistical methods were studied in order to perform a first variable pre-selection among all of the variables of a large-scale system in order to provide FIR with a reduced set of variables. Yet in this first stay, only linear and static relations between variables were considered. As non-linear relations are very important in most engineering systems, and since the attained results suggested that linear techniques on their own were inappropriate to deal with the complex tasks at hand, a second research stay was organised to take place in the fall of 1999, when I had the chance to study non-linear statistical techniques in a research group of the Università degli studi di Napoli under the direction of Dr. Carlo Lauro, head of the Mathematical and Statistical department of that university. My research focused on searching for methods to simplify large-scale systems, but this time including potential non-linear relations among variables, as well as dynamic information, i.e., taking into account the time variable.

The two research stages abroad proved very valuable indeed and provided the framework for the research performed in this dissertation, as this dissertation deals with a variety of methodologies, predominantly statistical in nature, to reduce the mask search space of the FIR qualitative modelling methodology by pre-selecting a subset of variables and by reducing the number of potential inputs in the mask candidate matrix to be considered, thereby allowing the FIR methodology to deal with large-scale systems (real industrial systems) within reasonable time. Among all of the methods considered in this research work, two complete methodologies have been derived. They both can be viewed either as exploratory steps taken for the purpose of gaining information about the physical system under study, or as complete modelling methodologies when used in combination with the FIR modelling engine.

Those developed methodologies enhance the FIR methodology with a large-scale system modelling capability, by means of reducing the computing time needed to obtain a sub-optimal qualitative model. Previously, given a large-scale system, FIR was theoretically capable of finding the optimal model, but in practice, the computational cost was so high that this could not be accomplished within reasonable time. Now, it has become possible to obtain a qualitative FIR model of a large-scale system within a few hours, where previously the same task would have required years if not centuries of computing time. This was the case of the gas turbine system for electric power generation analysed in Chapter 6; the given methods allowed to compute a qualitative model for this system in a time of approximately 35.5 hours on a SUN/Solaris workstation.

Both of the developed techniques are designed for use as precursors to a FIR optimal mask search. They both reduce the computational burden of the subsequent FIR modelling task by reducing the number of potential inputs ('-1' entries in the mask candidate matrix). Both techniques employ heuristic parameters by which the number of remaining '–1' entries can be indirectly controlled.

How many '–1' entries should remain in the mask candidate matrix? If the techniques are tuned such that the number of '-1' entries is reduced to 4 or less, the techniques become alternatives to FIR rather than precursors to FIR, since no further model simplification is then needed. The resulting potential inputs can in this case simply be taken as the inputs of a suboptimal qualitative model. Although it would be possible to tune the algorithms in such a way, this is not desirable, since FIR is clearly the best technique available for finding good masks. Thus, if the FIR step were bypassed, the algorithms would often result in suboptimal models of significantly inferior quality compared to the optimal model. The trick is to keep the number of remaining '-1' entries large enough to include, among the masks compatible with that mask candidate matrix, one or several masks of very high quality, not significantly inferior to the truly optimal mask (which often may indeed be among those spanned by the mask candidate matrix), yet keep the number of remaining '-1' entries small enough so that the subsequent FIR model search can be performed exhaustively within reasonable time limits.

What is the largest number of '-1' elements that FIR can handle within reasonable time? A mask candidate matrix with somewhere between 15 and 25 '-1' entries can be searched very quickly. Depending on the size of the database, the exhaustive search algorithm may take from a few seconds to a few minutes to find the optimal mask. Given a mask candidate matrix with somewhere between 25 and 40 '-1' entries, it takes the search algorithms from a few minutes to a few hours to complete the search. If a mask candidate matrix with somewhere between 40 and 60 '-1' entries is considered, the modelling engine will take from a few hours to a few days to complete its search. Anything beyond that is unreasonable. Due to the exponential computational complexity of the exhaustive search algorithm, the amount of time needed grows rapidly with increasing numbers of '-1' elements. Soon we would need weeks or months or even years to complete a search, and with a mask candidate matrix containing only a few hundred '-1' entries, the search algorithm would take the entire life span of this universe to complete an exhaustive search.

What is the smallest number of '-1' elements that should be retained? The answer to this question may depend on the application. Evidently, most real-time applications requiring on-line modelling are out of the question. Luckily, many applications allow a model to be made off-line, such that only the FIR simulation is to be performed on-line. Then, a few hours or even days for obtaining a good model may be acceptable, if the resulting model is of high quality and capable of performing its designated task. Thus, the smallest number of '-1' elements is one that avoids unnecessary computation while preserving a high probability that a model of excellent quality can be found. It has been our experience that the optimal number of '-1' entries to be retained is somewhere between 15 and 50.

How can a promising set of variables be pre-selected? First investigations, and many of those are presented in Chapters 3 and 5 of this dissertation, tried to answer this question by throwing a palette of statistical methods at the problem. Unfortunately, the results obtained in this way were not very good. Most large-scale engineering problems are highly non-linear, whereas all of the standard statistical techniques are linear. Thus, important non-linear relations are being overlooked in this way, and variables are not included in the list of important variables because of this.

Hence another paradigm was needed: rather than searching for variables to be *included* in the set of good variables, it may be better to start by looking for variables that can be *discarded* from the set of good variables. The reasoning leading to this conclusion is straightforward: if two potential input variables are highly linearly correlated with each other, then they contain essentially redundant information. It is not necessary to keep both of them in the set. Any one of them suffices, while the other can be discarded without restrictions to generality. Non-linear relations do not matter in this context. If there exists a strong non-linear relation between two potential inputs, the algorithm would simply keep both variables, though one might suffice also. No harm is done in this way. No potentially valuable information is being thrown away.

Consequently, an algorithm was designed that groups all input variables into subsets of variables that are strongly linearly correlated with each other. Variables may belong to more than one such subset. Once the groups are formed, one variable is retained from each subset, as are all variables that are not included in any of the subsets. Usually, this technique gets rid of at least 80% of all variables forming a large-scale system.

Good techniques exhibit weak correlations among its inputs, minimising the redundancy of information contained in them, yet they exhibit strong correlation to the output. The previously outlined algorithm takes care of the former requirement, but ignores the latter, since the output has not yet been looked at. The second step of the method thus identifies groups of variables that are strongly correlated with the output. If there exists a strong non-linear relation between a potential input and the output, it will not be discovered; yet, no harm is done, since no variables are being discarded in this step. Variables that do not exhibit strong linear correlation with the output are simply set aside for later consideration.

The third step analyses non-linear relations between variables not chosen so far and the output. Since non-linear relations cannot be analysed in general using a closed-form algorithm, this step is iterative. It performs a parameterised non-linear transformation on the data available for the variable in question using a spline transformation. The parameters of that transformation are optimised so as to maximise the linear correlation of the transformed variable with the output. If a strong correlation can be found in this way, then the variable in question is non-linearly related with the output and joins the set of variables to be retained[1].

The resulting variables except for the output itself are proposed as potential inputs to the FIR model search engine. The algorithm, as it is currently implemented, retains usually in the order of 20 to 30 variables, which is just about right. We know that the output one time instant delayed, $y(t-\delta t)$, contains the most valuable information in any system for predicting $y(t)$. Thus, if that variable got thrown out in step 1, as this sometimes happens, it is immediately added again to the set of retained variables.

In spite of the fact that the algorithm summarized above gives excellent results most of the time, it *is* a suboptimal search algorithm. There is no guarantee that the optimal mask is retained within the set of masks to be searched, and therefore, the user is left unclear as to how optimal the "optimal mask" returned by the algorithm really is. The algorithm is certainly reasonable, and frankly, I have not found a single case where the optimal mask returned was far off from the truly optimal one (for those cases, where this could be checked, which is not true when dealing with large-scale systems).

Yet, due to the suboptimal nature of the proposed algorithm, it may be useful to have access to an alternative suboptimal search algorithm that operates on different principles, so that the user can run both suboptimal algorithms in parallel. If both algorithms return masks of similar quality, this will increase the confidence of the user that he or she has indeed found a high-quality mask that is not much inferior to the truly optimal one.

For this reason, a second suboptimal search algorithm has been developed as part of this dissertation. It is based on energy considerations. Each input/output pair can be considered as the input and output signals of a SISO system. It may make sense to look at the power spectrum relating the input to the output. The greater the "energy contents" that is being transferred from the input to the output, the larger the likelihood that there indeed can be postulated a system connecting the input to the output that is able to regenerate large portions of the output signal from the input signal. This is similar to a cross-correlation analysis, yet it is intrinsically non-linear, and makes use of engineering considerations to select an appropriate non-linear relation. Energy and power are at the heart of physics. These concepts are useful for arbitrarily non-linear physical systems, and the relations among power variables are essentially linear in every system. Let us consider, for example, a non-linear resistor, such as a hydraulic valve. The relations between the signal variables (pressure and volume flow) are non-linear, yet the power transfer across the valve is linear. It simply states that the power dissipated in the valve equals the heat generated by it.

Details of the algorithm were provided in Chapter 5. Of course, it may make sense to precede the application of this algorithm by the first step of the previous algorithm, i.e., replace only steps 2 and 3 by it; yet, by doing so, the two algorithms share an important component, and therefore, the argument of independence of the two algorithms is somewhat lost. For this reason, the energy-based method, as discussed in Chapter 5 and applied to a large-scale system in Chapter 6 did not make use of the variable reduction step of the subsystem decomposition algorithm.

[1] The algorithm as it is implemented works somewhat differently. Rather than looking for relations with the output only, groups of variables are formed, and those groups are retained that contain the output.

7.1 Contributions

There are two main lines of contributions in this dissertation. The first one concerns the contributions made to the FIR methodology, around which the dissertation has been developed. The second line relates to the methodologies presented so as to reduce the mask search space of FIR, so reducing the computational time needed to obtain a FIR qualitative model of a large-scale system. Those, although developed in a FIR context, can be used as precursors to any dynamic modelling methodology.

In the context of the FIR methodology, three main contributions were made:

- The **corrected five-neighbours prediction formula.**

  When working with an academic model of a water tank [Mirats and Escobet, 1997b] with one input (an electric signal control to drive a pump) and one output variable (the signal read out from a flow sensor), an error was found in one of the formulas of the FIR prediction module. A non-existing class was found to be predicted by FIR, causing the next input state vector in the behaviour matrix not to be found and making the prediction unstable, even when predicting already observed data. The error relates to the formula computing the new output state as a weighted sum of the output states of the five nearest neighbours. Due to the rounding used in that formula, it sometimes happened, when all five nearest neighbours were in the highest class, that a class was predicted for the output state that was one higher than the largest class (and similarly for the lowest class). Since these classes are not contained in the data base, already the next step, which invariably makes use of *y(t-δt)* among its inputs, would be confronted with an input state that is not contained in the data base at all, either leading to an immediate termination of the prediction or making the prediction of the subsequent output state arbitrary, depending on how the SAPS program had been written. This error has been corrected. The problem was briefly mentioned in Section 2.5 of the dissertation. This and similar improvements of the existing SAPS code were not elaborated on in the bulk of the dissertation, since they do not represent new concepts, and their detailed explanation would not have aided the understanding of the important new concepts developed in this dissertation; yet, their implementation consumed a considerable amount of time, especially during the early phases of the dissertation research.

- The **new hill-climbing algorithm for a suboptimal mask search.**

  A new hill-climbing approach for suboptimal mask search was designed and implemented as part of this dissertation. This aspect of my work was reported in Section 2.4.1 of the dissertation. It uses a bootstrapping approach to obtain successively masks of improved quality. It begins with a mask candidate matrix of depth one with '-1' entries in all input variable locations and determines the optimal mask, i.e., the optimal static model. It then keeps the selected inputs as potential inputs of a new mask candidate matrix of depth two, adding a second row to the mask candidate matrix from the top that is again filled with '-1' elements. A second optimal mask is computed in this way. The algorithm proceeds by successively incrementing the depth, until the quality of the resulting optimal mask can no longer be improved.

Why is this technique attractive? Instead of solving one optimal mask search problem with $d \cdot n - 1$ '-1' entries, where *d* is the depth of the mask and *n* is the number of variables, it solves *d* optimal mask search problems with usually somewhere between *n* and *2n* '-1' entries. The latter is far more economical than the former. In addition, the algorithm offers a natural criterion for the selection of *d*, which in most other algorithms must be manually chosen.

Contrary to the previously known hill-climbing algorithms, which often generate suboptimal masks of vastly inferior quality, this algorithm works amazingly well in most cases, generating a suboptimal mask of very high quality. Yet, the algorithm was not proposed as an alternative to the subsystem decomposition method and the energy method discussed in Chapter 5 and applied to a large-scale system in Chapter 6, because it is less economical than either of those approaches. The technique is still too slow to be used for modelling large-scale systems.

The method is nevertheless important, because its first two steps are used as part of the energy-based approach, since the energy consideration has a limited resolution and does not provide information for delays ≤ 2.

The algorithm was first proposed in [Mirats and Verde, 2000] and applied to a garbage incinerator system, which is described in the Appendix I.3.

- **The concept of variable acceptability, the so-called 'envelopes.'**

The concept of a variable acceptability interval, called "envelope," is a new idea introduced in the FIR methodology. It was derived when working with a B-747 aircraft in order to detect structural changes in the simulated aircraft model, so it was used for fault detection purposes (cf. Appendix II.2 for an application). The method was reported in Section 2.5.1 of the dissertation. However, since this contribution concerns the *simulation engine* of FIR, whereas the bulk of my research relates to its *modelling engine*, it was not further elaborated on in the subsequent chapters of the dissertation.

The algorithm works as follows. Apart from the predicted signal, which is a weighted sum of the five nearest neighbours, two more predictions are computed, representing the smallest and largest predictions possible with any one of the five nearest neighbours. In this way, an interval of acceptability of the real trajectory value is obtained for each forecast. By this method, three separate trajectories are computed, the predicted one, and an upper and lower limit to it, forming an acceptability interval into which the predicted signal (as well as the real monitored signal when the obtained model is good) should fit. The formed interval has two main applications: fault detection and model validation. The method was reported in [Mirats and Huber, 1999; Escobet *et al.*, 1999].

- **The use of a modified H_r value as a *variable similarity measure.***

  The quality of the mask is based on two quantities, H_r, representing an uncertainty reduction ratio, and O_r representing an observation ratio. H_r is based on the Shannon entropy formula applied to the so-called state transition matrix (STM). Each element of this matrix provides a measure of confidence in an output state given an input state vector. If this matrix is generated for a two-variable system, a measure of linear as well as non-linear correlation between the two variables can be derived. In this case, the state transition matrix is of size $N \times M$, where N and M are the numbers of classes into which the two variables have been recoded. When the two variables are strongly positively correlated with each other, the STM is close to a diagonal matrix with the diagonal elements assuming values close to 1. When the two variables are strongly negatively correlated, the STM is close to an anti-diagonal matrix. Finally, when the two variables are uncorrelated, the STM approaches a matrix where all elements assume the same value, this value being 1/number of classes of the output variable. Hence the STM can be used as an indicator of linear as well as non-linear correlation between two variables. The method was reported in Section 3.4.3 of the dissertation. It can be used as an alternative to the linear correlation technique in the first step of the subsystem decomposition method.

- **Improvement made to the FIR simulation engine.**

  When predicting the future behaviour of an electrical DC motor [Mirats and Escobet, 1997a], it was discovered that the peaks of the predicted signal were frequently underestimated, whereas the valleys were overestimated. Thus, the predictions did not span the entire range of the observations made. These errors were caused by a bug in the *fforecast* routine of SAPS. The same error was also present in the newer version of that routine, called *mexfforecast*, which uses dynamical memory programming techniques. Since the predictions had not been incorrect, only imprecise, the bug had not previously been discovered. I fixed this error, and all of the FIR simulations presented in the dissertations make use of the corrected version of the FIR simulation engine. The problem was briefly reported in Section 2.5 of the dissertation.

The contributions in the line of reducing the mask search space of FIR, also viewed as previous modules to modelling methodologies, are:

- **New use of the unreconstructed variance methodology**.

  This contribution represents not a new design, but only a new use of an already existent methodology that had originally been developed at the University of Texas in Austin [Dunia and Qin, 1998; Dunia, 1997; Qin and Dunia, 1998]. The unreconstructed variance methodology was previously used to select the number of principal components to be retained in a PCA model, based on the best reconstruction of the variables. The purpose of the PCA model was to identify faulty sensors in a system, and to reconstruct sensor data values from measurements of sensors attached to other signals, exploiting the redundancy inherent in multiple sensor data streams. The methodology had not been designed as a tool for finding an input/output model of a system, though the two tasks are evidently related to each other.

When a PCA model is used to reconstruct missing or faulty values, the reconstruction error is a function of the number of intervening principal components. In order to determine the number of principal components (PCs) to be used, the methodology proposes making use of the variance that the model cannot reconstruct; that is, it uses the variance of the reconstruction error.

Prior to determining the number of PCs to be retained in the model, the available measurement data are analysed to determine what variables are well reconstructed from which others. Given a system with *k* variables, every variable is reconstructed from the other *k-1* variables, and its unreconstructed variance is computed as a function of the number of retained principal components. The method then proceeds by summing up the unreconstructed variances of each column. The number of PCs for which the sum of the unreconstructed variances is a minimum is determined. The method then throws out those variables that exhibit, for the optimal number of retained PCs, unreconstructed variances that are bigger than the value that would be obtained if the measurement data for those variables were replaced by their mean values.

Now, if variable selection is intended with this methodology, two subsets of variables are obtained. The first one corresponds to a subset of variables with a high linear correlation between them. The second subset is formed with variables with low linear correlation with the first one. The concept has been reported in Section 3.3.2 of the dissertation. The performed variable selection can be used, as shown in Chapters 3 and 5, as a precursor to the FIR modelling methodology.

- **Delay estimation based on energy considerations**.

In the FIR context, each variable trajectory has always been analysed from a deterministic point of view, i.e., at each time interval, every measured value has been considered to match the real value of the physical observed variable. In reality, each observed variable trajectory represents a superposition of two values, one measuring a desired physical characteristic, such as the fuel flow through a pipe, the other denoting an added noise component, representing e.g. measurement noise or thermal noise. The energy-based approximation considers each of these trajectories as a realisation of a stochastic process, i.e., it acknowledges the existence of a deterministic as well as a random component in the available trajectory. With this interpretation, the energy of the signals can be computed and used to determine, at which delays each input variable contains more energy relating to the output. In this way, the most probable delays, for which the input variables are to appear in a FIR qualitative model, can be obtained. Details of the algorithm were presented in Section 5.3.2 of the dissertation, and the algorithm was applied to modelling a large-scale system in Section 6.3.3. This contribution represents one of the two most important cornerstones of this dissertation.

