# Research on Hybrid Black-Box Modeling for Nonlinear Systems and Its Applications

Jinglu Hu

March 1997

# Abstract

Linear system theory is very well developed and there exist many results which can be applied to obtained linear models. On the other hand, most of real processes are nonlinear to some extent. If no physical insight is available and linear approximative models are not good enough, one has to use nonlinear black-box models. The existing nonlinear black-box models (neural networks, adaptive fuzzy systems, etc.), however, do not contain those linearity properties required by linear system theory, so that the results based on linear system theory can not be applied to the obtained nonlinear black-box models. The motivation of this thesis is intended to develop a black-box modeling scheme, with which the techniques based on well developed linear system theory could be extended to nonlinear systems. A hybrid black-box modeling scheme is proposed. Investigations are made to do system identification, system analysis and control design of nonlinear systems under the framework of linear system theory based on the new hybrid modeling scheme.

A black-box model is a standard flexible structure which can be used to approximate a large variety of different systems. In this thesis, a new black-box model structure is proposed by incorporating *a group of* certain nonlinear structures into a linear model structure. A general nonlinear system is first expressed in a linear structure whose coefficients consist of constant parameters and nonlinear terms. Then a group of certain nonlinear nonparametric models (NNMs) (neural networks, adaptive fuzzy systems, etc.) are incorporated into the linear structure by using them to represent the nonlinear terms. In this way, we obtain a hybrid model structure which provides more freedoms so that particular effort can be made to find a better compromise between the model flexibility and the model simplicity by using knowledge information efficiently. The obtained hybrid model is equipped with linear structure, flexibility and simplicity.

Parameter estimates are usually based on criterion minimization. When a model includes a noise model part, the criterion function is not always unimodal, even though the model is built to be linear in the parameters. In order to solve such multimodality problem, a hybrid identification method using Genetic Algorithms (GAs) is considered. Particular compromises provided by optimization-based methods and GAs are obtained through introducing a new GA operator named as 'development' inspired by the fact that living beings adapt themselves to their environment. The proposed hybrid method combines the reliability properties of the GAs with the accuracy of optimization-based method, while requiring a computation time only slightly higher than the latter. Furthermore, the hybrid identification is typically suitable for solving the multimodal problem resulted from noise models.

One of the most challenging problems is to do control design and system analysis of nonlinear systems using the techniques based on the linear system theory. Since the proposed hybrid black-box model has the required linearity properties, it enables us to solve this challenging problem. First as an example of system analysis, a fault detection scheme based on the use of Kullback discrimination information (KDI) for model discrimination is extended to nonlinear systems. Two ways are considered. One is robust fault detection like approach. A two-step identification algorithm is suggested to identify the proposed hybrid model in such a way that the results give a best linear approximation of the system and the estimate of the modeling error due to nonlinear undermodeling. Then KDI-based robust fault detection scheme is applied. The second is multi-model based approach, where the proposed model is used as an interpolation based multi-ARMAX-model consisting of several local linear ARMAX models. The fault detection is then performed by applying the KDI to discriminate the identified local ARMAX models. Next as an example of control design, a robust STR adaptive controller is designed for general nonlinear stochastic systems in a similar way to the linear stochastic control theory, based on the use of a hybrid quasi-ARMAX predictor. For such purpose, the hybrid quasi-ARMAX modeling scheme is modified so that the obtained hybrid quasi-ARMAX model is linear not only in the parameters to be adjusted but also in the one-step past input variable, which is favorable to deriving a control law directly.

# Preface

The common theme of this thesis is developing a hybrid black-box modeling scheme for identification, fault detection and control of nonlinear systems. The material is organized in seven chapters. Most of the material has been published or considered to publish in book chapters, journal papers and conference papers. Some of the material in Chapter 2 can be found in

> J. Hu, K. Kumamaru, and K. Inoue, "A Hybrid Quasi-ARMAX Modeling Scheme for Identification and Control of Nonlinear Systems", In *Proc. of the 35th IEEE Conference on Decision and Control (Kobe)*, 1996, pp.1413-1418.
>
> J. Hu, K. Kumamaru, and K. Inoue, "A Guaranteed Nonlinear System Identification Using ARX Networks", in *Proc. of the 27th ISICE International Symposium on Stochastic Systems Theory and Its Applications (Beppu)*, 1995, pp.7-12.
>
> J. Hu and K. Kumamaru, "Identification of Nonlinear Systems Based on Adaptive Fuzzy Systems Embedding Quasi-ARMAX Model", in *Proc. of the 34th SICE Annual Conference (international session), (Sapporo)*, 1995, pp.1211-1216.

which are further arranged into a journal paper:

> J. Hu, K. Kumamaru, and K. Inoue, "A Hybrid Quasi-ARMAX Modeling Scheme for Identification of Nonlinear Systems ", to be submitted to *Trans. of the Society of Instrument and Control Engineers*, 1996.

The material in Chapter 3 appeared in

> J. Hu, K. Kumamaru, and K. Inoue, "A Hybrid Robust Identification Using Genetic Algorithm and Gradient Method", *Trans. of the Society of Instrument and Control Engineers*, Vol.32, No.5, pp.714-721, 1996.

The material in Chapter 4 can be found in

> K. Kumamaru, J. Hu, K. Inoue and T. Söderström, "Statistical Methods for Robust Change Detection in Dynamical Systems with Model Uncertainty", in *Statistical Methods in Control and Signal Processing*, T. Katayama and S. Sugimoto, Eds., Mercel Dekker Inc., New York, USA, 1997 (to appear).
>
> K. Kumamaru, J. Hu, K. Inoue and T. Söderström, "Robust Fault Detection Using Index of Kullback Discrimination Information", in *Proc. of the 13th IFAC World Congress (San Francisco)*, Vol.N, 1996, pp.205-210.
>
> K. Kumamaru, J. Hu, K. Inoue and H. Ono, "Fault Detection via KDI in Presence of Unmodeled Uncertainty", in *Proc. of the 26th ISICE International Symposium on Stochastic Systems Theory and Its Applications (Osaka)*, 1994, pp.173-178.

The material in Chapter 5 can be found in

> K. Kumamaru, J. Hu, K. Inoue and T. Söderström, "A Method of Robust Fault Detection for Dynamic Systems by Using Quasi-ARMAX Modeling", in *Proc of the 11th IFAC Symposium on System Identification (Kitakyushu)*, 1997, (to appear).
>
> K. Kumamaru, J. Hu, K. Inoue and T. Soderstrom, "Fault Detection of Nonlinear Systems by Using Hybrid Quasi-ARMAX Models", Submitted to IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (Kingston Upon Hull), 1997.

Two journal papers based on Chapter 4 and Chapter 5 are under preparation

> "Robust Fault Detection Using the Index of Kullback Discrimination Information"
>
> "Fault Detection of Nonlinear Systems by Using Hybrid Quasi-ARMAX Models"

The material in Chapter 6 has been presented in

> J. Hu, K. Kumamaru, and K. Inoue, "Adaptive Control of Nonlinear Stochastic Systems Based on A Hybrid Quasi-ARMAX Model", *the 28th ISICE International Symposium on Stochastic Systems Theory and Its Applications (Kyoto)*, 1996.

which is extended into a journal paper

> J. Hu, K. Kumamaru, and K. Inoue, "Adaptive Control of Nonlinear Stochastic Systems Based on Hybrid Quasi-ARMAX Model", to be submitted to Trans. of the Institute of Systems, Control and Information Engineers, 1996.

## Acknowledge

First I would like to thank my supervisor Professor Kousuke Kumamaru for giving me the opportunity to join his group, for guiding me around all the obstacles, for providing me valuable freedom of research throughout the course of my Ph.D.

I am grateful to Professor Kousuke Kumamaru, Professor Masumi Ishikawa, Professor Fujio Ohkawa, and Professor Toshio Toyota who proofread all the thesis. Many thanks are for their valuable comments.

I am also very grateful to Associate Professor Katsuhiro Inoue for his discussion throughout my thesis research.

I would also like to thank all the staff and students in Kumamaru & Inoue laboratory, especially Makoto Maeda, Shingo Horinouchi for their efforts to keep the Workstation running and help me solve a lot of problems concerning the Workstation.

Finally, I would like to thank my parents, Zhu Xi and Gui Ying, my wife Lin Jing and my son Meng Qi for their love and support. This thesis is dedicated to them.

# Contents

# Glossary

Some notations may have different meaning locally.