- **Subsystem decomposition.**

The bulk of this dissertation focuses on the exponential computational complexity of the FIR modelling engine, which hampers its application to large-scale systems. If a decomposition of the complex system into subsystems can be achieved, the computational cost may be dramatically reduced: given a *k*-variables system, the cost of computing one global model involving all *k>>1* variables is much higher than that of computing *p* submodels, each of which involves a much smaller number of variables. Several subsystem decomposition methods were studied in this dissertation.

▪ **Subsystem decomposition using FRA.**

A new complete subsystem decomposition algorithm employing Fuzzy Reconstruction Analysis (FRA) was proposed in Section 4.2.7. It makes use of one of the suboptimal structure search algorithm of FRA, called the single structure refinement algorithm. It makes use of the relative strengths of the binary relations between each pair of variables to derive a meaningful subsystem decomposition of the overall system. Unfortunately, the technique is not applicable to large-scale systems, as the computational complexity properties of FRA are even worse than those of FIR.

▪ **Subsystem decomposition using FIR.**

A new complete subsystem decomposition algorithm employing Fuzzy Inductive Reasoning (FIR) was proposed in Section 4.3. The algorithm starts with the output, finds an optimal mask of the output. It then treats each of the inputs of that model as new outputs, determining additional optimal masks to compute those. The algorithm proceeds until all variables of the overall system appear in the decomposition. The method offers an alternative to the FRA-based method, but it also cannot be applied to large-scale systems. Its very first step already involves finding a FIR model of the overall system, which, as we meanwhile know, cannot be done in the case of a large-scale system, using FIR alone.

▪ **Subsystem decomposition using statistical techniques.**

A new complete subsystem decomposition algorithm based on linear and non-linear statistics was proposed in Sections 4.4 and 4.5 of the dissertation. The algorithm uses statistical considerations to determine a set of input variables that are strongly correlated with the output, yet weakly correlated among each other. The algorithm consists of two steps. In a first step, only linear correlations are being studied. The second step takes also non-linear correlations into account. Just like the FIR-based approach, the method starts with the output and identifies a set of input variables, i.e., a model of that output. It then treats each of the inputs as a new output, and finds a model for each of them in the same way. The algorithm proceeds until all variables of the overall system appear in the decomposition. Albeit much more complex than its two competitors, this is the algorithm of choice, as it is the only one of the three algorithms that can be applied to large-scale systems.

- **Decomposition-based method for variable pre-selection.**

  An algorithm was developed to determine a sparsely populated mask candidate matrix by reduced set of '-1' elements, to be used as a precursor to FIR modelling of a large-scale system. The algorithm is based on the statistical subsystem decomposition method. In a first step, the method reduces the number of variables by considering the redundancy of information contained in the available potential inputs. The second and third steps of the algorithm represent variants of the statistical subsystem decomposition method mentioned above. The fourth step is a FIR modelling step using a sparsely populated mask candidate matrix. This algorithm is the second cornerstone of the dissertation. Its details are explained in Section 5.3.1 of the dissertation. Its application to modelling a large-scale system is presented in Section 6.3.2.

- **Re-implementation of the *Fuzzy Reconstruction Analysis* module.**

  As FRA in its former implementation was limited to studying systems with no more than 20 variables, I decided to re-implement the entire FRA module from scratch using Matlab. The FRA methodology and software module are presented in Section 4.2 of the dissertation. All FRA results presented in this dissertation were obtained using the new implementation of the FRA methodology.

7.2 Future work

There are two main aspects of the methodologies discussed in this dissertation that should be further investigated. The first of them, and perhaps the most important issue, is the manner, in which time is included in the analysis when lots of variables are taken into account. As stated in the work presented here, time can be introduced in the analysis by means of duplicating the raw data model, shifting it by one row, and adding the shifted data to the original data as additional variables. Yet, this form of including the inherent time dependency of the data makes the obtained raw data model huge, which in turn makes the analysis exceedingly slow. Other forms of including time information in the data should be researched.

Also, both the decomposition-based method and the energy-based approach proposed as precursors to FIR modelling of a large-scale system contain parameters that indirectly influence the sparsity of the resulting FIR mask candidate matrix. No investigation was done trying to optimise the values of these parameters with the goal to maximise the computational efficiency of the overall algorithms. This should be further investigated.

The second main aspect refers to the decomposition of a large-scale system into subsystems. Three algorithms for obtaining complete subsystem decompositions were outlined in Chapter 4 of the dissertation. Yet, it was not further analysed whether such complete subsystem decompositions might be preferable to the approaches taken in Chapters 5 and 6 when simulating a large-scale system using FIR. Questions revolve around the resolution of simulation results obtainable by such complete subsystem decompositions and around the optimal number of sensors that a large-scale system should be equipped with.

7.3 Concluding remarks

Ph.D. research is necessarily open-ended. If the results to be obtained could be assessed in advance, the work would not have the depth and breadth expected of a Ph.D. research effort. However, a beginning Ph.D. student is not yet an expert in the field of his or her research area. It is difficult for such a student to assess, ahead of time, how promising a given research direction might be. This requires a lot of experience, experience that the student is lacking at that time in his or her career.

Due to the open-ended nature of Ph.D. research, the precise duration of a Ph.D. research effort cannot be known in advance. Yet, research grants are usually of a pre-determined duration, and students deserve to have confidence that their respective research efforts are likely to be completable within the foreseen time frame. This can only happen if the student is guided well by his or her dissertation advisor(s). I was lucky in this respect. My two dissertation advisors, Dr. Huber and Dr. Cellier, devoted an unusual amount of time and dedication to me and to my research, for which I am eternally grateful. Unfortunately, this is far from the norm. I have known a fair number of Ph.D. students who started their respective projects with a lot of enthusiasm, but got lost in the details of their work due to a lack of devotion and, it seems, interest on the part of their advisors, and finally gave up, when their respective research grants expired, without ever completing their dissertations. This is unjust, and it frightens the other students, when they see occurrences of such outcomes over and over again. It also frightened me. Would I be able to complete my thesis, or would my efforts end in the same way as those of numerous others? This surely gave me a good number of nightmares as well as sleepless nights throughout my dissertation research.

Would I do it again, knowing what I know now? I guess so, but I am glad I didn’t know the size of the wall ahead of me when I started climbing it.

Cuando alguien busca, puede ocurrir fácilmente
que su espíritu sólo vea el objeto que busca
-que no sea capaz de encontrar nada ni de
admitir la entrada de ninguna cosa en sí mismo,
porque sólo piensa en lo buscado, porque tiene
un objetivo y está poseído de él.
Buscar es tener un objetivo, pero hallar es ser libre, estar abierto, no tener una meta.

Hermann Hesse

Appendices

I Case Studies

In this appendix, the different systems used throughout this dissertation as examples are presented.

I.1 Water tank

This system is presented as an academic example to illustrate the description of the FIR methodology given in Chapter 2. The water tank system exhibits a second order dynamics. In Figure I-1, a plant scheme of the system is given, [Mirats and Escobet, 1997].

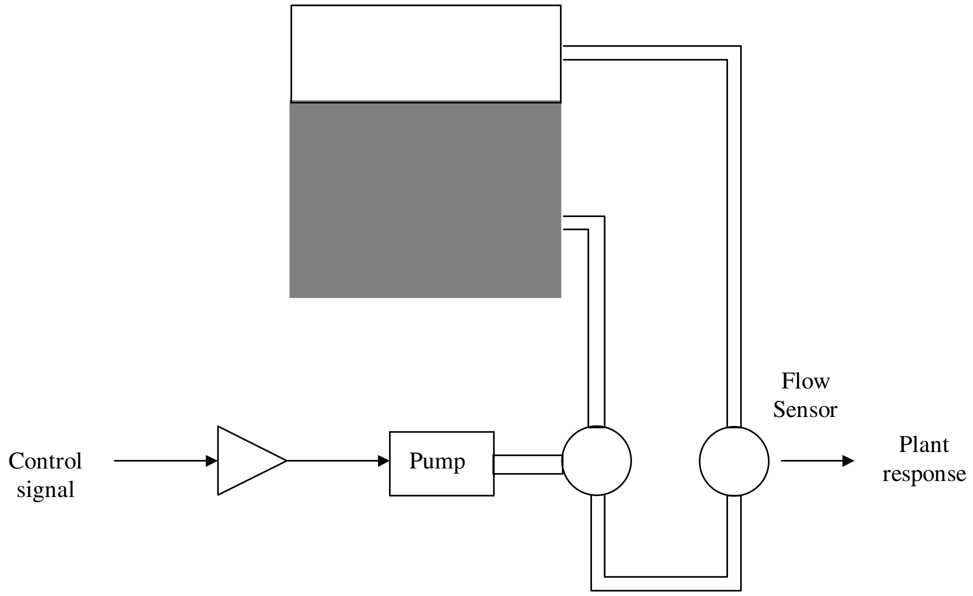


Figure I-1 Plant scheme for the water tank system

The electric pump is driven by a DC voltage, ranging between zero and ten Volts. It makes the water stored in the tank re-circulate. The flow of this water is measured by means of a flow sensor, giving the considered output for this academic example. This system is available in the labs of the department, thus it is easy to excite it in different ways so as to obtain data for the qualitative modelling effort.

In order to gather dynamic data from the water tank, a control signal is used that can take any value between 0 and 10 Volt, allowing random changes in the control signal value once every second. Data is gathered during 100 seconds, and the used sampling period is 0.02 seconds. Under those conditions, 5000 data points were obtained from the system. In Figure I-2, the input signal to the pump is shown as a continuous line, whereas the obtained output is depicted as a dotted line.

Prior to the qualitative modelling process a quantitative model is to be identified. Considering the given system as a second order system, it can be modelled by the following transfer function:

$$H(s)=\frac{30.9136}{s^2+8.5257s+30.9136}$$

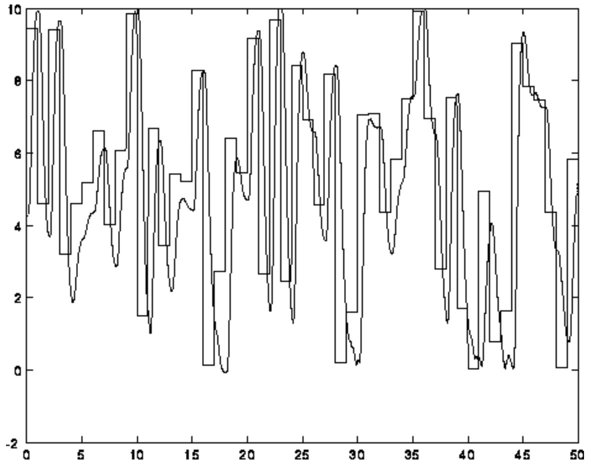


Figure I-2 Real input and output signals of the system

The system is simulated in Matlab by means of its transfer function representation. If a Bode analysis is performed, the cut-off frequency when the gain is *-3 dB* is w_{3db} *= 5.56*. In this case, the w_{3db} frequency coincides with the bandwidth of the system, so the minimum required sampling frequency for this system would be *2 Hz*. When gathering data, the system was sampled at a *50 Hz* frequency, so the data have been gathered correctly.

The presented quantitative model can be used to predict the future of the system. The results of predicting the last 10 seconds of the output, when using the real input for this time span are shown in Figure I-3.

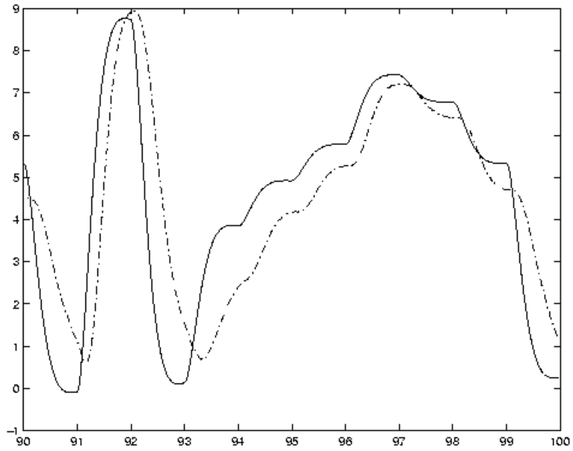


Figure I-3 Real and simulated (dotted) output signals when using the quantitative model

The relative error for this quantitative simulation is shown in Figure I-4.

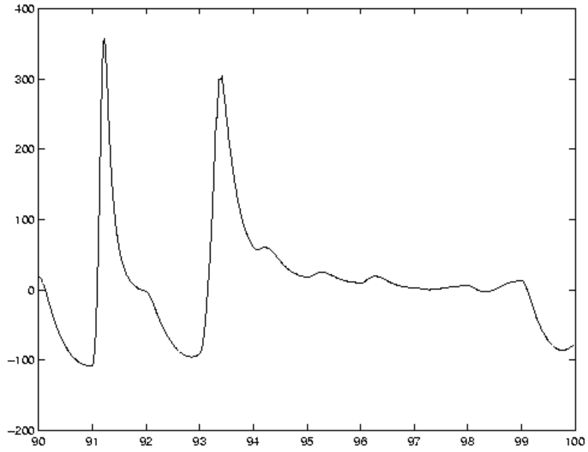


Figure I-4 Percent of relative error in the quantitative simulation

Now, a FIR qualitative model is to be obtained. The input and output variables are to be recoded into 3 classes each. The model is constructed using 90% of the available data, whereas the remaining 10% are used to validate it. The depth of the proposed candidate mask is 15, and the maximum allowed complexity is 5. All elements of the mask candidate matrix are set to '-1' except the element (15,2), which is set to '+1' as it represents the *m*-output. The mask candidate matrix is provided together with the measured data to the qualitative modelling engine. The optimal masks found for complexities 2, 3, and 4 are:

$$m1a = \begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -2 \\ 0 & 1 \end{pmatrix} \quad m1b = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} \quad m1c = \begin{pmatrix} 0 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -2 \\ -3 & 1 \end{pmatrix}$$

The qualities of these masks are:

$$q_a = 0.9530 \qquad q_b = 0.9357 \qquad q_c = 0.7124$$

Figure I-5 shows the real and simulated output obtained with the optimal FIR model, which is of complexity 3, and Figure I-6 exhibits the percentage of relative error of the simulation results. The obtained results are considerably better than those obtained with the quantitative simulation.

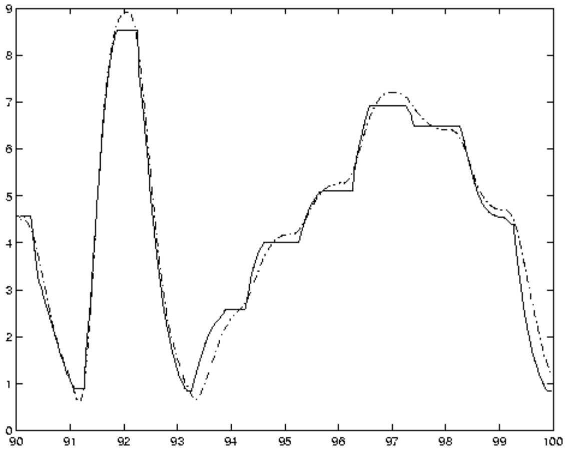


Figure I-5 Real and simulated output when using the qualitative model

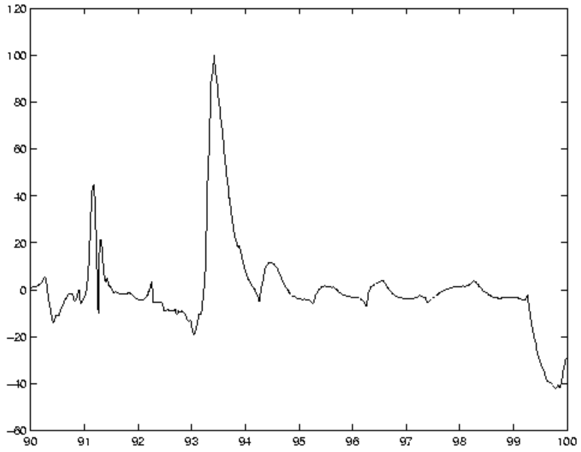


Figure I-6 Relative error (percent) of the qualitative simulation

I.2 Steam generator (boiler)

Data stemming from a steam generator process have been used throughout the dissertation as an example, in order to compare the results obtained using different variable selection techniques. Concretely, the system analysed in this section has been utilised in Sections 4.2, 4.3, 4.4, and 4.5.

Figure I-7 shows a schematic of the boiler process. The variable to be predicted is the *NOx* content sampled from the boiler stack. Eight input variables are considered to have influence on the *NOx* emission level. Table I-I shows the considered variables.

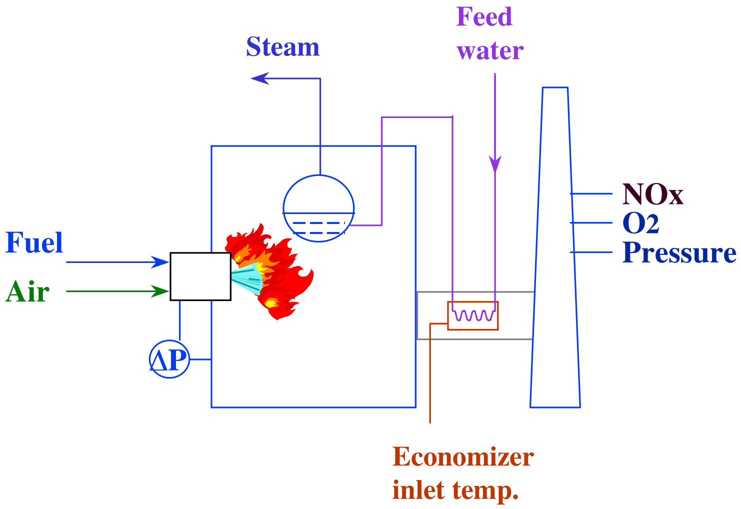


Figure I-7 Schematic of the boiler process

Variable	Physical meaning
1 (input)	AirFlow (KPPH)
2 (input)	Fuel Flow (Pct)
3 (input)	Stack Oxygen (%)
4 (input)	Steam Flow (KPPH)
5 (input)	Economiser Inlet Temp (F)
6 (input)	Stack Pressure (in H_2O)
7 (input)	Windbox Pressure (in H_2O)
8 (input)	Feedwater Flow (KPPH)
9 (output)	NOx, PPM

Table I-I Variables of the boiler system

Using a sampling interval of five minutes, 632 data points were collected during a period of significant change in the boiler throughput so as to cover a wide range of process behaviour.

A full FIR model was constructed using 85% of the gathered data. The remaining 15% of the measurement data were used to validate the model. Since the system under investigation contains only 9 variables, the FIR approach can be used directly to model the system under study, and to select a group of input variables to predict the output of the boiler.

It was decided to recode each of the 9 variables separately into 3 classes using equal frequency partitioning to determine the landmarks. A mask candidate matrix of depth 5 was proposed, postulating that significant *m*-inputs may lag no more than 20 minutes behind the *m*-output. All elements of the mask candidate matrix were preset to ‘–1,’ except for the element (5,9), which was set to ‘+1,’ as it represents the location of the *m*-output.

The optimisation problem was solved using exhaustive search, except that the maximum allowed complexity of the mask (the maximum number of mask elements different from 0) was limited to five. The set of best masks of each complexity was retained as promising masks to be investigated further.

The retained masks are shown below. The mask of complexity 4 exhibits the highest quality, followed by the mask of complexity 5, followed by that of complexity 3. The mask of complexity 3 treats the *NOx* level as a univariate time series, since it proposes that the future behaviour of the *NOx* level can best be predicted taking into consideration its own past behaviour only.

$$m4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$m5 = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$m3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

None of the masks made use of any variables except variables 2, 3, 8, and 9. Hence any successful variable pre-selection algorithm should retain these same variables, filtering out the ones that FIR would not consider in an optimal mask analysis.

FIR can itself be used as a variable pre-selection algorithm, as long as the overall number of variables is not too large. In the example presented here, it is not evident that a mask depth of 5 suffices to capture the best possible masks. Yet, a candidate mask of greater depth would already, in the case of a 9 variable system, lead to unacceptably large optimisation cost. Hence, the previously obtained results were used to postulate a new candidate mask, this time of depth 16, spanning a time period of 75 minutes, in which the variables 1, 4, 5, 6, and 7 were disabled by presetting all elements of the mask candidate matrix located in those columns to 0.

The same optimisation approach was used as before to determine the set of best masks. Evidently, the masks found earlier are still within the search space, i.e., if other masks are retained, they must be better than those found earlier. The three retained models are as follows:

complexity 5 : $y(t) = f\{x_3(t),x_3(t-15),x_8(t-9),y(t-1)\}$

complexity 4 : $y(t) = f\{x_2(t-1),x_3(t-15),y(t-1)\}$

complexity 3 : $y(t) = f\{y(t-1), y(t-5)\}$

None of the retained masks is identical to any of the ones found earlier, i.e., the larger mask depth indeed paid off. As before, FIR chose an autoregressive model in the case of the mask of complexity 3.

Figure I-8 shows the output data validation set and the prediction of the output variable using the three retained masks. To this end, another facet of the FIR methodology is being used. In a prediction, i.e., a qualitative simulation, FIR not only forecasts future values of the *m*-output; in addition, it generates estimates of the quality of these predictions in the form of a confidence value. How FIR estimates the confidence it has in its own predictions is explained in detail in [López, 1999]. The predicted value at each time instant is determined as follows. In each simulation step, the three masks are used to compute three separate predictions of the *m*-output value. Each prediction is accompanied by a confidence value. The prediction with the highest confidence value is retained as the true prediction for that step.

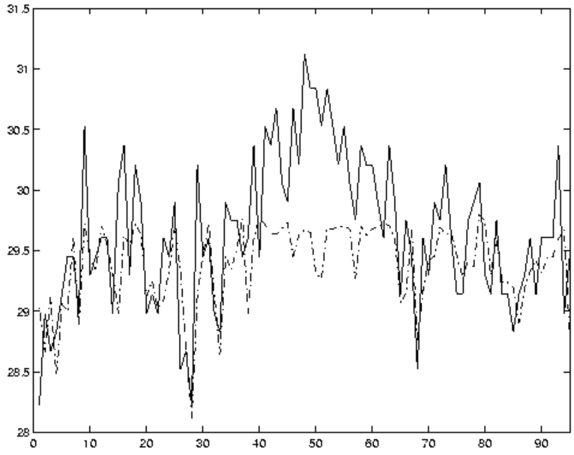


Figure I-8 Original and predicted validation set for the output using FIR

The reader may notice that no *NOx* value beyond 29.7 PPM was ever predicted, i.e., whereas the lower *NOx* levels are predicted (dashed line) fairly accurately, the higher values are not. The reason is that the training data do not contain any *NOx* levels beyond 29.7 PPM. FIR can only predict patterns that it has observed before. Since it has never seen such high *NOx* levels, it cannot predict their existence. The MSE error of this forecast is 0.5522.

I.2.1 FIR models excluding temporal relations

In order to be able to compare the quality and the computing reduction achieved when using different variable selection techniques (see Chapters 3 and 5), static FIR models of the boiler are needed, that is, models excluding temporal relationships. The candidate mask of depth 1 proposed for this purpose is:

$$mcan = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 \end{pmatrix}$$

Table I-II shows the retained masks for each of the allowed complexities, as well as the normalised MSE values of the predictions obtained using each one of them.

Complexity	Model	Quality	MSE
2	(0 0 0 0 0 0 0 -1 1)	0.1532	1.1510
3	(0 0 0 0 0 0 -1 -2 1)	0.1523	1.2087
4	(0 0 -1 -2 0 0 0 -3 1)	0.1418	1.0974
5	(0 -1 -2 0 -3 0 0 -4 1)	0.0900	0.8234

Table I-II FIR models without temporal relations

Since the search space is now much smaller, the optimal masks can be found much more rapidly. However, the information obtained is also less valuable. FIR concludes that variables 1 and 6 can be safely discarded, whereas all other variables need to be retained for the time being. All static models found are of fairly low quality, and the MSE values resulting from their use are consequently rather high.

Notice that, in the FIR methodology, the quality of masks can only be truly compared to each other as long as their complexities are the same. Moreover for a given mask, the MSE value depends on the validation data set, and both considerations must be taken into account when attempting to extract conclusions from Table I-II.

I.3 Garbage incinerator system

Another of the processes analysed in this work is a thermal incinerator. The description given here is a general description of the functioning of these kinds of systems [Valle *et al.*, 1999]. In this unit, high temperature and chemical reactions burn up process fumes and change them into harmless carbon dioxide and water vapours, which are then released through a stack into the atmosphere. The fumes processed in the incinerator are commonly called NO_x fumes, and mainly consist of: nitric oxide (NO), nitrogen dioxide (NO_2), and nitrous oxide (N_2O). Hydrogen Cyanide (HCN) is also present. The gases are converted into nitrogen and water vapour in a three-stage combustion process. The first stage is reduction, the second is re-oxidation, and the third is catalytic oxidation. A de-mineralised water heat exchanger is located between the second and third combustion stages.

Data were gathered from the system at a sampling rate of 1 minute, and 43200 data points were recorded. The considered output variable for this study is the emission of NOx gas. Table I-III gives a short description of the variables in the system that this paper deals with. Figure 1 shows a general diagram describing the layout of the system.

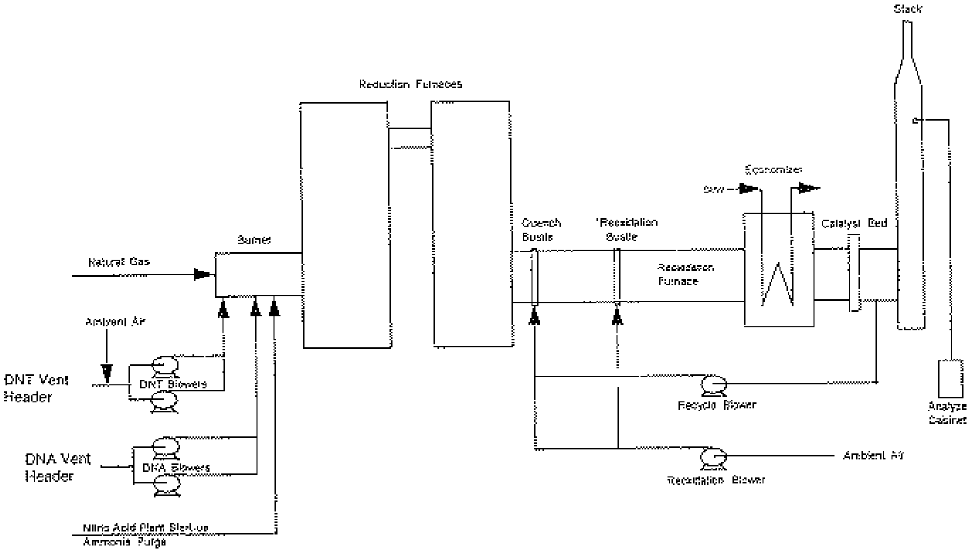


Figure I-9 Incineration process scheme

Variable	Physical meaning	Variable	Physical meaning
1 (input)	DNT unit one vent (SCFM)	11 (input)	Column B top pressure (inch W)
2 (input)	DNT unit two vent (SCFM)	12 (input)	Column C top pressure (inch W)
3 (input)	A column overhead (F)	13 (input)	Column D top pressure (inch W)
4 (input)	C column overhead (F)	14 (input)	DP on A column (PSIG)
5 (input)	DNT vent rate (SCFM)	15 (input)	DP on B column (PSIG)
6 (input)	Stack gas recycle (ACFM)	16 (input)	DP on C column (PSIG)
7 (input)	Stack gas recycle (F)	17 (input)	DP on D column (PSIG)
8 (input)	Strip vent rate (SCFM)	18 (input)	Excess O2 in stack (%)
9 (input)	DNT vent head VA (WC)	19 (input)	Reduction furnace (F)
10 (input)	Column A top pressure (inch W)	20 (output)	NOx, (PPM)

Table I-III Garbage incinerator system variables.

The reduction section is a large natural gas (methane) furnace. The air needed for the combustion is provided by either a vent header (DNT) or a stripper vent header (DNA). The burning of natural gas provides all the heat required for the reduction stage. There are four major chemical reactions that occur in the reduction section furnace:

1. Burning of natural gas $CH_4 + 2O_2 \Rightarrow 2H_2O + CO_2$

2. NO_x gas is destroyed by $CH_4 + 2NO_2 \Rightarrow N_2 + 2H_2O + CO_2$

3. Additional fuel is added without air to increase the reaction between natural gas and the NO_x, and to remove any trace of oxygen in the system

$$CH_4 + 4NO => 2N_2 + 2H_2O + CO_2$$

4. CO_2 gas is destroyed by $CH_4 + CO_2 => 2CO + 2H_2$

The hot gases leaving the reduction furnace thus consist of: carbon monoxide, nitrogen, hydrogen, water vapour, carbon dioxide, NO_x (below 200 ppm), and HCN (below 500 ppm). After the reduction furnace, hot gases are quenched, by mixing them with cooler recycled stack gas. The quenched gas then flows into the stage section called the re-oxidation stage. This section is not used for NO_x abatement, but to convert CO and H_2, the combustibles from the reduction furnace into CO_2 and water vapour. In this section, fuel gas is not used, and a blower adds ambient air to the section. Two additional reactions occur:

$$2CO + 2O_2 => O_2 + 2CO_2 \qquad 2H_2 + O_2 => 2H_2O$$

The hot gases from the re-oxidation section then flow through an economiser where they are cooled by de-mineralising water. The hot de-mineralised water (DMW) is then routed to the plant. Gases exiting the economiser flow through a honeycomb grid of a platinum catalyst, where the CO and the organics are converted to inert flue gases before their discharge from the stack to the atmosphere:

$$CO + 2HCN + C_6H_5CH_3 + O_2 => CO_2 + H_2O + N_2$$

Analyser probes in the stack monitor the O_2, CO, CO_2, and NO_x levels.

I.4 Gas turbine for electric power generation

In this appendix, a full list of the variables of the gas turbine system, analysed in Chapter 6, is given.

Variable name	Description
96HQ1 (bar)	Hydraulic circuit pressure
ACCUM_1_LSW(logic)	Manually initiated starts counter lsw
ACCUM_1_MSW(logic)	Manually initiated starts counter msw
ACCUM_12_LSW(logic)	Total fired hours lsw
ACCUM_12_MSW(logic)	Total fired hours msw
ACCUM_13_LSW(logic)	Peak fired hours lsw
ACCUM_13_MSW(logic)	Peak fired hours msw
ACCUM_14_LSW(logic)	Fired hours – gas fuel lsw
ACCUM_14_MSW(logic)	Fired hours – gas fuel msw
ACCUM_15_LSW(logic)	Fired hours – liquid fuel lsw
ACCUM_15_MSW(logic)	Fired hours – liquid fuel msw
ACCUM_2_LSW(logic)	Total starts counter lsw
ACCUM_2_MSW(logic)	Total starts counter msw
ACCUM_3_LSW(logic)	Fast load starts counter lsw

ACCUM_3_MSW(logic)	Fast load starts counter msw
ACCUM_4_LSW(logic)	Fired starts counter lsw
ACCUM_4_MSW(logic)	Fired starts counter msw
ACCUM_5_LSW(logic)	Emergency trips counter lsw
ACCUM_5_MSW(logic)	Emergency trips counter msw
ACCUM_6_LSW(logic)	Spare accumulator lsw
ACCUM_6_MSW(logic)	Spare accumulator msw
ACCUM_7_LSW(logic)	Spare accumulator lsw
ACCUM_7_MSW(logic)	Spare accumulator msw
ACCUM_8_LSW(logic)	Spare accumulator lsw
ACCUM_8_MSW(logic)	Spare accumulator msw
ACCUM_9_LSW(logic)	Spare accumulator lsw
ACCUM_9_MSW(logic)	Spare accumulator msw
AFPAP (mm Hg)	Ambient barometric pressure
AFPBD (mm H_2O)	Compressor differential pressure
AFPCD (bar)	Compressor discharge pressure
AFPCS (mm H_2O)	Inlet duct differential pressure
AFPEP (mm H_2O)	Exhaust duct pressure
AFQ (Kg/s)	Compressor inlet air flow
AFQD (Kg/s)	Compressor inlet dry airflow
ATID (deg C)	Inlet air heating thermocouple
BB_MAX (mm/s)	Vibration max select
BB1 (mm/s)	Vibration transducer #1 (turbine)
BB10 (mm/s)	Vibration transducer #10 (generator)
BB11 (mm/s)	Vibration transducer #11 (generator)
BB12 (mm/s)	Vibration transducer #12
BB2 (mm/s)	Vibration transducer #2 (turbine)
BB3 (mm/s)	Vibration transducer #3 (turbine)
BB4 (mm/s)	Vibration transducer #4 (turbine)
BB5 (mm/s)	Vibration transducer #5 (turbine)
BB6 (mm/s)	Vibration transducer #6
BB7 (mm/s)	Vibration transducer #7 (load gear)
BB8 (mm/s)	Vibration transducer #8 (load gear)
BB9 (mm/s)	Vibration transducer #9 (load gear)
BTGJ1 (deg C)	Bearing metal temperature generator journal #1
BTGJ2 (deg C)	Bearing metal temperature generator journal #2
C48DSX (logic)	Diesel cooldown counter
CAGV (%)	IGV control servo current
CMHUM (kg/kg)	Specific humidity
CNCF	Speed correction factor
CPD (bar)	Compressor discharge pressure
CQTC	Compressor airflow temperature correction
CSGRVOUT	Gas turbine inlet guide vane servo valve
CSGV	Gas turbine inlet guide vane lvdt – feedback
CSRIH (%)	Inlet heating
CSRIH1 (%)	Inlet bleed heat setpoint
CT_BIAS (deg C)	Compressor inlet temperature bias
CTD (deg C)	Compressor discharge temperature
CTDA1 (deg C)	Compressor temperature thermocouple #1
CTDA2 (deg C)	Compressor temperature thermocouple #2
CTIF1 (deg C)	Compressor inlet flange temperature – thermocouple #1
CTIF2 (deg C)	Compressor inlet flange temperature – thermocouple #2
CTIM (deg C)	Max compressor inlet flange temperature
DF (Hz)	Generator frequency
DTGGC10 (deg C)	Generator cold gas rtd

DTGGC11 (deg C)	Generator cold gas rtd
DTGGH18 (deg C)	Generator hot gas rtd
DTGGH19 (deg C)	Generator hot gas rtd
DV (KV)	Generator line voltage
DVAR (Mvar)	Generator load vars (scaled)
DWATT (MW)	Generator load watts
DWATT_NOX (Kg/s)	MW based WLNOX injection flow reference input
FAG (%)	Gas control valve servo current
FAGR (%)	Speed ratio valve servo current
FAL (%)	Liquid fuel bypass valve servo current
FD_INTENS_1 (logic)	TCEA flame uv counter 1
FD_INTENS_2 (logic)	TCEA flame uv counter 2
FD_INTENS_3 (logic)	TCEA flame uv counter 3
FD_INTENS_4 (logic)	TCEA flame uv counter 4
FPG2 (bar)	Intervalve gas fuel pressure input
FPKGNG (bar/%)	Fuel gas pressure ratio control gain
FPKGNO (bar)	Fuel gas pressure ratio control offset
FPRGOUT (bar)	Gas ratio valve servo command
FQG (Kg/s)	Gas fuel flow
FQG1 (Kg/s)	Gas fuel flow right sensor
FQG2 (Kg/s)	Gas fuel flow left sensor
FQL1 (%)	Liquid fuel flow magnetic pickup input
FQLM1 (Kg/s)	Liquid fuel mass flow, LFS
FQROUT (%)	Liquid fuel flow reference
FSG (%)	Gas control valve lvdt position feedback
FSGR (%)	Speed ratio valve lvdt position feedback
FSR (%)	Fuel stroke reference
FSR_CONTROL (logic)	Fuel control enumerated state
FSR1 (%)	Liquid fuel stroke reference from fuel splitter
FSR2 (%)	Gas fuel stroke reference from fuel splitter
FSRACC (%)	FSR acceleration control
FSRMAN (%)	FSR manual control
FSRMIN (%)	FSR minimum
FSRN (%)	Speed control fuel stroke reference
FSRNH (%)	FSR hp speed control
FSROUT (%)	Gas control valve position output
FSRSD (%)	Shutdown FSR signal
FSRSU (%)	FSR start up control
FSRT (%)	Temperature control fuel stroke reference
IT_PROP (%)	Calculated inlet heating reference
ITPD (deg C)	Anti icing dew point temperature
L14DMY (logic)	Diesel minimum speed aux timer logic
L14HA (logic)	HP accelerating speed signal
L14HM (logic)	Minimum speed signal
L14HR (logic)	HP zero speed signal
L14HS(logic)	HP operating speed signal
L14HSX (logic)	Auxiliary signal to L14HS
L14P1 (logic)	Starting device above min speed
L14P2 (logic)	Starting device above cranking speed
L20CS1X (logic)	Starting clutch solenoid driver
L20DAR1 (logic)	Diesel accelerating 4 way solenoid driver
L26DT1H (logic)	Gas turbine diesel engine water temperature high
L2DW (logic)	Diesel warm up timer
L2DW1 (logic)	Diesel accelerate timer
L33CSE (logic)	Starting clutch engaged