## Notations

| | |
|---|---|
| $x^T$ | transpose |
| $u^N, U_N$ | $[u(1), ..., u(N)]^T$ |
| $Z^N$ | $[y^N, u^N]$ |
| $\chi^2(d)$ | $\chi^2$-distribution with $d$ degrees of freedom |
| $y(t)$ | output signal at time $t$ |
| $u(t)$ | input signal at time $t$ |
| $\varepsilon(t)$ | prediction error at time $t$ |
| $e(t), v(t), \nu_t$ | noise, system disturbance |
| $\varphi(t)$ | regression vector |
| $\bar{\varphi}(t)$ | regression vector $\varphi(t)$ whose element $u(t-i)$ changed to $q^{-1}u(t-i)$ |
| $\varphi_e(t)$ | regression vector including past prediction errors as elements |
| $\varphi_{NL}(t)$ | regression vector containing nonlinear regressors |
| $V_N(\theta)$ | loss function |
| $\theta, \Omega_j, \Theta, \Theta_c$ | parameter vectors |
| $\theta_e, \Omega_{ej}$ | parameter vector including the parameters of noise model |
| $\xi$ | parameter vector including the parameters describing unmodeled dynamics |
| $\Theta_t$ | coefficient vector which is function of input-output variables |
| $\hat{\theta}$ | estimate of $\theta$ |
| $p_j$ | scale and position parameter vector of the 'basis functions' in nonlinear nonparametric model |
| $\omega_{ij}$ | coordinate parameters of nonlinear nonparametric model |
| $\bar{a}_{i,t}$ | coefficients consisting of constant parameters and nonlinear terms |
| $n$ | number of old output values in $\varphi(t)$ |
| $m$ | number of old input values in $\varphi(t)$ |
| $r$ | number of old input and output values in $\varphi(t)$, $r = n + m$ |
| $l$ | number of old $\varepsilon(t)$ values in $\varphi_e(t)$ |
| $M$ | number of 'basis functions' in nonlinear nonparametric model, in particular for adaptive fuzzy systems, the number of rules |
| $\delta$ | dead zone in robust adaptive scheme |
| $\lambda_0$ | forgetting factor in recursive algorithm |
| $\lambda$ | weighting factor for control input in control law |

## Operators and Functions

| | |
|---|---|
| $\arg\min f(x)$ | minimizing argument of $f(x)$ |
| $\dim(\theta)$ | dimension of the vector $\theta$ |
| $q^{-1}$ | the backward shift operator, $q^{-1}f(t) = f(t-1)$ |
| $Ex$ | Expected value of stochastic variable $x$ |
| $\wedge$ | minimum operator, $0.8 \wedge 0.3 = 0.3$ |
| $\otimes$ | Kronecker production |
| $\prod$ | production |
| $\mu_{A_i^j}(x)$ | fuzzy membership function of fuzzy set $A_i^j$ |
| $\mathcal{N}_f(x)$ | 'basis function' in the nonlinear nonparametric model |
| $p(y(t+1)\|\hat{\theta}, Y_t, U_t)$ | likelihood function of model $\mathcal{M}\{\hat{\theta}\}$ |
| $G(q^{-1})$ | rational function in $q^{-1}$ describing system dynamics |
| $\Delta G(q^{-1})$ | unmodeled dynamics |
| $H(q^{-1})$ | rational function in $q^{-1}$ describing noise dynamics |

## Abbreviations

| | |
|---|---|
| KDI | Kullback discrimination information |
| NNM | nonlinear nonparametric model |
| GA | Genetic Algorithm |
| NSGA | non-standard Genetic Algorithm |
| FDI | fault detection and isolation |
| SISO | single-input-single-output |
| ARMAX | autoregressive moving average model structure with exogenous inputs |
| ARX | autoregressive model structure with exogenous input |
| MA | moving average model structure |
| FDS | fault detection system |
| STR | self-tuning regulator |
| LIP | linear in the parameters |
| AFS | adaptive fuzzy system |
| RBFN | radial basis function network |
| NN | neural network |
| WN | wavelet networks |
| PEM | prediction error method |
| ELS | extended least square method |
| ML | maximum likelihood method |
| RMS | root mean square |
| IIR | infinite impulse response model |
| FIR | finite impulse response model |

# Chapter 1

# Introduction

## 1.1 System Identification

A model describes reality in some way, and system identification is the theory of how mathematical models for dynamical systems are constructed from observed data.

Typically, a parameterized set of models, a *model structure*, is hypothesized and data is used to find the best model within this set according to some criterion. The choice of model structure is guided by prior knowledge or assumptions about the system which generates the data. It is customary to distinguish between three levels of prior knowledge, which have been color-coded as follows [92]

- White Box models: This is the case when a model is perfectly known; it has been possible to construct it entirely from prior knowledge and physical insight.
- Grey Box models: This is the case when some physical insight is available, but several parameters remain to be determined from observed data. It is useful to consider two subcases:
  - Physical Modeling: A model structure can be built on physical grounds, which has a certain number of parameters to be estimated from data. This could, e.g., be a state space model of given order and structure.
  - Semi-physical modeling: Physical insight is used to suggest certain nonlinear combinations of measured data signal. These new signals are then subjected to model structures of black box character.
- Black Box models: No physical insight is available or used, but the chosen model structure belongs to families that are known to have good flexibility and have been "successful in the past".

The terms of "white-box model", "grey-box model" and "black-box model" may not be so popular, but in system identification community, they have been frequently used recently, see e.g. [92, 46]. In this thesis, we also use the term "black-box model" based on the above definition.

### 1.1.1 Black-Box Modeling

When little prior knowledge is available, it is common to use a *black-box* model. A black-box model is a standard flexible structure which can be used to approximate a large variety of different systems. In real applications it is impossible to obtain a model structure that is capable of describing the system exactly. Instead one tries to make reasonable assumptions about the system so that the hypothesized model structure is "close" to the true system.

One common assumption in system identification is that the unknown system is linear. This is never true in real applications, but often it is a good approximation. Linear system theory is very well developed and there exist many results which can be applied to the obtained linear models. Some typical linear black-box model structures are reviewed in Appendix A.1.

If the linear assumption is relaxed, one turns to nonlinear models. The nonlinear black-box situation is much more difficult. The main reason for that is that nothing is excluded, and a very rich spectrum of possible model descriptions must be handled. In recent years, nonlinear modeling and identification have attracted much interest in control and system identification community. Many nonlinear models have been proposed in the literatures: 'classic' models derived from Volterra series or Winner series [8, 10], and nonlinear black-box models based on the nonlinear nonparametric models (NNMs) (neural networks, fuzzy models, wavelet, etc.) [92, 46]. Especially, the latter ones have gained increasing interest due to their ability to encompass truly nonlinear behaviors. Appendix A.2 gives a summary for those nonlinear black-box structures.

As pointed in Appendix A.2, a great deal of attention for the nonlinear black-box models is so far paid only to the flexibility of the model structures. The structural linearity and simplicity, which are important features of highly successful linear black-box models, have been ignored. That is, there is a gap between the existing linear and nonlinear black-box models. It is highly motivated to develop a black-box modeling scheme to fill the gap so as to extend the well developed linear system theory to nonlinear systems. A hybrid (linear-nonlinear) modeling scheme seems to be the best idea. In the literature, some authors have used a "linear model + neural network" type hybrid scheme for identification and control design of nonlinear system [44, 102, 34]. However as a hybrid black-box modeling scheme, linear structures and nonlinear structures should be combined in a more effective and efficient way. In this thesis, we will propose a hybrid quasi-linear black-box modeling scheme which is obtained by incorporating a group of certain NNMs into a linear structure. Particular effort will be made to find a better compromise to the trade-off between the model flexibility and the model simplicity using knowledge information efficiently. The obtained hybrid model is equipped with a linear structure, flexibility and simplicity, which enables us to do identification, system analysis and control design of nonlinear systems under the framework of linear system theory.

### 1.1.2 Parameter Estimation

Fitting a model within a given structure (parameter estimation) is in most cases a lesser problem. However, since model parameters are usually determined as the global minimum point of the loss function which is not guaranteed to be unimodal, an optimization-based algorithm has a potential risk to be stuck at a local minimum [94]. There is no easy solution to the multimodality problem [92]. A nice way to solve this problem so far is to search a good initial value where to start the iterations or to estimate the parameters directly using global search techniques such as random search and genetic algorithms [88, 90, 51]. However, all these techniques are rather time-consuming, and not so effective when the number of parameters to be estimated is large.