L33HRF (logic)	Hydraulic ratchet in forward stroke
L48DS (logic)	Diesel incomplete sequence timer
L48DSX (logic)	Diesel starts counter
L4DE (logic)	Diesel engine control signal
L4DEY (logic)	TD loss of diesel master control
L4DEZ (logic)	Diesel stop
L4DS (logic)	Perm to start diesel engine starter
L4HR (logic)	Perm to start hydraulic ratchet pump
L60BOG (logic)	Starting device bogged down
L63QDN (logic)	Diesel lube oil pressure normal
L72HR (logic)	Lube oil starts counter
L90PSEL (MW)	Preselected load analog setpoint
LK90PSR (Meg/s)	Preselect load ramp rate
LTB1D (deg C)	Lube oil temperature – brg. drain #1
LTB2D (deg C)	Lube oil temperature – brg. drain #2
LTBT1D (deg C)	Lube oil temperature – thrust drain #1
LTE1D (deg C)	Lube oil temperature – exciter brg. drain #1
LTG1D (deg C)	Lube system temperature – generator drain #1
LTG2D (deg C)	Lube system temperature – generator drain #2
LTTH1 (deg C)	Lube oil temperature – turbine header
PN (%)	Turbine starting device speed
PO (bar)	Liquid fuel pressure
SDSJ1 (mm H_2O)	Steam injection low-range differential pressure
SDSJ2 (mm H_2O)	Steam injection high-range differential pressure
SFL2 (Hz)	Ioma bus_pt frequency
SPSJ1 (bar)	Steam injection supply pressure
SS43	Command state
SS43F	Fuel selection enumerated state variable
SS43LOAD	Load selection enumerated state variable
SS43SYNC	Sync control selection
STSJ (deg C)	Steam injection temperature
SVL (KV)	System line voltage
SVP	Stop fuel control valve position
TGSDIFU1 (deg C)	Generator U phase winding temperature. #1, main input exciter
TGSDIFU2 (deg C)	Generator U phase winding temperature. #2, centre
TGSDIFV1 (deg C)	Generator V phase winding temperature. #1, exciter end
TGSDIFV2 (deg C)	Generator V phase winding temperature. #2, centre
TGSDIFW1 (deg C)	Generator W phase winding temperature. #1, exciter end
TGSDIFW2 (deg C)	Generator W phase winding temperature. #1, centre
TIFDP1 (mm H_2O)	Gas turbine inlet filter differential pressure, #2
TIFDP2 (mm H_2O)	Gas turbine inlet filter differential pressure, #3
TIFDP3 (mm H_2O)	Gas turbine inlet filter differential pressure, #4
TNH (m/rad)	Turbine rotation speed
TNR (%)	Speed control reference
TOP	Trip oil pressure
TTRX (deg C)	Temperature control reference
TTWS1AO1 (deg C)	Turbine wheelspace temperature 1st stage aft. outer, #1
TTWS1AO2 (deg C)	Turbine wheelspace temperature 1st stage aft. outer, #2
TTWS1FI1 (deg C)	Turbine wheelspace temperature 1st stage fwd. inner, #1
TTWS1FI2 (deg C)	Turbine wheelspace temperature 1st stage fwd. inner, #2
TTWS1FO1 (deg C)	Turbine wheelspace temperature 1st stage fwd. outer, #1
TTWS1FO2 (deg C)	Turbine wheelspace temperature 1st stage fwd. outer, #2
TTWS2AO1 (deg C)	Turbine wheelspace temperature 2st stage aft. outer, #1
TTWS2AO2 (deg C)	Turbine wheelspace temperature 2st stage aft. outer, #2
TTWS2FO1 (deg C)	Turbine wheelspace temperature 2st stage fwd. outer, #1

TTWS2FO2 (deg C)	Turbine wheelspace temperature 2st stage fwd. outer, #2
TTWS3AO1 (deg C)	Turbine wheelspace temperature 3st stage aft. outer, #1
TTWS3AO2 (deg C)	Turbine wheelspace temperature 3st stage aft. outer, #2
TTWS3FO1 (deg C)	Turbine wheelspace temperature 3st stage fwd. outer, #1
TTWS3FO2 (deg C)	Turbine wheelspace temperature 3st stage fwd. outer, #2
TTXD_1 (deg C)	Exhaust temperature thermocouple #1
TTXD_10 (deg C)	Exhaust temperature thermocouple #10
TTXD_11 (deg C)	Exhaust temperature thermocouple #11
TTXD_12 (deg C)	Exhaust temperature thermocouple #12
TTXD_13 (deg C)	Exhaust temperature thermocouple #13
TTXD_14 (deg C)	Exhaust temperature thermocouple #14
TTXD_15 (deg C)	Exhaust temperature thermocouple #15
TTXD_16 (deg C)	Exhaust temperature thermocouple #16
TTXD_17 (deg C)	Exhaust temperature thermocouple #17
TTXD_18 (deg C)	Exhaust temperature thermocouple #18
TTXD_2 (deg C)	Exhaust temperature thermocouple #2
TTXD_3 (deg C)	Exhaust temperature thermocouple #3
TTXD_4 (deg C)	Exhaust temperature thermocouple #4
TTXD_5 (deg C)	Exhaust temperature thermocouple #5
TTXD_6 (deg C)	Exhaust temperature thermocouple #6
TTXD_7 (deg C)	Exhaust temperature thermocouple #7
TTXD_8 (deg C)	Exhaust temperature thermocouple #8
TTXD_9 (deg C)	Exhaust temperature thermocouple #9
TTXD2_1 (deg C)	Exhaust temperature thermocouple sorted by temperature value
TTXD2_18 (deg C)	Exhaust temperature thermocouple sorted by temperature value
TTXM (deg C)	Exhaust temperature median
TTXSP1 (deg C)	Combustion monitor actual spread, #1
TTXSP2 (deg C)	Combustion monitor actual spread, #2
TTXSP3 (deg C)	Combustion monitor actual spread, #3
TTXSPL (deg C)	Combustion monitor allowable spread
WQJ (Kg/s)	Wet low NOX injection flow
WQJR2REF (Kg/s)	Wet low NOX injection reference
WQR1 (Kg/s)	Required wet low NOX injection flow
WQR2 (Kg/s)	Wet low NOX injection reference
WXC	Required WLNOX injection to fuel ratio
WXJ	Actual WLNOX injection to fuel ratio

II Assessment of results

II.1 Application of the algorithm discussed in Section 2.4.1 to the garbage incinerator system

This appendix presents an application of the new suboptimal mask search algorithm, detailed in Section 2.4.1, for Fuzzy Inductive Reasoning (FIR).

The modelling engine of FIR determines a so-called *optimal mask*, that indicates, which variables best explain any given output, and how much time delay these variables should have relative to the chosen output. Unfortunately, any algorithm that can find the optimal mask is necessarily of exponential complexity, i.e., the number of masks to be visited during the search for the optimal mask grows exponentially with the number of available input variables, and with the allowed depth of the mask. For this reason, suboptimal search algorithms are needed for dealing with large-scale systems.

The used technique operates on the already recoded, i.e., *qualitative*, data. It starts with a mask depth of *1* and performs an exhaustive search. It subsequently increments the depth of the mask, while making use of the results of the previous searches to select, which elements of the mask candidate matrix can be eliminated from the list of potential inputs. Instead of solving one optimal search with a mask candidate matrix of size $d{\cdot}n$, it solves d separate optimal searches of sizes n, $2n$, …, $d{\cdot}n$, yet with ever more sparsely populated mask candidate matrices. The maximal mask depth d does not need to be pre-selected. The algorithm proposes an optimal value for d.

II.1.1 Reducing the mask search space using qualitative data

As has been outlined in Appendix I.3 for the incinerator system, 19 input variables and one output variable are taken into account. The task to be accomplished is to derive a qualitative model of this process with high predictive power. The process was sampled in 1-minute intervals. 43,200 data records were obtained, representing a time period of 30 days. The first 85% of these data records were used to perform the variable selection with the two previously described suboptimal search procedures. The remaining 15% of the data records were used to validate the obtained model.

Hence, the system under study has 20 variables leading to a large search space of possible masks when an exhaustive search is applied to all possible combinations of input variables. Following the procedure for selecting variables outlined in Section 2.4.1, the first mask candidate matrix proposed is of depth $d{=}1$, so only static models are being considered. As, at this stage, the number of masks to be evaluated is still fairly low, the maximum complexity is set to 6. Potential inputs of the proposed mask are set to '–1,' except for the output element, which is set to '+1,' so that all the possible masks are being considered. The corresponding mask candidate matrix is:

$$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & +1 \end{pmatrix}$$

The number of '-1' elements of the mask is 19. Table II-I presents, for each allowed complexity, the variables that are used as inputs by the *good masks*, i.e., the masks with $Q>0.975Q_{best}$.

	Q_{best}	Variables used in masks with $Q>0.975Q_{best}$	Number of masks with $Q>0.975Q_{best}$
C2	.0214	19	1
C3	.0643	5,19	1
C4	.1185	5,6,19	1
C5	.1773	2,4,5,6,9,19	3
C6	.2170	1,4,5,6,7,9,12,19	4

Table II-I Static model search results

None of these masks is particularly good. The best masks are those of complexity 6, but even they are of relatively low quality. Evidently, none of the static models will do a very good job at predicting the output.

Proceeding to the second step of the algorithm of Section 2.4.1, a mask candidate matrix of depth *d=2* is now proposed. Its lower row, i.e., the row corresponding to time *t*, that is representing inputs without time delay relative to the output, obtains '-1' elements only in those positions, where *significant inputs* were discovered in the previous step of the algorithm, i.e., in columns 1, 2, 4, 5, 6, 7, 9, 12, and 19. Since no information is available with respect to the upper row, all of its elements are set to -1. The proposed mask candidate matrix thus takes the form:

$$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 0 & -1 & -1 & -1 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 \end{pmatrix}$$

The new mask candidate matrix has 29 '-1' elements. Since there are now more possible masks, it was decided to reduce the maximum allowed complexity from 6 to 5. Table II-II shows the inputs used by the *good masks* for each of the allowed mask complexities. Since every mask includes the output variable, the last row shows, how many good masks were found for each of the allowed complexities. There were 25 good masks of complexity 3, 313 good masks of complexity 4, and 2009 good masks of complexity 5. The masks of complexity 2 were not presented, because they are trivial. The only good mask of complexity two is the one that uses $y(t\text{-}\delta t)$ as its input. The qualities of the good masks of depth *d*=2 are considerably higher than of the good static masks, because now, the output to be predicted, $y(t)$, may depend on its own past, i.e., the value of the output one sample back, $y(t\text{-}\delta t)$, which helps a lot with the prediction.

Every single one of the good masks makes use of $y(t\text{-}\delta t)$ as an input. Therefore, the values in the "delay-1" columns of the row entitled "y" are the same as the values in the "delay-0" columns of the same row, which denote the output itself. This search step indicates that most of the 20 variables one time-step back are of similar usefulness for predicting the output, as they are used almost uniformly by the good masks. There exists a much better discrimination concerning the usefulness of the variables at time *t*.

	Complex. 3 (Q_{best}= 0.5975)		Complex. 4 (Q_{best}= 0.5915)		Complex. 5 (Q_{best}= 0.5927)	
	$Q>0.975Q_{best}$ masks		$Q>0.975Q_{best}$ masks		$Q>0.975Q_{best}$ masks	
Delay	0	1	0	1	0	1
x_1	1	1	24	24	214	208
x_2	1	1	26	26	232	236
x_3	—	1	—	21	—	200
x_4	1	1	22	22	229	226
x_5	1	1	21	21	195	198
x_6	1	1	26	26	247	241
x_7	1	1	26	26	238	237
x_8	—	1	—	26	—	241
x_9	—	—	11	10	118	123
x_{10}	—	—	—	11	—	98
x_{11}	—	1	—	20	—	184
x_{12}	1	1	25	27	238	239
x_{13}	—	1	—	24	—	225
x_{14}	—	1	—	24	—	208
x_{15}	—	1	—	24	—	225
x_{16}	—	1	—	23	—	224
x_{17}	—	1	—	22	—	219
x_{18}	—	1	—	16	—	187
x_{19}	1	1	26	26	302	295
y	25	25	313	313	2009	2009

Table II-II Good masks obtained by a depth-2 model search.

For each complexity separately, only those inputs are considered *significant inputs* that are present in at least 10% of the good masks of that complexity. Using this heuristic rule, a mask candidate matrix of depth *d=3* can now be constructed. Since no information is available about the usefulness of any of the variables two time-steps back, the top row of the new mask candidate matrix must be filled with '-1' elements.

$$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 \end{pmatrix}$$

The new mask candidate matrix contains 41 out of 59 possible '-1' elements. An optimal model search is now performed, proposing this mask candidate matrix to FIR. Table II-III shows the results of this search. The qualities of the best masks of each complexity are only slightly higher than in the case of the depth-2 models. 4 good masks of complexity 3, 254 good masks of complexity 4, and 4047 good masks of complexity 5 were found. At this point, FIR shows preferences for some variables one time-step back over others. The reason is that FIR now has more choices, and often prefers the same variable two time-steps back.

The way the algorithm is implemented, once an input variable at a certain time delay has been eliminated from one of the mask candidate matrices, it will never show up again in any of the subsequent mask candidate matrices. Therefore, the *discriminator* value (in the example shown here set to 10%) must be chosen carefully, in order not to eliminate potentially useful inputs too early. A smaller discriminator might generate a better suboptimal mask at the end, but this goes at the expense of having to evaluate more masks in the process. A higher discriminator value leads to a faster search, but may result in a suboptimal mask of lower quality.

	Complex. 3 (Q_{best}= 0.6113) $Q>0.975Q_{best}$ masks			Complex. 4 (Q_{best}= 0.5996) $Q>0.975Q_{best}$ masks			Complex. 5 (Q_{best}= 0.5931) $Q>0.975Q_{best}$ masks		
Delay	0	1	2	0	1	2	0	1	2
X_1	—	—	—	17	17	17	—	307	322
X_2	1	1	1	24	24	24	405	418	426
X_3	—	—	—	—	—	1	—	—	148
X_4	—	—	—	1	1	1	225	239	250
X_5	—	—	—	—	—	1	—	—	200
X_6	—	—	—	20	20	20	379	415	409
X_7	—	—	—	22	22	22	417	410	408
X_8	—	—	—	—	8	7	—	331	326
X_9	—	—	—	—	—	—	—	—	58
X_{10}	—	—	—	—	—	1	—	—	61
X_{11}	—	—	—	—	—	1	—	—	175
X_{12}	—	—	—	17	20	20	340	410	415
X_{13}	—	—	—	—	20	20	—	406	408
X_{14}	—	—	—	—	4	4	—	237	232
X_{15}	—	—	—	—	14	14	—	304	324
X_{16}	—	—	—	—	1	1	—	257	266
X_{17}	—	—	—	—	1	1	—	203	210
X_{18}	—	—	—	—	—	1	—	—	108
X_{19}	—	—	—	20	20	21	530	530	536
y	4	4	1	254	254	38	4047	4047	154

Table II-III Good masks obtained by a depth-3 model search.

Using the information provided in Table II-III, a new mask candidate matrix of depth *d=4* can now be proposed.

$$mcan = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 \end{pmatrix}$$

In this new candidate mask only 37 -1 elements out of 79 possible are used, so leading to a significant reduction of the model search space. This depth-4 candidate mask is given as an input to the modelling engine of FIR, which performs an optimal search in this reduced model search space. Results of this simulation are given in Table II-IV. Again, a slight increase in the mask qualities is found with respect to the case of the depth-3 models. Models with complexities 2 and 3 are not reported in Table II-IV. The reason for that is that they are trivial autoregressive models. For the complexity-2 model, the optimal quality was *0.6128* for a mask in which the output depends on itself one time step in the past, i.e., $y = f(y\text{-}\delta t)$. In the case of complexity 3, $Q_{best} = 0.6230$, and two masks are found that accomplishes the condition $Q>0.975Q_{best}$: the first one models the output variable as $y(t) = f(y(t\text{-}1),y(t\text{-}3))$, and the second one uses the input-output model $y(t) = f(y(t\text{-}2),y(t\text{-}3))$.

	Complex. 4 (Q_{best}= 0.6127) Q>0.975Q_{best} masks				Complex. 5 (Q_{best}= 0.6088) Q>0.975Q_{best} masks			
Delay	0	1	2	3	0	1	2	3
X_1	—	—	—	1	—	—	—	30
X_2	2	2	2	2	30	30	30	30
X_3	—	—	—	1	—	—	—	29
X_4	—	—	—	1	—	—	—	27
X_5	—	—	—	1	—	—	—	26
X_6	—	1	1	1	—	31	26	29
X_7	2	2	2	2	29	30	30	30
X_8	—	—	—	1	—	—	—	32
X_9	—	—	—	—	—	—	—	20
x_{10}	—	—	—	1	—	—	—	25
x_{11}	—	—	—	1	—	—	—	26
x_{12}	—	1	1	1	—	29	27	29
x_{13}	—	1	1	1	—	29	30	29
x_{14}	—	—	—	1	—	—	—	31
x_{15}	—	—	—	1	—	—	—	26
x_{16}	—	—	—	1	—	—	—	27
x_{17}	—	—	—	1	—	—	—	28
x_{18}	—	—	—	1	—	—	—	27
x_{19}	1	1	1	1	30	30	30	30
Y	41	41	8	33	487	487	—	487

Table II-IV Good masks obtained by a depth-4 model search.

The information contained in Table II-IV is used to derive a new depth-5 candidate matrix. In this case, the discriminator value (up to now, it has been set to 10%) has to be changed in order not to eliminate potentially useful inputs. Note that if a value of 10% is used for the discriminator parameter no input variables are selected as possible *m*-inputs to the next model search. Hence, for each complexity separately, only those inputs are considered *significant inputs* that are present in at least 6% of the good masks of that complexity. This discriminator value keeps the variables that seem to be more important in the analysis performed up to this point, yet conserves a reduced search space model. Using this heuristic rule, a mask candidate matrix of depth *d=5* can now be constructed.

$$mcan = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

This new proposed candidate mask has 46 -1 elements. Results are given in Table II-V. The best complexity 2 mask is the same that in the previous computation and it will be the same all over the next runs to be performed, so it will not be named again. The models of complexity 3 that satisfies the conditions of the used algorithm explain the output with a maximum found quality of $Q_{best} = 0.6272$ as *y(t) = f(y(t-1), y(t-3))* and *y(t) = f(y(t-1), y(t-4))*. Again a slight overall increase in the mask quality is obtained.

Notice that, as the procedure is proceeding, the input variable space is decomposed into subspaces of more and less related input variables to the output, so allowing a simplification of the model search.