The best way for solving the problem, however, is to construct a model which is linear in the parameters to be estimated, since the criterion function becomes simple in such case and its multimodality appears only when a noise model is employed for system disturbance. In this thesis, the proposed hybrid quasi-linear black-box model has been constructed into a form that is linear in the parameters to be estimated, the feature of which is called as 'model simplicity'. On the other hand, there are few publications dealing with the multimodal problem resulted from noise models. In some cases, the reliability of parameter estimation is very important. Therefore, it is crucial to develop an algorithm which may find global minimum reliably. Since in the case of multimodality resulted from noise model, the fitness is difficult, if not impossible, to calculate, the conventional genetic algorithms (GA) can not be employed. To solve this problem, we will propose a Non-Standard GA (NSGA) by introducing a new GA operator named as *development* inspired by the fact that living beings adapt themselves to their environment, and perform the operation by using an optimization-based method. The NSGA is very efficient in parameter estimation and can be used to solve the multimodal problem resulted from noise model. Then a hybrid identification algorithm is developed, in which the NSGA is used to search for a good initial value and the estimation is continued using an optimization-based method with the initial value.

## 1.2 Fault Detection of Dynamical Systems

Due to the increasing complexity and riskiness of modern control system and the growing demands for quality, cost efficiency, availability, reliability and safety, the call for fault tolerance in automatic control systems is gaining more and more importance. One of the approaches to achieve fault tolerance is to provide fault *accommodation*, i.e. a reconfiguration of the system when a fault has occurred. For the fault accommodation, one of the most important and difficult tasks is fault *diagnosis*. The purpose of fault diagnosis is to detect the faults of interest and their causes early enough so that a failing of overall system can be avoided, which consists of three tasks: fault detection, fault isolation and fault analysis. In this thesis, we concentrate on the first task, i.e., fault detection.

Depending on the method of residual generation, the methods of fault detection can be divided into three categories: signal-based, model-based and knowledge based. The model-based approach has received an increasing attention recently because of its portability. Many methods of model-based residual generation have been developed during the last two decades, see [19, 21, 41, 3]. They can be divided into three groups: parity space approach [13, 15, 97], observer-based approach [22, 84] and parameter estimation approach [4, 41]. Several researchers have pointed out that there are close relationships among the different approaches [20].

### 1.2.1 Parameter estimation approach

The parameter estimation approach is based on the assumption that the faults are reflected in the physical system parameters such as friction, mass, viscosity, capacitance, inductance, etc. The basic idea of the detection method is that the parameters of the actual process are repeatedly estimated online using well known parameter estimation methods and the results are compared with the parameters of the reference model obtained initialy under fault-free conditions. Any substantial discrepancy indicates a change in the process and may be interpreted as a fault. Many researchers have made contributions to this approach, e.g. Isermann et al. [41], Rault et al. [87], Goodwin et al. [26] and Kumamaru et al. [61, 60].

Since an exact mathematical modeling of the system is impossible in practice, the effects of modeling uncertainties have to be taken into account with respect to which the residuals must be robust [20]. A typical scheme to the robust fault detection so far is proposed by Frank and Wunnenberg (1989) [22] and Patton and Kangethe (1989) [84] based on observer design approaches, in which the robustness is achieved by appropriately design the observers. In their approach, however, the distribution matrices of fault should be known in the state-space representation. In this thesis, we will propose an input-output model-based robust fault detection scheme, which is obtained by extending the fault detection scheme using Kullback discrimination information (KDI) as an index for model discrimination [61] to the cases of unmodeled dynamics. The idea is to apply the KDI to discriminate the identified models with unmodeled dynamics and to consider the estimates of unmodeled dynamics in the the KDI analysis and thresholding decision for robustness realization.

### 1.2.2 Fault Detection of Nonlinear Dynamical Systems

When the system to be diagnosed is nonlinear, the thing becomes difficult. One of the main reasons is that it is available few nonlinear black-box models whose parameters have useful physical interpretations that are important for fault analysis. Recently, neural networks based methods have been proposed for fault detection and isolation of nonlinear systems. Neural networks are typical used in two ways: (1) as classifiers; (2) as nonlinear black-box models [83, 96, 108, 95, 74]. Recently, Q. Zhang (1996) [118] proposed a method using nonlinear black-box models in fault detection and isolation. In all these approaches, however, the information provided by model parameters can not be used. In this thesis, we will propose an alternative method for fault detection of nonlinear systems, in which the identified model parameters can be used in the fault analysis. Since the hybrid quasi-linear black-box modeling scheme proposed can be considered as a modeling approach based on global and local linear approximations of nonlinear systems, the model parameters have global or local physical interpretations. Based on these features, we will extend the KDI-based fault detection scheme to nonlinear systems by applying the KDI to discriminate the identified hybrid quasi-linear black-box models. Two strategies will be considered. One is a robust fault detection like approach, in which the hybrid quasi-linear black-box model is identified in such a way that the results give a best linear approximation of system and the estimates of error due to nonlinear undermodeling. The second is a multi-model based approach, in which the hybrid quasi-linear black-box model will be used as a multi-model consisting of several local linear model.

## 1.3 Adaptive Control of Nonlinear Systems

When the systems to be controlled contain unknown parameters, adaptive controller offers certain advantages over conventional controller. Adaptive control theory based on linear models has been developed into a considerable mature stage [26, 63]. However, in the case of black-box type nonlinear systems to be controlled, things become difficult. The difficulty is that a linear black-box model can not provide enough accuracy, while a suitable nonlinear model is rather difficult to find. Recently, many authors suggested to use neural networks as a nonlinear black-box model for the adaptive control and proposed many approaches [40]. Generally, neural networks have been incorporated into adaptive control systems in two ways. One is to use neural networks as nonlinear models . Most of such schemes are working as indirect control [76, 75]. The second is to use neural networks as nonlinear compensators to compensate the system nonlinearity or the error due to nonlinear undermodeling [102, 34]. Loosely speaking, because of the nonlinearity in the parameters to be adjusted, the control systems incorporating neural networks usually have some drawbacks concerning the *convergence rate* and the *noise sensitivity* [39].

On the other hand, since the adaptive control theory based on linear models is very well developed, it is highly motivated to develop a nonlinear black-box modeling scheme so that the existing adaptive control theory becomes applicable to general nonlinear systems. In this thesis, we will propose a hybrid quasi-ARMAX modeling scheme for such purpose. Particular efforts will be made to obtain a *nonlinear* black-box model which (1) has a similar form to linear ARMAX model; (2) is linear not only in the parameters to be adjusted but also in the one-step past input variable ($u(t-1)$) that should be synthesized in a control system. We believe that this is a perspective approach for adaptive control of general nonlinear systems. To demonstrate such perspective, we will design a STR adaptive controller for general nonlinear stochastic system under the framework of linear stochastic control theory by using the hybrid quasi-ARMAX predictor structure, and apply the STR adaptive controller to a variety of nonlinear stochastic systems to test its effectiveness.

## 1.4 Thesis Outlines and Main Contributions

The thesis consists of seven chapters. Chapter 1 gives a background and an outline for whole thesis. Chapter 2 and Chapter 3 are devoted to system modeling and identification. Chapter 4, Chapter 5 and Chapter 6 are dealing with applications for fault detection and control design. Finally, Chapter 7 gives a summary for the whole thesis (Conclusions).

**Chapter 2**: Chapter 2 is devoted to developing a hybrid modeling scheme. A general nonlinear system is first expressed in a linear structure, then a group of nonlinear nonparametric models (NNMs) are incorporated into the linear structure, so that a hybrid quasi-linear black-box model structure is obtained. Estimations of the hybrid model using both knowledge information and observed data are discussed. Several real systems and simulated systems are used to test the proposed scheme. It is also point out that further investigation is needed to develop an algorithm for using knowledge information efficiently in the parameter estimation (*intelligent identification*).

The main achievement in Chapter 2 is that a new hybrid model structure is proposed, which incorporates *a group of* NNMs into a *linear model structure*. The new hybrid model structure is distinctive to other hybrid ones in the following issues.

- It is possible to provide better compromises to the trade-off between the model flexibility and the model simplicity and thus to obtain a model equipped with a linear structure, flexibility and simplicity.

- The obtained model can be considered to be an extension of linear models, which avoids abandoning the properties of highly successful linear models as soon as a small amount of nonlinearity introduced.
- The parameters are divided into two groups, one of which is estimated from observed data, while the other of which is determined using knowledge information. This may give an alternative perspective for *intelligent identification.*
- It is easier to do identification, system analysis and control design for nonlinear systems with the new model structure, since the techniques based on well developed linear system theory may be applied.

**Chapter 3**: Chapter 3 discusses hybrid methods using Genetic Algorithms for global optimization. Particular compromises provided by traditional optimization-based methods and Genetic Algorithms (GAs) are addressed and illustrated by a particular application in the case of identifying a general system (linear, nonlinear, etc.) based on a linear ARMAX model.

The main contribution in Chapter 3 is introducing a new GA operator named as *development*, which is inspired by the purpose to solve multimodal problem resulted from noise models and the fact that living beings adapt themselves to their environment and is performed by optimization-based methods. The obtained Non-Standard GA (NSGA) is different with other hybrid GAs because it combines the reliability properties of the GAs with the accuracy of optimization-based method, while requiring a computation time only *slightly higher* than the latter. Furthermore, it is possible to solve the multimodal problem resulted from noise model, which it is known to be difficult to solve via conventional GAs.

**Chapter 4**: Chapter 4 concerns with the KDI-based robust fault detection of dynamic systems. We first point out that in the fault detection scheme based on the use of Kullback discrimination information (KDI) for model discrimination, there are two important assumptions: (1) The system to be diagnosed is linear; (2) The model parameterization is chosen adequately for the system, which are never true in real applications. Then we devote the rest of chapter to relaxing the assumption (2) by extending the scheme into robust ones. The robust fault detection is performed by applying the KDI to discriminate the identification linear model with unmodeled dynamics and considering the estimate of unmodeled dynamics in the KDI analysis and thresholding decision for robustness realization.

**Chapter 5**: Chapter 5 is devoted to relaxing the assumption (1) so far made in the KDI-based fault detection scheme, that is, extending the scheme to nonlinear systems. It is first shown that the hybrid quasi-ARMAX model can be transformed into a combined form of a linear ARMAX model and a multi-ARX-model consisting of several local linear ARX models. Next, the hybrid is applied to fault detection of nonlinear systems in two ways: robust fault detection like approach and multi-model based approach.

The main contribution in Chapter 4 and Chapter 5 is to relax the two important assumptions so far made in the KDI-based fault detection scheme, which are never true in real applications. In Chapter 4, the fault detection scheme is extended to the case of unmodeled dynamics. The contributions related to this extension are that

- it is first time to construct a robust fault detection system in a similar way to robust control theory, that is, to build a fault detection system based on a model with unmodeled dynamics which is quantified in some forms.
- the KDI is analyzed with incorporating the description of unmodeled dynamics, by which it can be evaluated in a feasible way. And several indexes for evaluating unmodeled dynamics are also introduced.

- two fault decision schemes are introduced for robustness realization so that the fault detection is carried out on the basis of the KDI, the index of unmodeled dynamics and other information about the systems.

In Chapter 5, the fault detection scheme is extended to nonlinear systems. The contributions related to the extension are that

- a new model-based fault detection scheme is proposed for black-box type nonlinear systems. It is distinctive in that the model is based global and local linear approximations of the system, so that its parameters have physical interpretations which are useful in the followed fault analysis.
- a two-step identification algorithm is suggested to identify the hybrid quasi-ARMAX in such a way that the results give a best linear approximation of the system and the estimate of the error due to nonlinear undermodeling. Such identification scheme can also be considered as an alternative approach for describing the error due to nonlinear undermodeling. It may have potential applications in robust control.
- a multi-model based approach is proposed for KDI-based fault detection scheme, in which the hybrid quasi-ARMAX model is used as a multi-ARMAX-model consisting of several local linear ARMAX models. The approach is distinctive in that the KDI is not applied to discriminate the identified model describing the system as a traditional way, instead it is applied to discriminate the identified local linear models, of which the system model is composed.

**Chapter 6**: Chapter 6 concerns adaptive control of nonlinear systems. First, based on the basic idea of quasi-ARMAX modeling and the fact that the NNMs have universal approximation ability, a hybrid quasi-ARMAX predictor structure is proposed for adaptive control of general nonlinear stochastic systems. Next, based on the predictor structure, a robust STR nonlinear control is developed in a similar way to the linear stochastic control theory. Finally, a variety of nonlinear stochastic systems are used to test the developed robust nonlinear STR adaptive controller.

Chapter 6 offers two contributions. One is that we propose a hybrid quasi-ARMAX predictor structure for adaptive control of general nonlinear stochastic systems. It is distinctive in that

- the hybrid quasi-ARMAX predictor is linear in the parameters to be adjusted.
- the hybrid quasi-ARMAX predictor is linear in the one-step past input variable, so that a control law can be derived directly.

The second is that a robust nonlinear STR controller is developed under the framework of linear stochastic control theory. The controller has the following advantages compared with the controller based on Neural Networks:

- it has a simple structure, which is similar in form to the linear STR controller. In fact, it can be considered as an extension of linear STR controller.
- it has better convergence properties, since the prediction model is linear in the parameters to be adjusted.
- it is not so sensitive to noise, because it is a stochastic STR adaptive controller.

# Chapter 2

# Hybrid Quasi-Linear Black-Box Modeling and Identification

## 2.1 Introduction

The key problem in system identification is to find a suitable model structure, within which a good model is to be found. According to the levels of prior knowledge used, there are three types of models: white-box models, grey-box models and black-box models. When no physical insight is available or used, one has to choose black-box model structure which belongs to families that are known to have good flexibility and have been "successful in the past".

Under the assumption that the unknown system is linear, linear black-box models can be chosen for the system identification. The identification based on linear approximation has been extensively and successfully handled within some well known linear black-box structures [70, 69, 94]. If the linear assumption is relaxed, one has to use nonlinear black-box models. For nonlinear black-box modeling, the "classic" literature seems to have concentrated on global basis function expansions, such as Volterra expansions [10]. These have apparently had limited success. Recently, some authors have suggested the use of nonlinear structures based on neural networks (NN), wavelet networks (WN), radial basis function networks (RBFN), etc, and have achieved reasonable success, see [92, 46]. However, the latter ones have resulted in an abrupt abandonment of the highly successful linear black-box modeling methods, which have some useful properties, as soon as a small amount of nonlinearity is introduced.

From a user's point of view, a nonlinear black-box model is preferred to have the following properties:

- **A linear structure.** In order to take advantage of linear system theory that is very well developed, a linear structure may be useful. Therefore, one would benefit by constructing a nonlinear black-box model as an extension of linear model instead of abandoning the properties of linear model totally.

- **Flexibility.** Since a nonlinear system can be nonlinear in so many ways, a nonlinear black-box model structure, in general, must be feasible enough to deal with various nonlinear systems.

- **Simplicity.** A nonlinear black-box model usually offer a large amount of parameters. If the model is constructed to be linear in the parameters, its estimation becomes simple. Furthermore, if the model is constructed to be linear in the one-step past input, it is simple to derive a control law based on the model.

Unfortunately, no existing nonlinear black-box models have those properties simultaneously. For those nonlinear black-box model structures, the three features appear to be conflicting ones. For example in the nonlinear black-box models described by (A.16) in Appendix A, if the scale and position parameter vectors $\boldsymbol{p}_j$ are determined *a priori* for simplicity, it is necessary to use a rather fine grid of 'basis functions' in order to get a sufficient approximation capability, which then typically would lead to many coordinate parameters $\omega_j$. Therefore, it is highly motivated to develop a modeling scheme in order to obtain a nonlinear model equipped with a linear structure, flexibility and simplicity. For this purpose, we have the following motivations:

- To find a better compromise to the trade-off between the conflicting properties, since some of the properties, for instance *flexibility* and *simplicity*, usually appear to be conflicting ones.
- To divide the model parameters into two groups, and to determine one of which by using knowledge information, since some of knowledge information are always available or can be obtained via some ways in practice.

It seems natural to consider the use of hybrid (linear-nonlinear) structure [1] for realizing these motivations. In the literature, some authors have employed a "linear model + neural network" type hybrid scheme for identification and control of nonlinear systems, in which neural network is simply used as a compensator to describe the error due to nonlinear undermodeling [44, 102, 34]. Such simple hybrid scheme does not, however, seem to have the properties of simplicity. In this chapter, we will propose a new hybrid model structure based on an effective combination of a linear structure and a group of certain nonlinear nonparametric models (NNMs) (neural networks, adaptive fuzzy systems, etc.). The basic idea of such hybrid modeling is first to increase the overall model flexibility by using a group of certain NNMs and then to restrict the flexibility in the higher order nonlinearity band to achieve the model simplicity. Note that by saying 'nonlinearity band' we have used a concept of 'nonlinearity spectrum'[2], which may be not so strict. The model constructed using this idea will be favorable to the lower order nonlinearity band than the higher one, which may give a property that it is more robust to noise in the parameter estimation. Furthermore, the idea also accords with the motivation to obtain a nonlinear model by extending the linear model to nonlinear one without totally abandoning the properties of the linear model.