	Complex. 4 (Q_{best}= 0.6257) $Q>0.975Q_{best}$ masks					Complex. 5 (Q_{best}= 0.6146) $Q>0.975Q_{best}$ masks				
Delay	0	1	2	3	4	0	1	2	3	4
x_1	—	—	—	1	1	—	—	—	70	73
x_2	2	2	2	2	2	73	73	73	74	73
x_3	—	—	—	—	—	—	—	—	49	44
x_4	—	—	—	—	—	—	—	—	—	45
x_5	—	—	—	—	—	—	—	—	—	57
x_6	—	1	—	1	1	—	48	—	73	74
x_7	1	1	1	1	1	74	74	74	74	74
x_8	—	—	—	—	—	—	—	—	70	73
x_9	—	—	—	—	—	—	—	—	—	29
x_{10}	—	—	—	—	—	—	—	—	—	35
x_{11}	—	—	—	—	—	—	—	—	—	54
x_{12}	—	1	—	1	1	—	60	—	74	63
x_{13}	—	1	1	1	1	—	73	74	59	75
x_{14}	—	—	—	—	—	—	—	—	73	65
x_{15}	—	—	—	—	—	—	—	—	—	52
x_{16}	—	—	—	—	—	—	—	—	—	58
x_{17}	—	—	—	—	—	—	—	—	—	53
x_{18}	—	—	—	—	—	—	—	—	—	56
x_{19}	1	1	1	1	1	75	—	74	74	74
y	34	34	1	6	29	1228	1228	—	503	729

Table II-V Good masks obtained by a depth-5 model search.

Using the information of Table II-V, again with the discriminator value set to the 6%, the following depth-6 candidate mask is proposed for the next model computation:

$$mcan = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

This new candidate mask has again 46 -1 elements. Results obtained with this mask are given in Table II-VI.

	Complex. 4 (Q_{best}= 0.6301) Q>0.975Q_{best} masks						Complex. 5 (Q_{best}= 0.6228) Q>0.975Q_{best} masks					
Delay	0	1	2	3	4	5	0	1	2	3	4	5
x_1	—	—	—	—	1	1	—	—	—	—	60	59
x_2	2	2	2	2	2	2	68	72	69	70	74	73
x_3	—	—	—	—	—	—	—	—	—	—	—	8
x_4	—	—	—	—	—	—	—	—	—	—	—	26
x_5	—	—	—	—	—	—	—	—	—	—	—	20
x_6	—	—	—	1	1	1	—	—	—	43	48	43
x_7	2	2	2	2	2	2	60	60	64	62	62	63
x_8	—	—	—	—	—	—	—	—	—	—	41	64
x_9	—	—	—	—	—	—	—	—	—	—	—	1
x_{10}	—	—	—	—	—	—	—	—	—	—	—	1
x_{11}	—	—	—	—	—	—	—	—	—	—	—	19
x_{12}	—	—	—	1	—	1	—	—	—	64	—	53
x_{13}	—	1	1	—	1	1	—	64	65	—	65	53
x_{14}	—	—	—	—	—	—	—	—	—	44	—	39
x_{15}	—	—	—	—	—	—	—	—	—	—	—	36
x_{16}	—	—	—	—	—	—	—	—	—	—	—	43
x_{17}	—	—	—	—	—	—	—	—	—	—	—	30
x_{18}	—	—	—	—	—	—	—	—	—	—	—	7
x_{19}	1	—	1	1	1	1	60	—	67	63	65	64
y	43	43	—	2	14	30	1080	1080	—	104	419	605

Table II-VI Good masks obtained by a depth-6 model search.

The obtained complexity-3 models that comply with the imposed conditions are simple autoregressive models with $Q_{best} = 0.6384$. These models are:

$$y(t) = f(y(t\text{-}1), y(t\text{-}3)); \quad y(t) = f(y(t\text{-}1), y(t\text{-}4)); \quad y(t) = f(y(t\text{-}1), y(t\text{-}5))$$

In this new simulation, a better discrimination of the variables is obtained. For example, the models satisfying the conditions of the algorithm have used variables 9 and 10 only once, which says that those variables are not important to model the output. The discriminator value used is again 6%. Therefore, variables used more than twice, in the case of the complexity-4 models, and more than 64 times for complexity-5 models, are selected to propose the next depth-7 candidate mask.

$$mcan = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

	Complex. 4 (Q_{best}= 0.6319) $Q>0.975Q_{best}$ masks							Complex. 5 (Q_{best}= 0.6250) $Q>0.975Q_{best}$ masks						
Delay	0	1	2	3	4	5	6	0	1	2	3	4	5	6
x_1	—	—	—	—	—	—	—	—	—	—	—	—	—	75
x_2	3	3	3	3	3	3	3	72	73	73	71	77	76	73
x_3	—	—	—	—	—	—	—	—	—	—	—	—	—	3
x_4	—	—	—	—	—	—	—	—	—	—	—	—	—	3
x_5	—	—	—	—	—	—	—	—	—	—	—	—	—	3
x_6	—	—	—	—	—	—	—	—	—	—	—	—	—	51
x_7	1	3	3	3	3	3	1	61	60	61	64	67	64	62
x_8	—	—	—	—	—	—	—	—	—	—	—	—	54	54
x_9	—	—	—	—	—	—	—	—	—	—	—	—	—	3
x_{10}	—	—	—	—	—	—	—	—	—	—	—	—	—	3
x_{11}	—	—	—	—	—	—	—	—	—	—	—	—	—	3
x_{12}	—	—	—	—	—	—	—	—	—	—	65	—	—	50
x_{13}	—	—	—	—	—	—	—	—	64	67	—	64	—	67
x_{14}	—	—	—	—	—	—	—	—	—	—	—	—	—	25
x_{15}	—	—	—	—	—	—	—	—	—	—	—	—	—	53
x_{16}	—	—	—	—	—	—	—	—	—	—	—	—	—	30
x_{17}	—	—	—	—	—	—	—	—	—	—	—	—	—	3
x_{18}	—	—	—	—	—	—	—	—	—	—	—	—	—	3
x_{19}	—	—	—	—	—	—	—	—	—	75	—	75	75	75
y	48	48	—	—	1	1	1	1060	1060	—	88	288	390	417

Table II-VII Good masks obtained by a depth-7 model search.

The new candidate mask has 44 -1 elements. Results of the optimal search performed when giving this candidate mask to FIR are provided in Table II-VII.

The discriminator value has again been chosen to be 6%. In this simulation, more discrimination about variables is being obtained; note that variables x_3, x_4, x_5, x_9, x_{10}, x_{11}, x_{17}, and x_{18} have only been used 3 times each in a total of 1060 considered models. This suggests that those variables are not of big importance in modelling the output of the system. In the light of the obtained results, a new candidate mask of depth 8 is proposed:

$$mcan = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

This mask has a total of 47 -1 elements. Results for this simulation are shown in Table II-VIII.

	Complex. 4 (Q_{best}= 0.6331) $Q>0.975Q_{best}$ masks								Complex. 5 (Q_{best}= 0.6264) $Q>0.975Q_{best}$ masks							
Delay	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
x_1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	90	86
x_2	3	3	3	3	3	3	3	3	90	92	95	94	94	95	94	93
x_3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	70
x_7	—	3	3	3	3	3	—	3	—	92	92	92	91	91	—	89
x_8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	48
x_9	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	3
x_{10}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	3
x_{11}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_{12}	—	—	—	—	—	—	—	—	—	—	—	89	—	—	—	70
x_{13}	—	—	—	—	—	—	—	—	—	—	81	89	—	88	82	79
x_{14}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	6
x_{15}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	34
x_{16}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	6
x_{17}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_{18}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	6
x_{19}	—	—	—	—	—	—	—	—	—	—	95	1	94	84	94	94
y	52	52	—	4	4	18	18	18	1478	1478	—	122	320	415	410	460

Table II-VIII Good masks obtained by a depth-8 model search.

The discriminator value has again been set to 6%. As the depth is increased, more and more discrimination about variables is obtained; note that variables x_3, x_4, x_5, x_9, x_{10}, x_{11}, x_{14}, x_{16}, x_{17}, and x_{18} have only been used between 3 and 6 times each in a total of 1478 considered models. This suggests that those variables are not of big importance in modelling the output of the system. In the light of the obtained results, a new candidate mask of depth 9 is proposed:

$$mcan = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

In this new candidate mask, there are 48 elements set to ‘–1’ out of 179. The achieved simplification is considerable. Results from this simulation run are given in Table II-IX.

	Complex. 4 (Q_{best}= 0.6340) $Q>0.975Q_{best}$ masks									Complex. 5 (Q_{best}= 0.6276) $Q>0.975Q_{best}$ masks								
Delay	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8
x_1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	109	—	108
x_2	4	3	2	2	6	4	6	2	6	99	110	94	93	114	111	116	95	116
x_3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	10
x_4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	10
x_5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	9
x_6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	73
x_7	—	—	—	—	—	—	—	—	—	—	102	102	103	102	101	—	99	98
x_8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	10
x_9	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	8
x_{10}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	9
x_{11}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	10
x_{12}	—	—	—	—	—	—	—	—	—	—	—	—	107	—	—	—	—	51
x_{13}	—	—	—	—	—	—	—	—	—	—	—	—	111	—	121	—	—	107
x_{14}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	10
x_{15}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	10
x_{16}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	10
x_{17}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	10
x_{18}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	10
x_{19}	—	—	—	—	—	—	—	—	—	—	—	127	—	121	—	120	120	120
y	50	50	—	2	17	1	2	22	21	1636	1636	—	95	368	94	96	520	569

Table II-IX Good masks obtained by a depth-9 model search.

The adopted criterion for the discriminator parameter of the algorithm is the same as in the last simulations, i.e., those variables appearing in more than 6% of the total of considered masks are taken into account to form the next depth-10 candidate mask.

$$mcan = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

This new candidate mask contains 48 '-1' elements out of 199 possible. Results of this new simulation are given in Table II-X.

Note that in this simulation the overall quality of the complexity-4 masks is not increased any more with respect to the qualities obtained in previous simulations. Yet for the complexity-5 models, a slight increase in the quality of the masks is still obtained. This suggests performing another iteration, proposing a new depth-11 candidate mask (using again a discriminator value of 6%). When doing so, no increase in the quality of either the complexity-4 models or the complexity-5 models is obtained, so the iteration can be terminated. At this point, the FIR qualitative models that may best describe, in terms of quality, the system at hand are those models represented by a depth-10 mask.

	Complex. 4 (Q_{best}= 0.6340) $Q>0.975Q_{best}$										Complex. 5 (Q_{max}= 0.6312) $Q>0.975Q_{best}$									
Delay	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
x_1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	33	—	4	4
x_2	2	2	—	—	2	2	2	—	2	2	39	42	—	—	48	47	48	—	53	45
x_3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_7	—	—	—	—	—	—	—	—	—	—	—	37	37	40	39	37	—	38	36	36
x_8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_9	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	3
x_{10}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_{11}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_{12}	—	—	—	—	—	—	—	—	—	—	—	—	—	4	—	—	—	—	—	4
x_{13}	—	—	—	—	—	—	—	—	—	—	—	—	—	32	—	33	—	—	34	33
x_{14}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_{15}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_{16}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_{17}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_{18}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	4
x_{19}	—	—	—	—	—	—	—	—	—	—	—	—	33	—	33	—	7	8	47	34
y	20	20	—	—	3	—	—	10	10	3	535	535	—	—	147	—	—	137	151	148

Table II-X Good masks obtained by a depth-10 model search.

This would tell the modeller to use a depth-10 FIR model (the depth-11 model being exactly the same), in which input variables x_3, x_4, x_5, x_6, x_8, x_9, x_{10}, x_{11}, x_{14}, x_{15}, x_{16}, x_{17}, and x_{18} have been discarded. Only variables x_1, x_2, x_7, x_{12}, x_{13}, and x_{19}, and evidently past values of the output, are used to model the output variable. From those variables, the reader may notice that variable x_{12} only appears in 4 out of 535 possible models, and that, in fact, if another iteration were performed it would be eliminated from the depth-11 models. The new depth-11 candidate mask is reported below, in which there are only 49 '–1' elements out of 119 possible. The results obtained with this simulation are given in Table II-XI. For this simulation, the qualities of the masks no longer increased.

With the presented procedure, not only is a FIR model found, but with it, also the optimal depth of the model is being determined. It is interesting to analyse how large the computation alleviation is that has been achieved with this method. The achieved model search space simplification is computed by comparing the numbers of models visited when using the presented algorithm to construct depth-10 models, with the numbers of depth-10 models to be computed using the classical FIR search method[1]. Table II-XII provides the number of models that have been visited in order to construct the depth-10 models presented in Table II-X, the number of theoretical models that should be computed for a full candidate mask of depth 10, and finally, column three shows the percentage of computation alleviation. The first column is calculated by adding the number of models computed, for all the allowed complexities, from depth 1 to 10 as presented in this section. How to obtain the number of models that have to be computed for each number of '-1' elements of the candidate mask is reported in Chapter 6.

[1] This implies a prior knowledge, or a heuristic decision, of the mask depth value to be used.

$$mcan = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

	Complex. 4 (Q_{max}= 0.6377) Q>0.975Q_{max} -- 15 masks											Complex. 5 (Q_{max}= 0.6319) Q>0.995Q_{max} -- 46 masks										
Delay	0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10
x_1	—	—	—	—	—	—	—	—	—	1	—	—	—	—	—	—	—	—	—	—	—	—
x_2	—	—	—	—	—	—	—	—	1	—	—	2	—	—	—	2	2	—	—	2	2	2
x_7	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2
x_{13}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1	2	1	1
x_{19}	—	—	—	—	—	—	—	—	—	—	—	—	—	1	1	1	—	—	—	—	1	1
y	—	15	—	—	14	—	—	—	2	—	2	—	46	—	—	46	—	—	—	28	—	19

Table II-XI Mask depth 11. Simulation results.

Number of visited models $\sum_{depth=1}^{10} visited\ masks_{depth}$	Number of models to visit using a depth 10 full candidate mask	Percent of computation alleviation
1.362.501	64.704.850	97.89%

Table II-XII Computation alleviation achieved when using the proposed algorithm

Of course when giving the computation alleviation one must take into account the time required to study the computed FIR models, to construct the table of variables/delays used, and from it, propose a new candidate mask[2]. These operations have been made by means of a function code programmed in Matlab, which, from the output text file of FIR containing all computed models, generates one table for each complexity that contains the primarily used variables/delays. The computation time of this function is orders of magnitude lower than that of the FIR model computation.

II.1.2 Qualitative simulation results

In order to assess whether the sub-optimal models found with the proposed algorithm are suitable to properly model the system under consideration, two simulations have been performed. The first simulation uses one model of complexity 4 from those found with depth 9 in Table II-IX. The quality of the best model in this class was 0.6340. The model is used to simulate the last 500 points of the garbage incinerator output variable (NOx gas

[2] As was outlined in Chapter 6, the required time to analyse one FIR simulation is about 0.5 hours, so 30 minutes should be added to the CPU time required to solve each of the proposed candidate masks.

emission). Figures II-1 and II-2 show in continuous line the real NOx data and in dotted line the simulated trajectory for this variable.

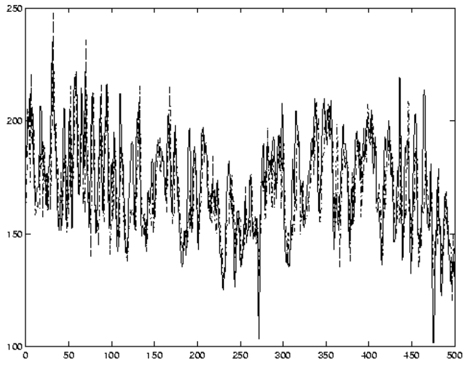


Figure II-1 Real and simulated NOx. Last 500 points.

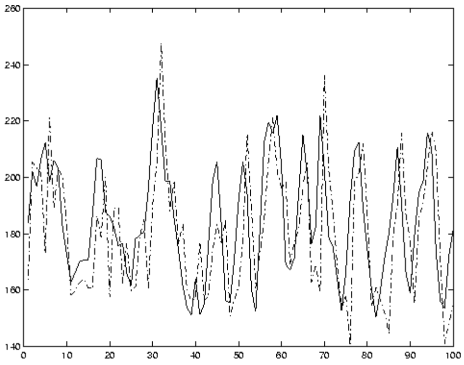


Figure II-2 First 100 NOx FIR simulated points.

In this first simulation, the used model has been:

$$y(t) = f\{y(t\text{-}1), y(t\text{-}4), x_2(t\text{-}8)\}$$

For the second simulation, a complexity-5 model has been used. In this case, the depth of the model is 10, and it is one of the models found in Table II-X. The maximum quality achieved in this case was 0.6312. The model used in this simulation is:

$$y(t) = f\{y(t\text{-}1), y(t\text{-}7), x_7(t\text{-}3), x_2(t\text{-}8)\}$$

Results of this simulation are presented in Figures II-3 and II-4, where again the continuous line represents the real data, whereas the dotted line depicts the simulated trajectory.

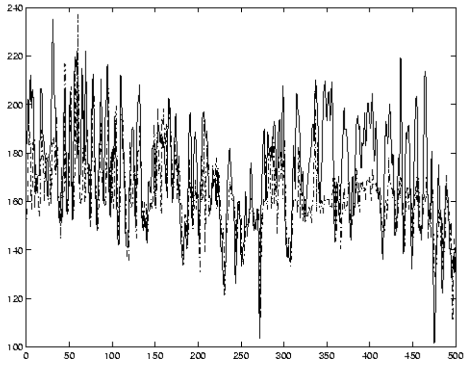


Figure II-3 Real and simulated NOx. Last 500 points.

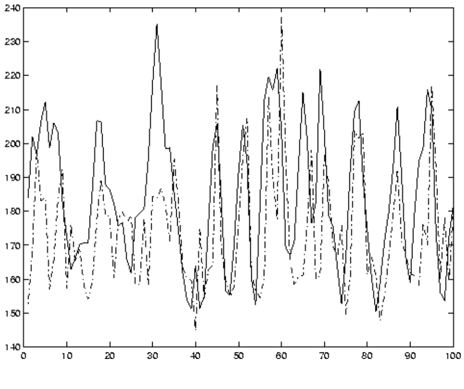


Figure II-4 First 100 NOx FIR simulated points.

As can be seen from the figures, both simulations give quite good results. They follow well the original NOx trajectory. In the first performed simulation, the average error is 1.25%; in the second case, the average error is 3.64%, whereby the averages were computed with all 500 simulated data points.

In order to assess the results obtained with the given algorithm even better, some additional simulations were performed using the best models up to depth 16. No increase in the mask qualities was obtained in any of those simulations, only a small fluctuation of the complexity-5 model quality was observed for the depth-14 models. Figure II-5 shows the mask quality plotted versus the depth of the mask. The continuous line is obtained for complexity-4 masks, and the dotted line represents complexity-5 masks. The qualities of the depth-1 static models are not presented, in order to avoid an unnecessary compression of the y-axis.