It is shown that a general nonlinear system can be expressed in a linear structure whose coefficients consist of constant parameters and nonlinear terms. The structure allows us to incorporate a group of certain NNMs into the linear structure by using them to represent the nonlinear terms. Since in the hybrid structure, what each NNM used to represent is only *one* nonlinear term of the *coefficients*, i.e., the role required for each NNM is reduced, the flexibility of individual NNM can be restricted to some extent. Therefore, the parameters specifying the 'basis functions' in the NNM can be appropriately determined by using knowledge information. The efficient use of various knowledge information will play a key role on the hybrid modeling. The model built in this way is named as *hybrid quasi-linear black-box model*, which has a linear structure, flexibility and simplicity.

This chapter is organized as follows: Section 2.2 proposes a hybrid quasi-linear black-box modeling scheme by joining the linear and nonlinear black-box models together. Our discussions are concentrated on the linear structure of the proposed hybrid model, its flexibility for describing various nonlinear systems and its simplicity for parameter estimation. In Section 2.3, we discuss the estimation of the hybrid model based on criterion optimization. In Section 2.4, we point out a possible perspective: intelligent identification of the hybrid model. Several numerical examples with real data and simulated data are given in Section 2.5, The relations between the proposed hybrid model and some existing models, and the possible applications are discussed in Section 2.6.

## 2.2 Hybrid Quasi-Linear Black-Box Modeling

In this section, we will propose a hybrid quasi-linear black-box modeling scheme. Without loss of generality, our discussions will be concentrated mainly on ARMAX model structure because of its popularity. The results can however be extended to general black-box model structure by considering the general regression vectors which are referred to Appendix A.1.

[1] By "hybrid structure", we mean that the structure is obtained by combining the linear model structure in 'classic' literature with the nonlinear model structure in AI (artificial intelligent) literature. Therefore, the term "hybrid" means a combination of 'classic' approach and 'AI' approach.

[2] Let $y(t) = g(\varphi(t))$ be a nonlinear function. Performing Taylor expansion to $g(\varphi(t))$ around the region $\varphi(t) = \varphi_0(t)$, we have $y(t) = \alpha_0 + \alpha_1(\varphi(t) - \varphi_0(t)) + ||\varphi(t) - \varphi_0(t)||^2_{\alpha_2} + ....$ Then the norms of coefficients $\alpha_i$ $(i = 0, 1, 2, ...)$ form a 'nonlinearity spectrum'.

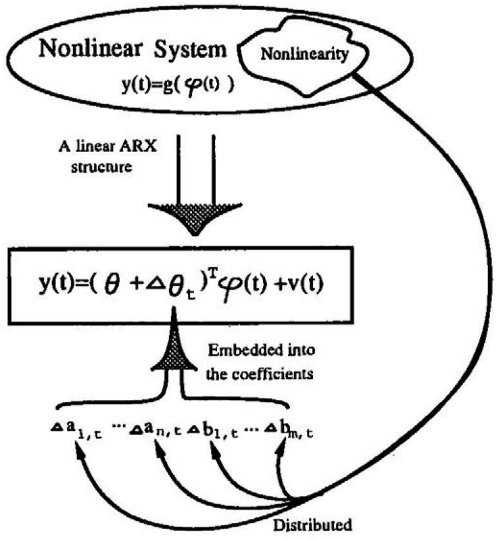

Nonlinear System
Nonlinearity
y(t)=g( φ(t) )
A linear ARX structure
y(t)=( θ +Δθ_t )^T φ(t) +v(t)
Embedded into the coefficients
Δa_{1,t} ···Δa_{n,t} Δb_{1,t} ··· Δb_{m,t}
Distributed

Figure 2.1: A schematic description of quasi-ARX modeling.

### 2.2.1 Hybrid Quasi-ARMAX Modeling

Let us consider SISO general nonlinear ARX (NARX) systems described by

$$\mathcal{S}: \quad y(t) = g(\varphi(t)) + v(t) \tag{2.1}$$

$$\varphi(t) = [y(t-1) \;...\; y(t-n) \;\; u(t-1) \;...\; u(t-m)]^T \tag{2.2}$$

where $y(t)$ is the output at time ($t = 1, 2, ...$), $u(t)$ the input, $\varphi(t)$ the regression vector, $v(t)$ the system disturbance, and $g(\,\cdot\,)$ the unknown continuously differentiable nonlinear function.

In Appendix B, we have shown that a general NARX system (2.1) can be expressed in a hierarchical structure, e.g. ARX networks, provided that $g(\,\cdot\,)$ is continuously differentiable. Based on this result, we can express the system (2.1) in a linear ARX structure, whose coefficients consist of constant parameters and nonlinear terms, see Fig. 2.1 where we call it quasi-ARX modeling in order to distinguish it to the NARX modeling

$$y(t) = \varphi^T(t)(\theta + \Delta\theta_t) + v(t) \tag{2.3}$$

where

$$\left.\begin{array}{l} \theta = [a_1 \;...\; a_n \;\; b_1 \;...\; b_m]^T \\ \Delta\theta_t = [\Delta a_{1,t} \;...\; \Delta a_{n,t}\Delta b_{1,t} \;...\; \Delta b_{m,t}]^T \end{array}\right\}. \tag{2.4}$$

and where the index $t$ in the nonlinear terms $\Delta a_{i,t}$ and $\Delta b_{i,t}$ denotes that they are functions of the regression vector $\varphi(t)$, instead of constants. We may call such treatment as "nonlinearity embedding" technique, which is actually not new. The techniques of embedding certain kind of nonlinear elements in the coefficients of a linear model are known to be effective in modeling some real processes [82]. When a moving average (MA) noise model is employed for the system disturbance $v(t)$, we have a *a quasi-ARMAX model*

$$y(t) = \varphi^T(t)(\theta + \Delta\theta_t) + C(q^{-1})e(t) \tag{2.5}$$

where $C(q^{-1}) = 1 + c_1q^{-1} + ... + c_lq^{-l}$ ($q^{-1}$: backward shift operator), and $e(t)$ is white noise.

For the case where the $g(\varphi(t))$ is a nonlinear system expressed in Kolmogorov-Gabor polynomial described by (B.2) of Appendix B, in which $\varphi(t)$ is assumed to be $[x_i(t),\ i = 1, ..., r]$, the nonlinear terms $\Delta a_{i,t}$ and $\Delta b_{i,t}$ can be explicitly expressed as (B.6) and (B.7), respectively. However for general nonlinear systems, the nonlinear terms $\Delta a_{i,t}$ and $\Delta b_{i,t}$ may become very complicated functions. We therefore represent them using the NNMs described by (2.7)

$$\left.\begin{array}{l}\triangle a_{i,t} = f_i(\varphi(t)) \quad (i = 1, ..., n) \\ \triangle b_{j,t} = f_{j+n}(\varphi(t)) \quad (j = 1, ..., m)\end{array}\right\} \tag{2.6}$$

$$f_i(\varphi(t)) = \sum_{j=1}^{M} \omega_{ij} \mathcal{N}_f \left(p_j, \varphi(t)\right) \tag{2.7}$$

where $\mathcal{N}_f \left(p_j, \varphi(t)\right)$'s are the 'basis functions', $\omega_{ij}$'s are the coordinate parameters, and $p_j$'s are the scale and position parameter vectors. The $\mathcal{N}_f(\cdot)$ is referred as 'basis functions', since the role they play in (2.7) is similar to that of a functional space basis. In some particular situations, they do constitute a functional basis. Typical examples are wavelet bases [46]. Without loss of generality, the NNMs used to realize $f_i(\varphi(t))$ are assumed to have the same structure, so that the 'basis function' is independent of the index $i$. A preferable candidate for the NNMs is *adaptive fuzzy systems* (AFS) [110], which can be explicitly expressed as

$$f_i(\varphi(t)) = \frac{\sum_{j=1}^{M} \omega_{ij} (\wedge_{k=1}^{r} \mu_{A_k^j}(x_k(t)))}{\sum_{j=1}^{M} (\wedge_{k=1}^{r} \mu_{A_k^j}(x_k(t)))} \tag{2.8}$$

where $\wedge$ is the minimum operator, $M$ is the number of rules, $x_k(t)$ are the elements of $\varphi(t)$, and $\mu_{A_k^j}$ is the membership function of fuzzy set $A_k^j$. The model described by (2.5)-(2.7) is named as *hybrid quasi-ARMAX model.*

It is well known that the NNMs are flexible enough to represent most reasonable systems in practice. It should be noted that there are a group of such certain NNMs in the hybrid quasi-ARMAX model. The hybrid quasi-ARMAX model becomes so flexible (complex) that it is impossible to estimate all of the parameters ($a_i$, $b_i$, $c_i$, $\omega_{ij}$ and $p_j$) from observed data as usual. In order to make the problem feasible, we will appropriately determine the parameter vectors $p_j$ specifying the 'basis functions' in the NNM by using knowledge information, and only estimate $a_i$, $b_i$, $c_i$ and $\omega_{ij}$ from observed data. We will discuss these estimations in the next section.

The followings are some interpretations for the hybrid quasi-ARMAX model:

## (1) Expression in Linear ARMAX Structure

Introducing a coefficient vector $\Theta_t$

$$\Theta_t = \theta + \triangle\theta_t \tag{2.9}$$

we have an expression of the hybrid quasi-ARMAX model given by

$$\mathcal{M}: \quad y(t) = \varphi^T(t)\Theta_t + C(q^{-1})e(t). \tag{2.10}$$

From (2.10), we can see that the hybrid quasi-ARMAX model has a linear ARMAX structure. It can be shown that such a linear ARMAX structure is useful for control design and system analysis, see Chapter 4, 5 and 6.

## (2) Expression in Combined Structure

Using (2.6), (2.7) and (2.4) in (2.5) we can obtain another expression of the hybrid quasi-ARMAX model as

$$\mathcal{M}: \quad y(t) = \varphi^T(t)\theta + C(q^{-1})e(t) + \sum_{j=1}^{M} \underbrace{\varphi^T(t)\Omega_j}_{\text{ARX}} \mathcal{N}_f \left(p_j, \varphi(t)\right) \tag{2.11}$$

where $\Omega_j = [\omega_{1j} ... \omega_{rj}]^T$. The expression (2.11) shows that the hybrid quasi-ARMAX model is equivalent to a model combining a linear ARMAX model and a multi-ARX-model. The multi-ARX-model consists of $M$ local ARX models and its overall performance is obtained via an interpolation using the 'basis function' $\mathcal{N}_f(x)$. It also implies that

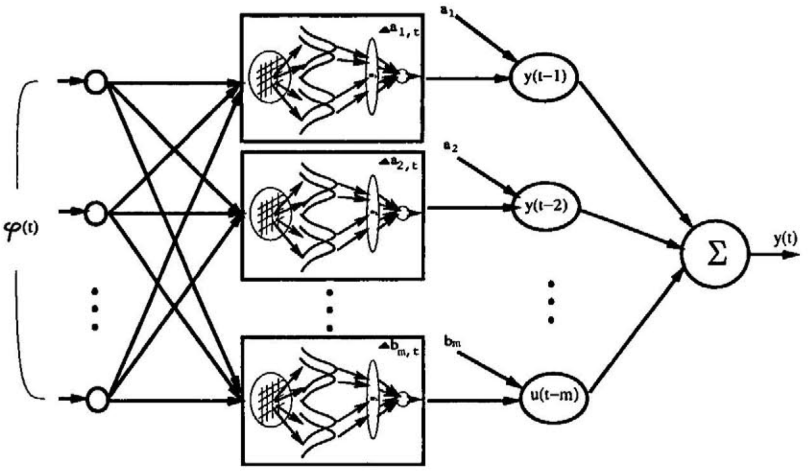


Figure 2.2: The hybrid quasi-ARMAX model shown as an associative memory networks. (the MA noise model has be omitted for clarity.)

- the proposed model can be shown to be able to describe any sufficiently smooth nonlinear function in (2.1) on a compact interval arbitrarily well by merely increasing the value of $M$.
- the proposed model is based on global and local linear approximations of system, which is favorable to lower order nonlinearity band. This makes the robustness of model to noise increased in the parameter estimation.

From the expression (2.11), we can see that the hybrid scheme proposed here is different from the "linear model + neural network" type hybrid schemes appeared in the literature [44].

## (3) Expression in Linear Regression Structure

Since the hybrid quasi-ARMAX model has been constructed to be linear in the parameters to be estimated, it is easy to express it in a 'pseudo-linear' regression structure. Introduce a parameter vector $\Theta$ and a regression vector $\varphi_{NL}(t)$ defined as

$$\Theta = [\theta^T,\ \omega_{11} \ldots \omega_{rM},\ c_1 \ldots c_l]^T \tag{2.12}$$

$$\varphi_{NL}(t) = [\varphi^T(t),\ \varphi^T(t) \otimes \varphi^T_{\mathcal{N}_f}(t),\ e(t-1) \ldots e(t-1)]^T \tag{2.13}$$

where $\varphi^T_{\mathcal{N}_f}(t) = [\mathcal{N}_f(p_j, \varphi(t)),\ j = 1, ..., M]$, and the symbol $\otimes$ denotes Kronecker production. Then we have the *third expression* of the hybrid quasi-ARMAX model given by

$$\mathcal{M}: \quad y(t) = \varphi^T_{NL}(t)\Theta + e(t) \tag{2.14}$$

The expression (2.14) shows that the proposed model is linear in the parameters, so that it is simple for parameter estimation.

From the expressions (2.10), (2.11) and (2.14), we can conclude that the hybrid quasi-ARMAX model is equipped with a linear ARMAX structure, flexibility and simplicity [3]. These features make the proposed hybrid model very practicable.

[3] It will be discussed in Chapter 6 about how to construct the hybrid quasi-ARMAX model to be linear in the one-step past input variable, which is typical useful for control design.

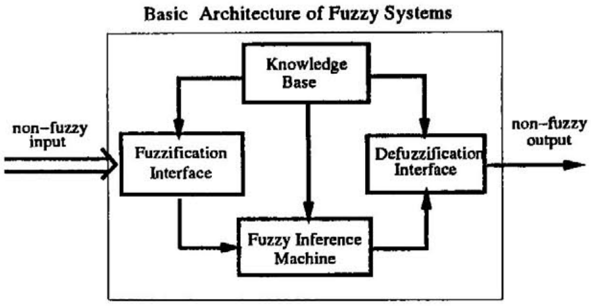


Figure 2.3: Basic configuration of MISO fuzzy system

### (4) Associative memory networks

Form a network's point of view, the hybrid quasi-ARMAX model can also be seen as an *associative memory networks*, which consists of two hidden layers: the first layer (next to the input layer) with weights determined by a set of simplified NNMs; the second layer with weights simply taking the time delayed value of the system input and output, see Fig. 2.2. We could expect that such a specially structured associative memory network is more suitable for control design or system analysis than a general neural network.

Finally, if the regression vector $\varphi(t)$ is considered to be a general one as that of linear black-box model, referred to Appendix A, the hybrid quasi-ARMAX modeling scheme can be generalized as *hybrid quasi-linear black-box modeling* scheme.

## 2.2.2 Fuzzy Inference Based Multi-Modeling

If we inspect the hybrid quasi-ARMAX model in the expression (2.10) from a viewpoint of linear approximation, we may find that in the hybrid quasi-ARMAX modeling, a nonlinear system is represented using local linear ARMAX models for each operating point. Furthermore, based on the expression (2.11), we know that it is equivalent to a hybrid model combining a linear ARMAX model and a multi-ARX-model consisting of several local linear ARX models with employing an interpolation using the 'basis functions'. In this subsection, we will derive the multi-ARX-model from the viewpoint of an adaptive fuzzy modeling. It is shown that if the AFSs are used as the NNMs, the multi-ARX-model part is actually equivalent to a Sugeno-Takagi fuzzy system [99]. This result leads to a fuzzy inference based multi-modeling scheme for identification of nonlinear systems.

### (1) Sugeno-Takagi Fuzzy System

Generally, as shown in Fig. 2.3, the basic configuration of a fuzzy system includes four principal elements: *fuzzification interface*, *knowledge base* (fuzzy rule base and data base), *fuzzy inference machine*, and *defuzzification interface* [64]. Here we use a MISO fuzzy system: $U \subset R^r \rightarrow W \subset R$, where $U$ and $W$ are compact.

The *fuzzification interface* is a mapping from the observed non-fuzzy input space $U \subset R^r$ to fuzzy sets defined in $U$, where a fuzzy set defined in $U$ is characterized by a membership function $\mu_{A_i^j} : U \rightarrow [0,\ 1]$, and is labeled by a linguistic term $A_i^j$ such as "small", "medium", "large", "very large", etc.