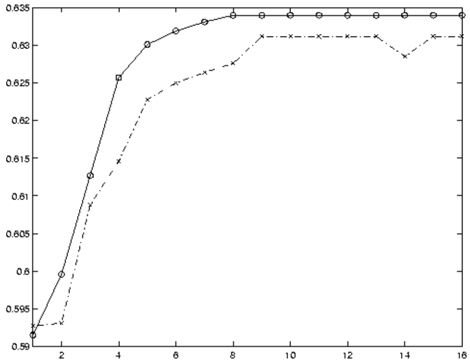


Figure II-5 Quality versus depth of the mask

II.2 Application of the accep tability interval to fault detection in a commercial aircraft

When modelling a system, it is possible to use a mixed quantitative/qualitative knowledge representation [Cellier *et al.*, 1992] that may combine the advantages of both types of approaches and may help to solve problems that neither a purely quantitative nor a strictly qualitative model may be able to solve on its own. Modern approaches to fault diagnosis include the so-called model based approach [Chen and Patton, 1998]. This kind of fault detection approach makes explicit use of a quantitative mathematical model of the system. The implementation of on-board digital computers allows the development of fault detection, fault isolation, and fault identification exploiting the use of analytical rather than hardware redundancy.

One of the objectives of the present application is to combine the quantitative simulation of a continuous process with a qualitative simulation technique to perform fault detection; fault isolation and fault identification are beyond the scope of this experiment. A new methodological aspect in the mark of the *Fuzzy Inductive Reasoning* methodology is applied to improve fault detection in large-scale systems. An example already employed in previous publications [Vesanterä, 1988; de Albornoz and Cellier, 1994; de Albornoz, 1996], namely the numerical model of a B747 aircraft, is used to compare the obtained results.

In this appendix, a brief review of the research work already done in previous publications as well as the used aircraft model is done, so as to be able to compare the previously obtained results with the ones obtained with the newly proposed method.

II.2.1 Previous research and th e aircraft model

In a previous research effort at the University of Arizona, qualitative simulation was applied to reason inductively about the behaviour of a quantitatively simulated B-747 aircraft model, to determine on-line when a malfunction occurs in the quantitative model. A **crisp inductive reasoner** (using a qualitative model computed only on the basis of class values, i.e., without membership and/or side information) was used to recognise that the aircraft had qualitatively changed its behaviour within a few seconds after a simulated malfunction had taken place. Crisp inductive reasoning and **crisp detection** were used to perform fault detection. The results of this study were reported in [Vesanterä, 1988; Vesanterä and Cellier, 1989].

Later on, continuing with the research in this area at the Polytechnic University of Catalonia, the former crisp inductive reasoner was replaced by a fuzzy inductive reasoner. This modified reasoning scheme had an enhanced discriminatory power and an improved forecasting capability. The new fuzzy inductive reasoner allows the prediction of a real-valued variable, whereas the crisp inductive reasoner was able to predict class values only. In this research effort, **fuzzy inductive reasoning** (*FIR*) and **crisp detection** (only using information given by the class values) were used to perform fault detection. The results of this study were reported in [de Albornoz, 1996; de Albornoz and Cellier, 1993a;1994].

An improved method based on fuzzy inductive reasoning is used here with the aim of fault detection. **Fuzzy inductive reasoning** and what is now called **envelope detection**, previously explained in Section 2.5.1 within Chapter 2 of the dissertation are used. As will be shown, using envelopes improves the fault detection approach, allowing the detection of faults at an earlier stage, and even the detection of faults that are not detected using either of the previous two approaches.

A numerical model of the B-747 aircraft has been used to generate episodes of the five variables shown in Figure II-6. Two input variables, the variation in the elevator deflexion $\Delta\delta_{\mathrm{trim}}$ and the variation in the thrust ΔT_{trim}, and three output variables, the lift *L*, drag *D,* and the flight path angle *GA*, are considered. The mathematical model used is exactly the same that was reported in [Vesanterä, 1988; Vesanterä and Cellier, 1989]. This model, named B4, is valid for a B-747 at high altitude horizontal flight.

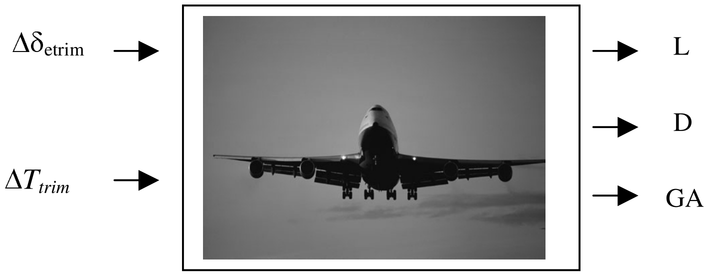


Figure II-6 Input and output variables of the aircraft model

The mathematical model described in the given references reflects an essentially longitudinal flight restricted to longitudinal deviations from a trimmed reference flight condition. The main characteristic of this reference flight is the requirement that the resultant force and moment acting on the aircraft's centre of mass are zero.

The original aerodynamic parameters of this model were modified to artificially generate faults to test the fault detection methodology (the generated faults do not necessary correspond to real fault situations). Hence different models are found that represent structural changes of the original plane. The main characteristics of these models are:

- Model B4 is the original model representing a B-747 in cruise flight at 20’000 feet altitude. Its aerodynamic parameters are used as reference for the other models.

- Model B13 is characterised by a much more damped step response to the same step inputs. The effect of the angle of attack on the lift coefficient *L* has been slightly increased. A significant change in the effect of the angle of attack on the drag coefficient *D* occurred. The effect of the elevator deflection, $\Delta\delta_{trim}$, on the pitching moment has also changed.

- Model B5 represents a change of the original B4 model, which completely alters the influence that the angle of attack has on the aerodynamic response of the aircraft.

- Model 747 represents an enlarged B-747 in cruise flight. In this case, the values of the lift *L*, the aerodynamic momentum, the drag *D,* and the pitch angle are changed.

Only two among the aforementioned aircraft models are being used in the concrete application, namely the B4 model, corresponding to the aircraft under normal operation, and the B13 model, representing a fault. Other malfunction situations described in the referenced literature can be tackled in the same way.

The simulation of the quantitative model is performed using *ACSL* (*Advanced Continuous Simulation Language*, [Mitchell & Gauthier, 1991]). Once the quantitative model has been implemented in *ACSL*, trajectories for the five named variables are generated. FIR is then applied, and a qualitative model of the aircraft is obtained. Details about the process of obtaining the qualitative aircraft model can be found in [Mirats, 1998]. As FIR works with MISO systems only, it is necessary to find one model for every output. After performing the optimal (exhaustive) mask search, the following masks are found:

$$B4_L = \begin{array}{c} \begin{matrix} \delta_e & \delta_T & L & D & \gamma \end{matrix} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & -2 \\ -3 & -4 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array} \quad B4_D = \begin{array}{c} \begin{matrix} \delta_e & \delta_T & L & D & \gamma \end{matrix} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ -2 & -3 & 0 & -4 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \quad B4_\gamma = \begin{array}{c} \begin{matrix} \delta_e & \delta_T & L & D & \gamma \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & -1 & -2 \\ -3 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

These three masks, together with the behaviour matrices and the fuzzification landmarks, constitute the qualitative model of the B4 aircraft. This model was obtained using SAPS in its present Matlab toolbox version.

Now, with the qualitative model, it is possible to compare trajectories from variables of the real system (*ACSL* simulation) with forecast episodes from the qualitative model. When a structural change has taken place, the model can no longer predict the system. It is at this point that fault detection is achieved. Results obtained with the new methodology are to be compared with those obtained in the previous publications.

II.2.2 Smooth changes in the aircraft parameters.

In order to generate trajectories with accidents, the flight starts out with the B4 model (normal situation), and at a given time, a malfunction is numerically simulated by changing some of the structural parameters of the aircraft. In the previous research studies [Vesanterä and Cellier, 1989; de Albornoz and Cellier, 1994], the change of parameters occurs abruptly, i.e., a step perturbation is performed so that the change on the parameters of the aircraft is immediate. Such a sudden change is relatively easy to detect, as it results in a violent transient behaviour of the aircraft, whereby the qualitative classes of the considered outputs change, making it possible to detect the fault by looking at the class values only. Precisely this technique, looking for discrepancies between the observed and the predicted class values of the output variable, has been employed to perform fault detection in the aircraft model in [Vesanterä and Cellier, 1989] and [de Albornoz and Cellier, 1994]. Although the former technique used a crisp inductive reasoner whereas the latter used an improved fuzzy inductive reasoner, the fault was, in both cases, detected using information about the class values only.

The performed instantaneous parameter change implies a sharp transient in the variables taken into account. This transient was used to detect the aeroplane malfunction, so in some way, the methodology did not really detect the model change, but rather the highly dramatic transition phase between the two models. For instance, Figure II-7 shows the transient in the output variable *D* resulting from an abrupt B4-B13 model transition.

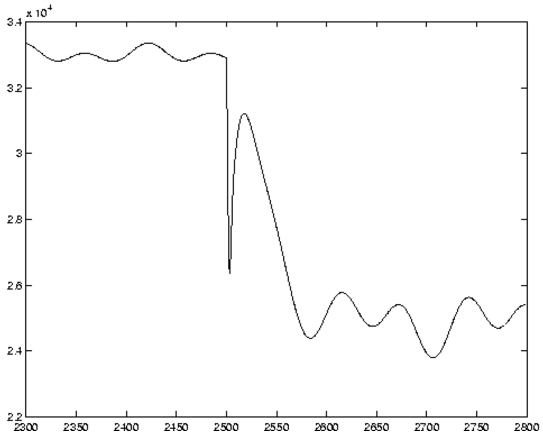


Figure II-7 Drag (D): trajectory with a sudden change in the parameters

Although some real malfunctions may indeed involve a sudden change in the parameters, quite often this is not the case, and it may therefore be more realistic to consider gradual rather than sudden changes. To model this situation, a smooth change in the aeroplane parameters has been simulated. The total range of parameter changes is the same as used previously, but now, the parameters change gradually by ramping them from their initial to their final values over a period of 10 seconds. Figure II-8 shows the change in the output variable *D* when the parameter values change smoothly.

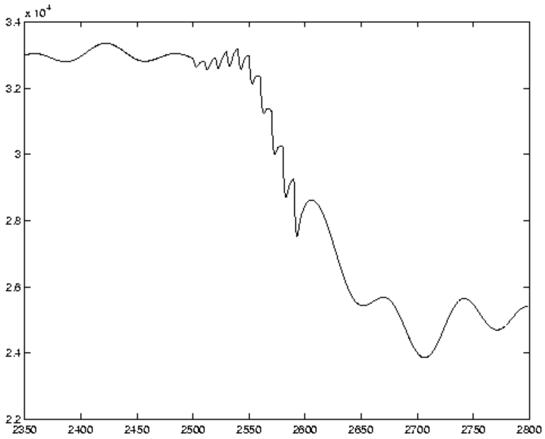


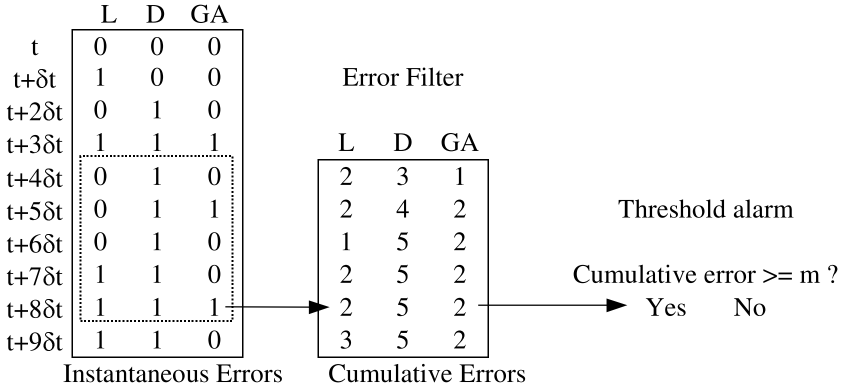
Figure II-8 Drag (D): trajectory with a smooth change in the parameters.

II.2.3 Crisp detection using smoothed data.

In order to compare the effectiveness of the new fault detection scheme, based on the so-called envelopes, with previous results, the fault detection method presented in [Vesanterä and Cellier, 1989; de Albornoz and Cellier, 1994] is now summarised and applied to the smooth parameter change explained in the previous section.

By using the numerical ACSL aircraft model, quantitative data representing the real system in a fault situation are gathered. The considered fault is the one named B13 in section II.2.1. The real-valued data obtained by means of a quantitative (ACSL) fault simulation is then converted to qualitative triplets of class, membership, and side values using the fuzzification module of FIR. Afterwards, the qualitative model previously obtained from the normal aircraft operation data (reported in section II.2.3) is used to predict the future behaviour of the aeroplane using the new situation fault data. Therefore, for each new data point coming from the real system (in the given experiment, the ACSL simulation), a prediction of the considered output variables is computed. The idea behind is that when a structural change occurs, the qualitative model will receive inputs that have never been seen before. Hence it will no longer be able to correctly predict the behaviour of the system, thereby triggering an alarm vector indicating that a system fault has occurred.

At each sampling interval, an instantaneous error for each system output is computed by subtracting the real-system (fuzzified) class values and the predicted class values using the qualitative model. As long as the prediction is correct, the subtraction results in a value of zero. Hence values different from zero indicate a false prediction that may be interpreted as a potential indicator of a fault having occurred. These errors are stored in a matrix. Then, a moving average error filter is shifted down this matrix, computing, for each output, the sum of instantaneous errors it covers. These cumulative errors are in turn stored in another matrix. If any of these values, at a given point of time, passes the threshold (*m*) of the alarm module, an alarm is immediately triggered, i.e., a fault has been detected. Figure II-9 may illustrate the process:



	L	D	GA
t	0	0	0
t+δt	1	0	0
t+2δt	0	1	0
t+3δt	1	1	1
t+4δt	0	1	0
t+5δt	0	1	1
t+6δt	0	1	0
t+7δt	1	1	0
t+8δt	1	1	1
t+9δt	1	1	0

Instantaneous Errors

L	D	GA
2	3	1
2	4	2
1	5	2
2	5	2
2	5	2
3	5	2

Cumulative Errors

Figure II-9 Fault detection scheme

Table II-XIII summarises the results obtained when applying the described fault detection method to the new data set, i.e., a gradual change of parameters. The resulting alarm vectors using two different threshold values, *m = 5* and *m = 2,* in conjunction with a filter depth of 5, are shown.

The change in the parameters starts at time instant *t = 2500*, and lasts 10 seconds. The alarm vectors in the table below show that within the next 18 seconds the malfunction is not detected when using a threshold in the alarm module of *m = 5*, and some abnormal situation is reported for time span (2505, 2515) when using *m = 2*. The latter would have been interpreted by the fault monitoring system as a false alarm, because from t = 2516 onward, the alarm vector is switched off again. Moreover, using low threshold values implies a higher probability of false alarms. It is interesting to detect faults as early as possible, but only real faults should be reported.

As presented in next section, this problem can be tackled using the so-called 'envelopes'.

Time	Alarm vector, m=5	Alarm vector, m=2
2500	0	0
2501	0	0
2502	0	0
2503	0	0
2504	0	0
2505	0	1
2506	0	1
2507	0	1
2508	0	1
2509	0	1
2510	0	1
2511	0	1
2512	0	1
2513	0	1
2514	0	1
2515	0	1
2516	0	0
2517	0	0
2518	0	0

Table II-XIII Alarm vectors using the detection approach proposed in [6,9] with smooth change

II.2.4 Detection with envelopes

The concept of 'envelopes', in the context of the FIR methodology, has been properly detailed in Section 2.5.1 of Chapter 2. In this appendix, the given method will be used to detect structural changes in an aircraft model.

When using a FIR qualitative model to predict a system, a measure of the model error can be obtained comparing the forecast value with the real (fuzzified) value of the concerned variable. Using the B4 aircraft qualitative model presented in Section II.2.1, new values of the outputs can be forecast and compared with the B4 validation data set (values of the variable trajectories not used in the FIR modelling process). If the mean square error is used, *mse = 0.1702* is the obtained value for the *forecasting average error* of the FIR qualitative model. Although there is a separate forecasting average error for every output variable, in this study, the largest of these *mse* values has been used to characterise the envelopes of all three of the output variables.

The idea behind the envelopes approach is to compute, for each forecast value, an *interval of acceptability* of the real trajectory value. Up to now, a single (defuzzified) prediction was made at each point in time, which had been computed as an average of the output values of the five nearest neighbours in the training data base. Yet, it is perfectly defendable to make predictions in different ways. For example, it may make sense to consider the range of predictions made by the five nearest neighbours as an envelope of acceptable predictions. The closer the five nearest neighbours are to each other, i.e., the smaller the dispersion among them, the more narrow that envelope will be. On the other hand, the larger the dispersion among the five nearest neighbours, the wider the envelope will become. Let α and β be the minimum and maximum predictions made by any of the five neighbours, the envelope will then be defined by the range $[\alpha,\beta]$, a time-varying *interval of forecasting acceptability* associated with the predicted output variable. This information can be exploited for fault monitoring.

Two experiments have been carried out using this new approach. First, the envelopes have been used to monitor a sudden change in the aircraft parameters, as proposed in the previous publications [Vesanterä and Cellier, 1989; de Albornoz and Cellier, 1994], and subsequently, they were employed to monitor a smooth change in the aircraft parameters as explained in Section II.2.2.

II.2.4.1 Sudden change detection

The method of the acceptability envelope of the variables has been applied to the case of an instantaneous change in the aeroplane parameters. Figure II-10 shows the trajectory of the real output variable *D* together with the interval of acceptable forecast values (the so-called envelope) obtained by the qualitative FIR model of the B4 aircraft. The figure covers a much shorter time interval than that provided in Figure II-7 and Figure II-8, because the envelope is narrow, indicating that the found qualitative model is of high quality. A wider time window would have made the figure less easily interpretable.

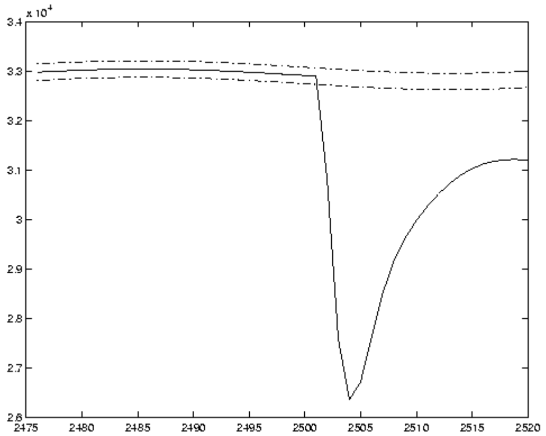


Figure II-10 Variable D: real system values and forecast acceptability envelope

In order to perform fault detection using the envelopes, the method explained in Chapter 2 is used. An instantaneous error matrix is constructed, where every column is associated with an output variable, and every row corresponds to a sampling instant. A zero value denotes no error (i.e. the value of the real system variable lies within the interval of variable acceptability), and a value of one means an instantaneous prediction error has been registered (i.e. the quantitative real value is outside the range of variable acceptability). Then the error matrix is filtered using a moving average filter, and when the output passes the specified error threshold, an alarm is triggered. Table II-XIV summarises the results obtained with this method when applied to the situation of a sudden change in aircraft parameter values.