The *knowledge base* contains rule base and data base which discretizes the universes of discourse and describes the membership functions $\mu_{A_i^j}$, $\mu_{B^j}$. The $j$-th linguistic rule has the form:

$$\begin{array}{ll} j\text{th}: & \text{If } x_1 \text{ is } A_1^j \text{ and, ... , } x_r \text{ is } A_r^j \text{ then } y \textit{ is } B^j \\ & \qquad j = 1, \ldots, M \end{array} \tag{2.15}$$

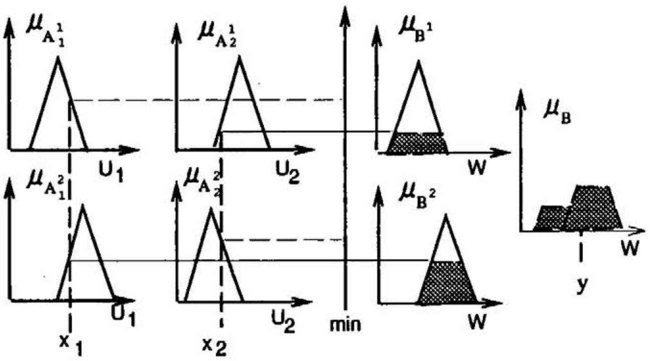


Figure 2.4: Diagrammatic representation of fuzzy reasoning

For a Sugeno-Takagi fuzzy system, the consequent ($y$ is $B^j$) is represented as a function of the process state variables $(x_1, ..., x_r)$, i.e.,

$$y = f_j(x_1, ..., x_r) \tag{2.16}$$

The *fuzzy inference machine* is decision making logic which employs fuzzy rules from the knowledge base to determine fuzzy outputs of a fuzzy system corresponding to its fuzzified inputs. There are different kinds of fuzzy inference machines [65], here we use Mamdani's minimum operation rule.

In the fuzzification interface, the crisp inputs may be treated as fuzzy singletons. Then the firing strengths $\alpha_j$ of the $j$th rule (2.15) may be expressed as

$$\alpha_j = \bigwedge_{i=1}^{r} \mu_{A_i^j}(x_i) \tag{2.17}$$

where $\wedge$ is the minimum operator. Based on Mamdani's minimum operation rule, the output of the $j$th rule defined as $\mu_{\bar{B}^j}$ is given by

$$\mu_{\bar{B}j}(y) = \alpha_j \wedge \mu_{Bj}(y) \tag{2.18}$$

which implies that the membership function $\mu_B$ of the inferred consequence $B$ is pointwise given by

$$\mu_B(y) = \bigcup_{j=1}^{M} \mu_{\bar{B}j}(y) = \bigcup_{j=1}^{M} (\alpha_j \wedge \mu_{Bj}(y)) \tag{2.19}$$

Figure 2.4 shows the fuzzy reasoning diagrammatically.

The *defuzzification interface* defuzzifies the fuzzy outputs of the fuzzy inference machine and generates a non-fuzzy output which is the actual output of the whole fuzzy system. There are a lot of defuzzification strategies [65], here we use centroid defuzzification method which is the most commonly-used method.

$$y_{C.G.} = \frac{\int y \; \mu_B(y)dy}{\int \mu_B(y)dy} \tag{2.20}$$

In the case of Sugeno-Takagi fuzzy system, $\mu_B(y)$ can be described as

$$\mu_B(y) = \begin{cases} \alpha_j & \text{if } y = f_j(x_1, ..., x_r) \quad j = 1, ..., M \\ 0 & \text{otherwise} \end{cases} \tag{2.21}$$

Using (2.21) in (2.20), we can then express the defuzzified output explicitly

$$y_{C.G.} = \frac{\sum_{j=1}^{M} f_j(x_1, ..., x_r) \left( \wedge_{i=1}^{r} \mu_{A_i^j}(x_i) \right)}{\sum_{j=1}^{M} \left( \wedge_{i=1}^{r} \mu_{A_i^j}(x_i) \right)}. \tag{2.22}$$

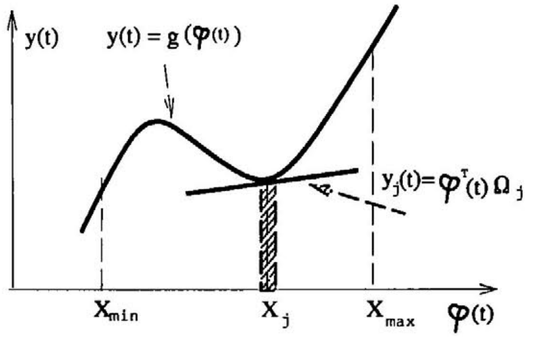


Figure 2.5: A linear approximation of the system in the region around $\varphi(t) = X_j$.

### (2) Fuzzy Inference Based Multi-Model

Consider a general nonlinear system described by (2.1), whose operating region is assumed to be mostly located in $X_{\min} \leq \varphi(t) \leq X_{\max}$. Based on multi-modeling approach, for a region near $X_j$ ($X_{\min} \leq X_j \leq X_{\max}$), the system can be approximated by using a linear black-box model

$$y_j(t) = \varphi^T(t)\Omega_j \tag{2.23}$$

where $\Omega_j = [a_{1j} \ldots a_{nj} \; b_{1j} \ldots b_{mj}]^T$ is unknown parameter vector. If the region is chosen to be so small that the approximation can be achieved with an arbitrary accuracy, see Fig. 2.5.

Now introduce a Sugeno-Takagi fuzzy system with the following fuzzy rule base to represent the systems

$$\begin{array}{ll} j\text{th}: & \text{If } \varphi(t) \text{ is } A^j \text{ then } y(t) = \varphi^T(t)\Omega_j \\ & \qquad j = 1, \ldots, M \end{array} \tag{2.24}$$

where $A^j = [A_1^j \ldots A_r^j]$ ($r = n + m$) is a vector with linguistic elements corresponding to $X_j$, and $M$ is the number of rules. Obviously, if $M$ is chosen large enough, the fuzzy system can approximate the sufficiently smooth continue nonlinear system with an arbitrary accuracy. Corresponding to the rule base (2.24), the fuzzy inference based multi-model can be expressed as

$$y(t) = \frac{\sum_{j=1}^{M} \varphi^T(t)\Omega_j \left(\wedge_{i=1}^{r} \mu_{A_i^j}(x_i(t))\right)}{\sum_{j=1}^{M} \left(\wedge_{i=1}^{r} \mu_{A_i^j}(x_i(t))\right)} + v(t) \tag{2.25}$$

where $x_i(t)$'s are the elements of regression vector $\varphi(t)$. The fuzzy sets in the input space $A_i^j$ (hence their membership functions $\mu_{A_i^j}$, $j = 1, \ldots, M$; $i = 1, \ldots, n+m$) will be determined by using knowledge information about system structure. For example, if we only know that the system operating region is mostly located in $X_{min} \leq \varphi(t) \leq X_{max}$, the possible fuzzy sets may be chosen to be something shown in Fig. 2.6. If we introduce the notation

$$\mathcal{N}_f(p_j, \varphi(t)) = \frac{\wedge_{i=1}^{r} \mu_{A_i^j}(x_i(t))}{\sum_{k=1}^{M} \left(\wedge_{i=1}^{r} \mu_{A_i^k}(x_i(t))\right)} \tag{2.26}$$

(2.25) can be further expressed as

$$y(t) = \sum_{j=1}^{M} \underbrace{\varphi^T(t)\Omega_j}_{\text{ARX}} \mathcal{N}_f(p_j, \varphi(t)) + v(t), \tag{2.27}$$

which is the same as the multi-ARX-model part in (2.11).