The fault alarm vector has been obtained using a threshold of *m = 5* and an error window of depth 5. Notice that the accident is detected at time instant 2503, only three seconds after it took place, and the fault remains flagged ever after. The new method of the envelopes is hence reported to detect the malfunction at an earlier time than the approach proposed in [Vesanterä and Cellier, 1989; de Albornoz and Cellier, 1994]. Moreover, the obtained alarm vector is more stable in the sense that it does not return to a zero value after the transient has taken place, thereby decreasing the probability of a true emergency being mistaken for a false alarm.

time	alarm
2500	0
2501	0
2502	0
2503	1
2504	1
2505	1
2506	1
2507	1
2508	1
2509	1
2510	1
2511	1
2512	1
2513	1
2514	1
2515	1
2516	1
2517	1
2518	1

Table II-XIV Alarm vector obtained when performing fault detection with the envelopes approach in a sudden parameter change situation.

II.2.4.2 Smooth change detection

In this simulation, a smooth parameter change is applied as explained previously, but now, the envelopes approach is applied to fault detection. Figure II-11, Figure II-12 and Figure II-13 show the trajectories of the three output variables together with the forecast envelopes obtained for the qualitative B4 model.

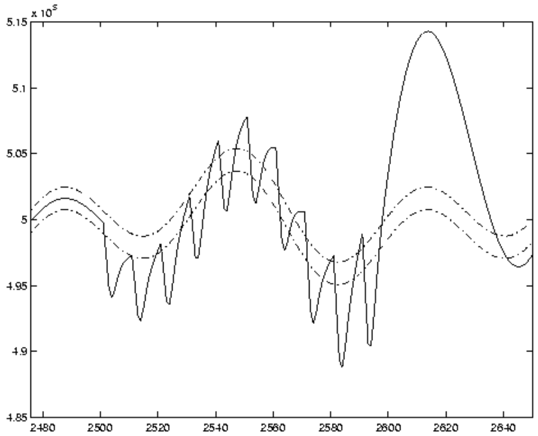


Figure II-11 Envelopes with smooth change in variable L.

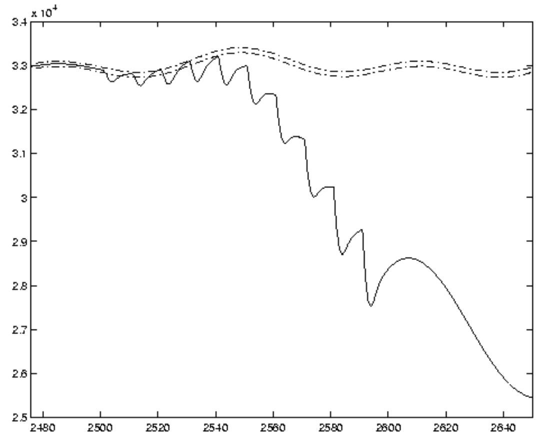


Figure II-12 Envelopes with smooth change in variable D.

Table II-XV summarises the results that have been obtained using the method of the envelopes when using two different alarm thresholds: $m = 5$ and $m = 2$ and an error window of depth 5.

In order to make it easy to compare the results with those obtained using fuzzy inductive reasoning together with crisp fault detection, as presented in [de Albornoz and Cellier, 1994], the two left-most columns of Table II-XV show the results obtained with the method of envelopes, whereas the two right-most columns reproduce the results discussed in Section II.2.3.

Using *m = 5*, the smooth fault is detected four seconds after initiating the change of the B4 parameters. If *m = 2* is applied, an earlier detection is achieved (three seconds instead of four), and conversely to the results obtained in Section II.2.3, the fault alarm remains flagged after time instant 2515.

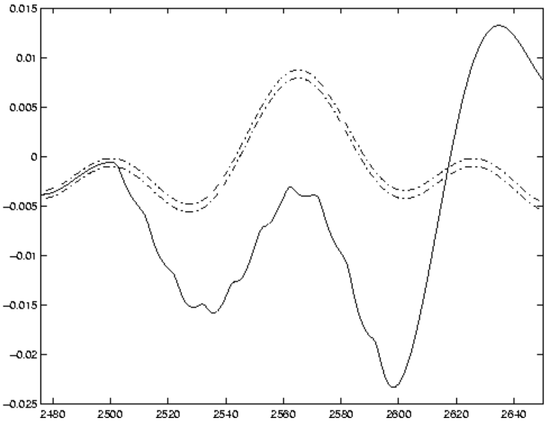


Figure II-13 Envelopes with smooth change in variable GA.

	Envelope method		Crisp detection method	
Time	m = 5	m = 2	m = 5	m = 2
2500	0	0	0	0
2501	0	0	0	0
2502	0	0	0	0
2503	0	1	0	0
2504	1	1	0	0
2505	1	1	0	1
2506	1	1	0	1
2507	1	1	0	1
2508	1	1	0	1
2509	1	1	0	1
2510	1	1	0	1
2511	1	1	0	1
2512	1	1	0	1
2513	1	1	0	1
2514	1	1	0	1
2515	1	1	0	1
2516	1	1	0	0
2517	1	1	0	0
2518	1	1	0	0

Table II-XV Comparing envelopes and crisp detection results for a smooth change situation

II.2.5 Discussion

When a malfunction occurs in a system, it is desirable to detect the malfunction as early as possible, but also in a reliable and robust manner, minimising the number of false alarms that are being interpreted as true emergencies, but also minimising the number of true emergencies that are being interpreted as false alarms. The application of the envelope method developed in chapter 2 concentrates on both aspects of fault monitoring, using a Boeing 747 aircraft model as a benchmark.

Two *Fuzzy Inductive Reasoning* (*FIR*) based approaches are compared. Both of them use FIR as qualitative modelling technique, but the former uses a crisp fault detection approach, whereas the latter makes use of a newly proposed envelope detection method. Crisp fault detection had successfully been used in previous research efforts [Vesanterä and Cellier, 1989; de Albornoz and Cellier, 1994] to detect sudden changes in system parameters, such as an engine falling off the aircraft, but is not well suited when confronted with smooth parameter changes that lead to a slow deterioration of the plant, such as ice building up on the wings of the aircraft.

The concepts of interval of forecast acceptability and interval of variable acceptability introduced in Chapter 2, and are used to propose a fault monitoring method based on these intervals. In the case of a sudden change of the aircraft parameters, the new proposed method detects the fault earlier. When the parameters of the aircraft are varied smoothly, the malfunction is detected using error threshold values that do not permit a fault detection using the crisp detection method. Moreover, the detection is more robust, since for a given threshold, the existing fault remains flagged, whereas using the crisp method, the alarm disappears again, making it likely that the true emergency would be interpreted by the flight engineer as a false alarm.

8. References and bibliography

[Adams, M.J., J.R. Allen, 1998], “Variable selection and multivariate calibration models for X-ray fluorescence spectrometry”, Journal of Analytical Atomic Spectrometry, vol. 13, n. 2, pp. 119-124, ISSN: 0267-9477.

[Ali M. and D. A. Scharnhorst, 1985], “Sensor-based fault diagnosis in a flight expert system,” Proc. IEEE Conference on Artificial Intelligence Applications, Miami, FL, USA, pp. 49-54.

[Al-Kandari, N.M., I.T. Jolliffe, 1997], “Variable selection and interpretation in canonical correlation analysis”, Communications in Statistics. Part B: Simulation and Computation, vol. 26, n. 3, pp. 873-900, ISSN: 0361-0918.

[Allen, D.M., 1971], “Mean square error of prediction as a criterion for selecting variables”, Technometrics, Vol. 13, pp. 469-475.

[Allen, D.M., 1974], “The relationship between variable selection and data augmentation and a method for prediction”, Technometrics, Vol. 16, No. 1, February 1974, pp. 125-127.

[Aoki, K., M. Yoshida, 1999], “The correlation between Si III lambda 1892/C III lambda 1909 and Fe II lambda 4500/H beta in low redshift QSOs”, Astronomical Society of the Pacific Conference Series, Vol. 162, pp. 385-394.

[Ashby, W.R., 1964], "Constraint analysis of many-dimensional relations", General Systems Yearbook, 9, 1964, pp. 99-105.

[Ashby, W.R., 1965], "Measuring the internal information exchange in a system", Cybernetica, 8, No. 1, 1965, pp. 5-22.

[Bathie, W. W., 1996], “Fundamentals of gas turbines”, John Wiley & Sons, Inc.

[Beale, E.M.L., M.G. Kendall, D.W., Mann, 1967], “The discarding of varibles in multivariate analysis”, Biometrika, 54, pp. 357-366.

[Bhattacharyya, G.K., R.A. Johnson, 1977], “Statistical concepts and methods”, New York, John Wiley.

[Brieman, L., J.H. Friedman, 1985], “Estimating optimal transformations for multiple regression and correlation”, Journal of the american statistical association, 77, pp. 580-619.

[Broekstra, G., 1976-77], "Constraint analysis and structure identification", Annals of Systems Research; **I**: 5, 1976, pp. 67-80; **II**: 6, 1977, pp. 1-20.

[Broekstra, G., 1978], "On the representation and identification of structure systems", International Journal of Systems Science, 9, No. 11, pp. 1271-1293.

[Broekstra, G., 1981], "C-Analysis of C-Structures: Representation and evaluation on reconstruction hypothesis by information measures", International Journal of General Systems, 7, No. 1, pp. 33-61.

[Carvajal, R., A. Nebot, 1997], “Growth model for white shrimp in semi-intensive farming using inductive reasoning methodology”, Computers and electronics in agriculture, 19, pp. 187-210.

[Cavallo, R.E., G.J. Klir, 1978], "A conceptual foundation for systems problem solving", International Journal of Systems Science, 9, No.2, pp. 219-236.

[Cavallo, R.E., G.J. Klir, 1979a], "Reconstructability analysis of multi-dimensional relations: A theoretical basis for computer-aided determination of acceptable systems models", International Journal of General Systems, Vol. 5, No. 3, pp. 143-171.

[Cavallo, R.E., G.J. Klir, 1979b], "The structure of reconstructable relations: A comprehensive study", Journal of Cybernetics, 9, No. 4, pp. 399-413.

[Cavallo, R.E., G.J. Klir, 1981a], "Reconstructability analysis: Overview and Bibliography", International Journal of General Systems, Vol. 7, No 1, pp. 1-6.

[Cavallo, R.E., G.J. Klir, 1981b], "Reconstructability analysis:Evaluation of Reconstruction Hypotheses", International Journal of General Systems, Vol. 7,No. 1, pp. 7-32.

[Cavallo, R.E., G.J. Klir, 1982], “Reconstruction of possibilistic behaviour systems”,Fuzzy sets and systems, 8, pp. 175-197.

[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.

[Cellier, F.E., 1991a], "Continuous System Modeling", Springer-Verlag, New York, USA.

[Cellier, F.E., 1991b], "General system problem solving paradigm for qualitative modelling", Qualitative simulation modelling and analysis, pp. 51-71, Springer Verlag, New York.

[Cellier, F.E., A. Nebot, F. Mugica, A. de Albornoz, 1992], "Combined qualitative/quantitative simulation models of continuous-time processes using Fuzzy Inductive Reasoning techniques", Proc. SICICA ‘92, IFAC Symposium on Intelligent Components and Instruments for Control Applications’. Malaga, Spain, May 22-24, pp.589-593.

[Cellier, F.E., F. Mugica, 1992], "Systematic design of fuzzy controllers using inductive reasoning", Proceedings ISIC-92, IEEE International Symposium on Intelligent Control, pp. 198-203, Glasgow, Scotland, U.K.

[Cellier, F.E., J. López, A. Nebot, G. Cembrano, 1996], "Means for estimating the forecasting error in fuzzy inductive reasoning", 2nd international conference on qualitative information, fuzzy techniques, and neural networks in simulation, pages 654-660, Budapest, Hungary.

[Cellier, F.E., and A. de Albornoz, 1998], "The problem of distortions in Reconstruction Analysis", *Systems Analysis, Modelling, Simulation*, **33**(1), pp.1-19.

[Chen J. and R. J. Patton, 1998], Robust Model-based Fault Diagnosis for Dynamic Systems, Kluver Academic Publishers.

[Chipman, H., M. Hamada, C.F.J. Wu, 1997], “Bayesian variable-selection approach for analyzing designed experiments with complex aliasing”, Technometrics vol. 39, n. 4, Nov 1997, pp. 372-381.

[Conant, R.C., 1972], "Detecting subsystems of a complex system", IEEE Trans. on Systems, Man, and Cybernetics, SMC-2, No. 4, pp. 550-553.

[Conant, R.C., 1976], "Laws of information which govern systems", IEEE Trans. on Systems, Man, and Cybernetics, SMC-6, No. 4, pp. 240-255.

[Conant, R.C., 1980], "Structural modelling using a simple information measure", International Journal of Systems Science, 11, No. 6, June, pp. 721-730.

[Conant, R.C., 1981], "Detection and analysis of dependency structures", International Journal of General Systems, 7, No. 1, pp. 81-91.

[Cueva, J., R. Alquézar, A. Nebot, 1997], "Experimental comparison of fuzzy and neural network techniques in learning models of the central nervous system control", EUFIT'97, September 8-11, 1997, Germany, pp.1014-1018.

[D'Ambra, L., N. C. Lauro, 1992], "Non symmetrical exploratory data analysis", Statistica applicata (Italian journal of applied statistics), Vol. 4, n. 4, pp. 511-529.

[Daling, J.R and H. Tamura, 1970], "Use of orthogonal factors for selection of variables in a regression equation - An illustration," Applied Statistics, 19(3), pp. 260-268.

[de Albornoz A., and F. E. Cellier, 1994], "Building intelligence into an autopilot using qualitative simulation to support global decision making", Simulation, **62**(6), pp. 354-364.

[de Albornoz, A., 1996], "Inductive Reasoning and Reconstruction Analysis: Two complementary tools for qualitative fault monitoring of large-scale systems", Ph.D. thesis, Dept. Llenguatges i sistemes informàtics, Universitat Politècnica de Catalunya.

[de Albornoz, A., F.E. Cellier, 1993a], "Qualitative simulation applied to reason inductively about the behaviour of a quantitatively simulated aircraft model", Quardet '93, IMACS Intl., Workshop on Qualitative Reasoning and Decision Technologies, Barcelona, Spain, June 16-18, pp. 711-721.

[de Albornoz, A., F.E. Cellier, 1993b], "Variable selection and sensor fusion in automated hierarchical fault monitoring of large scale systems", Quardet '93, IMACS Intl., Workshop on Qualitative Reasoning and Decision Technologies, Barcelona, Spain, June 16-18, pp. 722-734.

[de Boor, C., 1978], "A practical guide to splines", Springer Verlag, New York.

[De Volson, W., 1995], "Turbines: theoretical and practical", John Wiley & Sons, Inc. (New York), Chapman & Hall Ltd. (London), 2nd edition.

[Dunia R., S. J. Qin, T. F. Edgar, and T. J. McAvoy, 1996], "Identification of faulty sensors using principal component analysis," AIChE Journal, 42, pp. 2797-2812.

[Dunia R., 1997], "A Unified Geometric Approach for Process Monitoring and Control", Ph.D. dissertation, Department of Chemical Engineering, The University of Texas at Austin, USA.

[Dunia R., and S. J. Qin, 1997a], "Multi-dimensional fault diagnosis using a subspace approach", American Control Conference'97, Albuquerque, June 4-6, New Mexico.

[Dunia, R., and S.J. Qin, 1997b], "Multidimensional fault detectability, identifiabiltiy, and reconstructability", Presented at AIChE Annual Meeting, Nov. 16-21, Los Angeles, CA, paper 190f.

[Dunia R., and S.J. Qin, 1998a], "Joint diagnosis of process and sensor faults using principal component analysis", Control Engineering Practice, vol. 6, n. 4, 457-469.

[Dunia R., and S.J. Qin, 1998b], "Subspace approach to multidimensional fault identification and reconstruction", AIChE Journal, 44(8), pp. 1813-1831.

[Dunia, R., and S. J. Qin, 1998c], “A unified geometric approach to process and sensor fault identification and reconstruction: the unidimensional fault case,” Computers Chem. Engng., 22 (7-8), pp. 927-943.

[Escobet, A., A. Nebot, F.E. Cellier, 1999], "Model Acceptability Measure for the Identification of Failures in Qualitative Fault Monitoring Systems", Proc. ESM'99, European Simulation MultiConference, Warsaw, Poland, pp.339-347.

[Escobet, T., J. Quevedo, S. Tornil, L. Travé-Massuyès, 1999], “Integration of dynamic models in gas turbine supervision control”, 7th IEEE International conference on emerging technologies and factory automation (ETFA’99), Universitat Politècnica de Catalunya, Campus Nord, Barcelona, 18-21 October, pp. 995-1001.

[Ettaleb, L., G.A. Dumont, M.S. Davies, 1998], "An extended off-line least-squares method for parameter identification and time delay estimation", Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171). Tampa, FL, USA, December 1998, vol. 3, pp. 3423-3428.

[Fertner, A., A. Sjölund, 1986], "Comparison of various time delay estimation methods by computer simulation", Transactions on acoustics, speech and signal processing, Vol. ASSP-34, n.5, pp. 1329-1330

[Friedman, J.H., 1991], “Multivariate adaptive regression splines”, The annals of statistics, Vol. 19, No. 1, pp. 1-141.

[Gaines, B.R., 1977], "System identification, approximation and complexity", International Journal of General Systems, 3, No. 3, pp. 145-174.

[Geladi, P. and B.R. Kowalski, 1986], “Partial least squares regression: A tutorial”, Analytica Chimica Acta, 185, pp. 1-17.

[Glen, W.G., W.J. Dunn III, D.R. Scott, 1989a], “Principal components analysis and Partial least squares regression”, Tetrahedron Computer Methodology, Vol. 2, n. 6, pp. 349-376.

[Glen, W.G., M. Saker, W.J. Dunn III, D.R. Scott, 1989b], “Unipals: Software for principal components analysis and Partial least squares regression”, Tetrahedron Computer Methodology, Vol. 2, n. 6, pp. 377-396.

[Gloss, R., 1998], “New methods of signal pre-shaping strongly increase bandwidth of closed loop PZT actuators”, ACTUATOR 98, Proceedings of the 6th International Conference on New Actuators with Accompanying Exhibition, Bremen, Germany, pp. 285-287.

[Goguen, J.A., F.J. Varela, 1979], "Systems and distinctions; Duality and complementarity", International Journal of General Systems, Vol. 5, pp. 31-43.

[Händel, P., 1999], "Frequency selective adaptive time delay estimation", IEEE Transactions on Signal Processing, vol. 47, no. 2, pp. 532 - 535, ISSN: 1053-587X.

[Hoeting, J., A.E. Raftery, D. Madigan, 1996], “Method for simultaneous variable selection and outlier identification in linear regression”, Computational Statistics & Data Analysis, vol. 22, n. 3, pp. 251-270.

[Hoeting, J., J.G. Ibrahim, 1998], “Bayesian predictive simultaneous variable and transformation selection in the linear model”, Computational Statistics & Data Analysis, vol. 28, n. 1, Jul 1998, pp. 87-103.
[Israels Abby, 1992], “Redundancy analysis for various types of variables”, Statistica applicata (Italian journal of applied statistics), Vol. 4, n. 4, pp. 531-542.

[Jackson, J.E., 1991], “A user's guide to principal components”, a Wiley-Interscience Publication.

[Jacovitti, G., G. Scarano, 1993], "Discrete time techniques for time delay estimation", IEEE Transactions on Signal Processing, vol.41, no.2, Feb. 1993; pp.525-533, ISSN: 1053-587X.

[Jazwinski, A.H., 1970], "Stochastic processes and filtering theory", Academic Press, Inc.

[Jeffers, J.N.R., 1967], "Two case studies in the application of principal component analysis", Applied statistics (Journal of the Royal statistical society, serie C), Vol. 16,, No. 3, pp. 225-236.

[Jerez, A., A. Nebot, 1997], “Genetic algorithms vs. Classical search techniques for identification of fuzzy models”, Proc. EUFIT’97, 5th European congress on intelligent techniques and soft computing, Aachen, Germany, 8-12 September, pp. 769-773.

[Johnson, R.A., D. W. Wichern, 1992], “Applied Multivariate Statistical Analysis”, Prentice Hall, Third Edition.

[Jollife, I. T., 1972], “Discarding variables in a principal component analysis. I: Artificial Data”, Applied Statistics, 21, pp. 160-173.

[Jollife, I. T., 1973], “Discarding variables in a principal component analysis. II: Real Data”, Applied Statistics, 22, pp. 21-31.

[Kabaila, P., 1997], “Admissible variable-selection procedures when fitting misspecified regression models by least squares”, Communications in Statistics Theory and Methods, vol. 26, n. 10, Oct 1997, pp. 2303-2306.

[Kalouptsidis, N., 1997], "Signal processing systems, Theory and design", Wiley & sons, Inc.

[Keller, J. P., D. Bonvin, 1992], “Selection of input and output variables as a model reduction problem”, Automatica, vol. 28, n. 1, pp. 171-177, ISSN: 0005-1098.

[Kempthorne, P.J., 1984], ”Admissible variable-selection procedures when fitting regression models by least squares for prediction”, Biometrika, 1984, 71, 3, pp. 593-597.

[Kengerlinskiy, G.A., 1978], "An informational approach to the decomposition of complex systems", Engineering Cybernetics, 16, No. 1, pp. 91-97.

[Kim, K.S., J.W. Park, S.H., Nam, J.J. Im, E.S. Choi, B.H. Jun, 1998], “A study for the effect of electrical stimulation on tinnitus treatment based on the correlation analysis of ABR and EcochG”, Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vol. 20, Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286), Piscataway, NJ, USA, pp. 2456-2459.

[Klir, G.J., 1969], “An approach to general systems theory”, Van Nostrand Reinhold, New York, USA.

[Klir, G.J., 1981], "On systems methodology and inductive reasoning: the issue of parts and wholes", General systems yearbook, Vol. 26, pp. 29-38.

[Klir, G.J., 1985], "Architecture of System Problem Solving", Plenum Press, N.Y., U.S.A.

[Klir, G.J., 1989], "Inductive systems modelling: An overview", Modelling and simulation methodology: knowledge systems' paradigms, pp. 55-75, North-Holland, Amsterdam, The Nederlands. Elsevier science publishers.

[Klir, G.J., 1991], “Aspects of uncertainty in qualitative systems modelling”, in Qualitative simulation modelling and analysis, (Fishwick & Luker, Eds.), Springer-Verlak, USA.

[Klir, G.J., B. Yuan, 1995], “Fuzzy sets and fuzzy logic. Theory and applications”, Prentice Hall, Englewood Cliffs, New Jersey, USA.

[Klir, G.J., H.J.J. Uyttenhove, 1979], "Procedures for generating reconstruction hypotheses in the Reconstructability Analysis", International Journal of General Systems, Vol. 5, pp. 231-246.

[Klir, G.J., T.A. Folger, 1988], "Fuzzy sets, uncertainty and Information". Prentice-Hall, Englewood Cliffs.

[Krippendorff, K., 1979], "On the identification of structures in multivariate data by the spectral analysis of relations", Proc. 23rd Annual SGSR Meeting, Houston, January 3-6, 1979, pp. 82-91. (The Annenberg school of communications, University of Pennsylvania)

[Krippendorff, K., 1981], "An algorithm for identifying structural models of multi-variate data", International Journal of General Systems, 7, No. 1, pp. 63-79.

[Krzanowski, W.J., 1987], “Selection of variables to preserve multivariate data structure, using principal components”, Applied Statistics, 36, pp. 22-33.

[Law, A. and D. Kelton, 1990], "Simulation Modeling and Analysis", Englewood Cliffs, NJ: Prentice Hall.

[Li, D., and F.E. Cellier, 1990], "Fuzzy measures in Inductive Reasoning", Proceedings 1990 Winter Simulation Conference, New Orleans, La., pp. 527-538.

[Lindgren F., P. Geladi, A. Berglund, M. Sjostrom, S. Wold, 1995], “Interactive variable selection (IVS) for PLS .Part II: Chemical applications”, Journal of chemometrics, 9(5), pp. 331-342.

[Lisboa, P., A.R. Mehri-Dehnavi, 1996], “Sensitivity methods for variable selection using the MLP”, Proceedings of International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing, NICROSP 1996. IEEE, Los Alamitos, CA, USA. p 330-338.

[López, J., 1999], "Time Series Prediction Using Inductive Reasoning Techniques", Ph.D. dissertation, Organització i Control de Sistemes Industrials, Universitat Politècnica de Catalunya, Barcelona, Spain.

[López, J., F.E. Cellier, 1999], "Improving the Forecasting Capability of Fuzzy Inductive Reasoning by Means of Dynamic Mask Allocation", Proc. ESM'99, European Simulation MultiConference, Warsaw, Poland, pp.355-362.

[López, J., G. Cembrano, F.E. Cellier, 1996], "Time series prediction using Fuzzy Inductive Reasoning", ESM'96: European Simulation Multiconference, Budapest, Hungary, June 2-6, pp. 765-770.

[Madden, R.F., W.R. Ashby, 1972], "The identification of many-dimensional relations", International Journal of Systems Science, 3, No. 4, pp. 343-356.

[Mansfield, E.R., J.T. Webster, R.F. Gunst, 1977], "An analytic variable selection technique for principal component regression", Applied statistics, Vol. 26, n. 1, pp. 34 - 40.

[Marple, S.L., 1987], "Digital spectral analysis with applications", Prentice-Hall.

[Marple, S.L.Jr., 1999], "Estimating group delay and phase delay via discrete-time "analytic" cross-correlation", IEEE Transactions on Signal Processing, vol.47, no.9; Sept.; p.2604-2607,ISSN: 1053-587X

[MathWorks, 1997], "Getting Started with MATLAB; Using MATLAB (Version 5.1)", Natick, MA: The MathWorks Inc.

[McShane, M.J., G.L. Cote, C. Spiegelman, 1997], "Variable selection in multivariate calibration of a spectroscopic glucose sensor", Applied Spectroscopy, vol. 51, n. 10, pp. 1559-1564, ISSN: 0003-7028.

[Mirats i Tur J.M., R. Verde, 2000], "Subsystem identification within a complex system", Simulation and modelling: enablers for a better quality of life. 14th european simulation multiconference, May 23-26, 2000, Ghent, Belgium, pp. 773-777.

[Mirats Tur, J.M., 1998], "Detecció i reconeixement de fallades en un avió comercial usant Fuzzy Inductive Reasoning", internal report IRI-Dt-9804, Institut de Robòtica i Informàtica Industrial, Universitat Politècnica de Catalunya, Barcelona, Spain.

[Mirats Tur, J.M., 1997], "Implementació de la metodologia FIR. Aplicació a un sistema lineal", IRI-Dt-9701, Internal Report.

[Mirats Tur, J.M., A. Escobet, 1997a], "Model quantitatiu i qualitatiu d'un motor de corrent continua estudi comparat", IRI-Dt-9703, Internal Report.

[Mirats Tur, J.M., A. Escobet, 1997b], "Model quantitatiu i qualitatiu d'un dipòsit d'aigua estudi comparat", IRI-Dt-9702, Internal Report.

[Mirats Tur, J.M., R. Huber, 1999], "Fuzzy inductive reasoning model based fault detection applied to a commercial aircraft", Simulation, Vol. 75, n.4, October 2000, pp.188-198, ISSN 0037-5497/00.

[Mirats, J.M., F.E., Cellier, R.H., Huber, S.J., Qin, 2000], "On the selection of variables for Qualitative Modelling of Dynamical Systems", submitted for publication in the International Journal of General Systems (May 2001).

[Mirats, J.M., F.E., Cellier, R.H., Huber, 2000], "Variable selection procedures and efficient suboptimal mask search algorithms in Fuzzy Inductive Reasoning", submitted for publication in the International Journal of General Systems (May 2001).

[Mitchell & Gauthier Associates, Inc., 1991], ACSL: Advanced Continuous Simulation Language - Reference Manual, Edition 10.0, Concord, Mass.

[Moore, B.C., 1981], “Principal component analysis in linear systems: controllability, observability and model reduction”, IEEE Transactions on Aut. Control, AC-26, pp. 17-32.

[Moorthy M., F.E. Cellier, J.T. LaFrance, 1998], "Predicting U.S. Food Demand in the 20th Century: A New Look at System Dynamics", Proc. SPIE Conference 3369: "Enabling Technology for Simulation Science II", part of AeroSense'98, Orlando, Florida, pp. 343-354.

[Mugica, F. and F. E. Cellier, 1993], “A new fuzzy inferencing method for inductive reasoning,” Proc. Sixth International Symposium on Artificial Intelligence, Intelligent Systems in Industry and Business, Monterrey, Mexico, pp. 372-379.

[Mugica, F., 1995], "Diseño sistemático de controles difusos mediante razonamiento inductivo difuso", Ph.D. thesis, Dept. Llenguatges i Sistemes informàtics, Universitat Politècnica de Catalunya.

[Muñoz, A., T. Czernichow, 1998], “Variable selection using feedforward and recurrent neural networks”, International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, vol. 6, n. 2, Jun 1998, pp. 91-102.

[Nebot A., A. Jerez, 1997], “Assessment of classical search techniques for identification of fuzzy models”, Proc. EUFIT’97, 5th European congress on intelligent techniques and soft computing, Aachen, Germany, 8-12 September, pp. 904-909.

[Nebot, A., 1994], "Qualitative Modeling and Simulation of Biomedical Systems Using Fuzzy Inductive Reasoning", Ph.D. dissertation, Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Spain.

[Nebot, A., F.E. Cellier, D.A. Linkens, 1993], "Controlling an anaesthetic agent by means of fuzzy inductive reasoning", Proceedings Quardet'93, IMACS Intl., Workshop on Qualitative Reasoning and Decision Technologies, Barcelona, Spain, June 16-18, pp. 345-356.

[Nebot, A., F.E. Cellier, and M. Vallverdú, 1998], “Mixed Quantitative/Qualitative Modelling and Simulation of the Cardiovascular System", Computer Methods and Programs in Biomedicine, 55(2), pp.127-155.

[Nebot A., F. Mugica, 1996], "Using Causal Relevancy for the Selection of Models within FIR Qualitative Modelling and Simulation Environment", Proc. of InterSymp’96, 8th International Conference on Systems Research Informatics and Cybernetics, "Advances in AI and Engineering Cybernetics"; Baden-Baden, Germany. pp. 137-142. (Artículo invitado). ISBN 0921836-42-2.

[Osten, D. W., 1988], “Selection of optimal regression models via cross-validation”, Journal of Chemometrics, Vol. 2, pp 39-48.

[Papoulis, A., 1991], "Probability, Random Variables and Stochastic Processes", 3d. ed. McGraw-Hill.

[Peña, D., 1989], “Estadística modelos y métodos. Modelos lineales y series temporales”, Alianza editorial, 2ª Ed.

[Qin, S. J., and R. Dunia, 1998], “Determining the number of principal components for best reconstruction”, Proc. of the 5-th IFAC Symposium on Dynamics and Control of Process Systems, pp. 359-364, Corfu, Greece, June 8-10.

[Readle, J.C., R.M. Henry, 1994], "On-line determination of time-delay using multiple recursive estimators and fuzzy reasoning", International Conference on Control '94 (Conf. Publ. No.389). IEEE, London, UK, 21-24 March 1994, vol.2, pp.1436-41, ISBN: 0852966113

[Riitta, H., P. Minkkinen, V.M. Taavitsainen, 1994], “Comparison of variable selection and regression methods in multivariate calibration of a process analyzer”, Process Control and Quality, vol. 6, n. 1, Aug 1994, pp. 47-54, ISSN: 0924-3089.

[SAS, 1988], SAS Language guide, Release 6.03 Edition, SAS Institute Inc., Cary, NC, USA.

[SAS, 1988], SAS Technical Report: P-179, Additional SAS/STAT Procedures. Release 6.03.

[SAS, 1988], SAS/STAT User’s guide, Release 6.03 Edition, SAS Institute Inc., Cary, NC, USA.

[Schatzoff, M., R. Tsao, S. Fienberg, 1968], "Efficient calculation af all possible regressions", Technometrics, Vol. 10, n. 4, pp. 769-779.

[Seixas, J.M., L.P. Caloba, I. Delpino, 1996], “Relevance criteria for variable selection in classifier designs”, Solving Engineering Problems with Neural Networks, Proceedings of the International Conference on Engineering Applications of Neural Networks (EANN'96), Turku, Finland, vol. 1, pp. 451-454.

[Shafer, G., 1976], "A mathematical theory of evidence", Princeton University Press, Princeton, N.J., USA.

[Shannon, C.E., W. Weaver, 1978], "The Mathematical theory of communication", 7th ed., Urbana, U.S.A., University of Illinois Press.

[Siciliano, R., F. Mola, 1996], “A fast regression tree procedure”, Statistical modelling, Proceedings of the 11th international workshop on statistical modelling (ed. A. Forcina et al.), pp. 332-340, Perugia: Graphos.

[Siciliano, R., F. Mola, 1997], “Multivariable data analysis and modelling through classification and regression trees”, Computing Science and Statistics (ed. E. Wegman & S. Azen), 29, 2, pp. 503-512, Interface Foundation on North America, Inc.: Fairfax.

[Smith, P.L., 1979], “Splines as a useful and convenient statistical tool”, The american statistician, 33, pp. 57-62.

[Stearns, S.D., R.A. David, 1996], "Signal processing algorithms in Matlab", Prentice Hall.

[Tanaka, H., Y. Kadono, T. Dohi, S. Osaki, 1999], “Pricing of stock index options and their correlation analysis”, Transactions of the Institute of Systems, Control and Information Engineers. Vol. 12, n. 7, July 1999, pp. 379-389.

[Tanaka, Y., M. Yuichi, 1997], “Principal component analysis based on a subset of variables: Variable selection and sensitivity analysis”, American Journal of Mathematical and Management Sciences vol. 17, n. 1-2, pp. 61-89, ISSN: 0196-6324.

[Trankle, T. L., P. Sheu, U.H. Rabin, 1986], “Expert system architecture for control system design”, Proc. of the 1986 American Control Conference, Seattle, WA, USA, 18-20 June, vol. 2, pp. 1163-1169.

[Uhrmacher, A.M., F.E. Cellier, R.J. Frye, 1997], "Applying Fuzzy-Based Inductive Reasoning to Analyze Qualitatively the Dynamic Behavior of an Ecological System", International Journal on Applied Artificial Intelligence in Natural Resource Management, 11(2), pp. 1-10.

[Uyttenhove, H. J., 1979], "SAPS - System Approach Problem Solver", PhD. Dissertation, SUNY Binghampton, New York.

[Uyttenhove, H.J., 1978], “Computer aided systems modelling: an assemblage of methodological tools for systems problem solving”, Ph.D. dissertation, School of advanced technology, University of New York, SUNY-Binghamton, USA.

[Valle, S., W. Li, S. J. Qin, 1999], “Selection of the Number of Principal Components: The Variance of the Reconstruction Error Criterion with a Comparison to Other Methods”, Ind. Eng. Chem. Res., 38, pp. 4389-4401.

[Van Welden, D. F., 1999], "Induction of Predictive Models for Dynamical Systems Via Datamining", Ph.D. dissertation, Toegepaste Wiskunde en Biometrie, Universiteit Gent, Belgium.

[Van Welden, D. F., E. J. H Kerckhoffs, G. C. Vansteenkiste, 1998], “Extending a Fuzzy Inductive Reasoner with Classification Procedures,” *Proc. European Simulation Symposium,* Nottingham, UK, Oct 26-28, pp. 111-116.

[Verde, R., 1994], “Funzioni B-spline e codifica delle variabili quantitative nell’analisi non lineare dei dati”, Ph.D, Dipartimento di Matematica e Statistica, Università degli Studi di Napoli Federico II.

[Vesanterä P. J., 1988],”Qualitative Simulation: A Tool for Global Decision Making”, M.S. Thesis (F. E. Cellier, Adv.), Dept. of Electrical and Engineering, Univ. of Arizona, Tucson, AZ.

[Vesanterä, P.J., F.E. Cellier, 1989], "Building intelligence into an autopilot using qualitative simulation to support global decision making", Simulation, 52:3, pp. 111-121.

[Widmann, J.M., S.D. Sheppard, 1995], “Algorithm for automated design variable selection in structural shape optimization with intrinsically defined geometry”, 21st Annual Design Automation Conference American Society of Mechanical Engineers, Design Engineering Division (Publication) DE vol. 82, n. 1, pp. 399-406.

[Wold, S., 1978], “Cross-Validatory estimation of the number of components in factor and principal components models”, Technometrics, Vol. 20, n. 4, November 1978, pp. 397-405.

[Wold, S., K. Esbensen, P. Geladi, 1987], "Principal component analysis", Chemometrics and intelligent laboratory systems, 2, pp. 37-52.

[Wong, E. 1971], "Stochastic processes in information and dynamical systems", McGraw-Hill.

[Young, F.W., 1981], “Quantitative analysis of qualitative data”, Psychometrika, 46, pp. 357-388.

[Young, F.W., J. De Leeuw, Y. Takane, 1976], "Regression with qualitative and quantitative variables: An alternating least squares approach with optical scaling features", Psychometrika, 41, pp. 505-529.