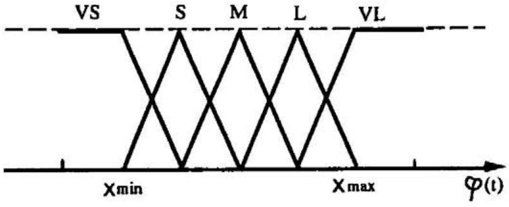


Figure 2.6: The fuzzy sets in the input space

Moreover, if a MA noise model is employed for system disturbance $v(t)$, we have

$$y(t) = \sum_{j=1}^{M} \underbrace{\varphi^T(t)\Omega_j}_{\text{ARX}} \mathcal{N}_f(p_j, \varphi(t)) + \sum_{i=1}^{l} c_i e(t-1) + e(t) \tag{2.28}$$

where $e(t)$ is white noise. The (2.28) can be further expressed in a 'pseudo-linear' regression form

$$y(t) = \sum_{j=1}^{M} \underbrace{\varphi_e^T(t)\Omega_{ej}}_{\text{ARMAX}} \mathcal{N}_f(p_j, \varphi(t)) + e(t) \tag{2.29}$$

where $\varphi_e^T(t)$ and $\theta_{ej}$ denote the extended regression vector and parameter vector defined as

$$\begin{aligned} \varphi_e(t) &= [y(t-1) \ ... \ y(t-n) \ u(t-1) \ ... \ u(t-m) \ e(t-1) \ ... \ e(t-l)]^T \\ \Omega_{ej} &= [a_{1j} \ ... \ a_{nj} \ b_{1j} \ ... \ b_{mj} \ c_1 \ ... \ c_l]^T. \end{aligned} \tag{2.30}$$

We will call the model described by (2.29) a fuzzy inference based model, which can be used to identify various nonlinear systems. A distinctive feature of the fuzzy inference based model is that it is typically easy to perform the identification in a multiresolution way. Furthermore, fuzzy systems are known to have good flexibility for describing various nonlinear systems. Therefore, it implies that the hybrid quasi-ARMAX model is flexible enough to identify various nonlinear systems in practice.

### 2.2.3 Determining $p_j$ Using Knowledge Information

As mentioned earlier, in order to achieve the model simplicity and the model flexibility simultaneously, in the hybrid black-box modeling, one first increases the overall flexibility of model by using a group of NNMs and then restricts the flexibility in the higher order nonlinearity band for the model simplicity. The latter is done by determining the scale and position parameter vectors $p_j$ of the 'basis functions' in the NNMs using knowledge information. Since in a black-box modeling, the physical insight of system is assumed to be not available, the knowledge information are mainly obtained from the observed data and the prediction error during the estimation. Several kinds of knowledge information can be considered to be useful. They are:

- the information concerning the operating region of $\varphi(t)$. This information can be obtained easily from the observed data.
- the information about the structure of nonlinearity. This may sometime be obtained by trying various linear black-box models to identify the system.
- the information concerning the relations among the elements in $\varphi(t)$. This is always known when $\varphi(t)$ is chosen.
- the information about the size of prediction errors and their relations with the operating region of $\varphi(t)$. This may be obtained during the estimation.

However, how to obtain and how to use those kinds of information are still under investigation. Here only some suggestions can be given.

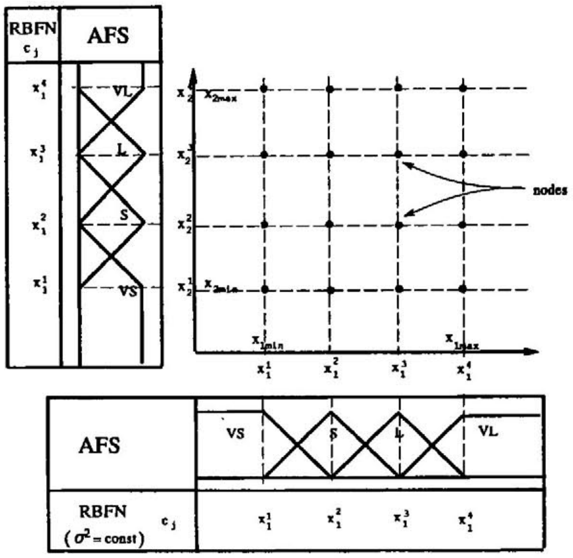


Figure 2.7: An example for determining $\mathbf{p}_j$ for AFS and RBFN.

### (1) A Strategy for Determining $\mathbf{p}_j$

How to determine the parameter vector $p_j$ for general NNMs is still an open problem to be solved, which depends on the kind of NNMs used. The following strategy is suitable for adaptive fuzzy systems (AFS), radial basis function networks (RBFN) and B-spline based models.

Suppose the NNM has $r$ inputs, $X = [x_i,\ i = 1, ..., r]$ and the operating region is mostly located in $X_{\min} \le X \le X_{\max}$, $X_{\min} = [x_{i\ \min},\ i = 1, ..., r]$, $X_{\max} = [x_{i\ \max},\ i = 1, ..., r]$. $X \notin [X_{\min},\ X_{\max}]$ is allowable in practice. We first put *nodes* into the input hyperplane. As shown in Fig. 2.7, if the number of nodes corresponding to $x_i$ is denoted as $n_i$, the total number of the nodes in the hyperplane will be $M = \prod_{i=1}^{r} n_i$. Next, the parameter vectors $p_j$ are chosen so that the 'basis functions' $\mathcal{N}_f(p_j, X)$ have appropriate shape and are put onto each node. Without using other knowledge information, the nodes will be uniformly assigned in the hyperplane. Figure 2.7 shows an example for determining $p_j$ for AFS and RBFN with $r = 2$ and $M = 4 \times 4$. It should however be noticed that this strategy is suitable for AFS, RBFN and B-spline based model, but it may not be suitable for other NNMs such as neural networks and wavelet networks. Further research is needed for using neural networks and wavelet networks as NNMs in the hybrid quasi-ARMAX model.

### (2) Several Hints for Reducing the Total Number of Nodes

The prior knowledge concerning the operating region $[X_{\min},\ X_{\max}]$ is the least information required for determining the scale and position parameter vectors $p_j$. However, when $\dim(X)$ is large, the total number of nodes ($M$) may be rather large. Therefore, further information should be used to reduce the number of nodes or to improve the node assignment. The following hints can be considered to reduce the total number of nodes $M$:

- Hint A: If the system is linear with respect to $x_i$, $n_i$ may be chosen to be 1.
- Hint B: If no other useful information, $n_1$ and $n_{n+1}$ corresponding to $y(t-1)$ and $u(t-1)$ are assigned with appropriate values, while all other $n_i$'s are set to 1, since $y(t-1)$ and $u(t-1)$ include the information about other variables.
- Hint C: If the role of nodes can be replaced by employing interpolation of NNMs, those nodes may be removed from the hyperplane.

It should be pointed out here that user's experience will play an important role on using these simple hints efficiently. In order to make the modeling scheme less heuristically dependent, further investigations are needed to develop an algorithm for incorporating knowledge information automatically.

## 2.3 Estimation of the Hybrid Models

In the previous section, we discussed the hybrid quasi-linear black-box modeling scheme. Now let us have a summary on what steps are needed in order to find a hybrid quasi-linear black-box model.

(1) **Select the regression vector $\varphi(t)$.** This is equivalent to determining the order $n$ and $m$. Since the hybrid quasi-linear black-box model is basically an extension of linear black-box model to the nonlinear cases, $\varphi(t)$ will be determined based on the results of identifying the system using a linear black-box model. Therefore, many existing approaches for determining the order of linear models such as Akaike criteria AIC and FPE can be applied. However, we do not limit ourselves only to use the 'optimal' order [4], the values $n$ and $m$ for $\varphi(t)$ should be chosen as small as possible, so far as the performance of the linear model is not significantly worse.

(2) **Select a scalar 'mother basis function' $\mathcal{N}_f$.** Theoretically, all the NNMs which can be described by (2.7) can be used. However, the parameters $p_j$ specifying the nature of the 'basis function' will be determined using knowledge information, so some of them (e.g. AFS, RBFN and B-spline based model) are more feasible, while some others (e.g. NN and WN) are less feasible. Based on author's experience and simulation results, the adaptive fuzzy systems (AFS) seem to be preferred.

(3) **Determine the parameter vectors $p_j$.** For general NNMs, this is still an open problem to be solved, which depends on the kind of NNMs used. Section 2.2.3 describes a strategy for the cases where AFS, RBFN or B-spline based model are used.

(4) **Determine the order $M$.** This is related with determining the parameter vectors $p_j$. Section 2.2.3 gives several hints to reduce the order $M$. When the AFSs are used as the NNMs, the order $M$ denotes the number of rules in the AFS. Therefore, this is a problem similar to building an adaptive fuzzy system using knowledge information and observed data. Many existing results for fuzzy system design can be applied. An alternative perspective of determining $M$ is *Intelligent Identification*. We will discuss the possibility later.

(5) **Estimate model parameters $\Theta$.** The $\Theta$ denotes the unknown parameters to be estimated, which includes the ARMAX parameters ($a_i$, $i = 1, ..., n$; $b_i$, $i = 1, ..., m$; $c_i$, $i = 1, ..., l$) and the coordinate parameters of the NNMs ($\omega_{ij}$, $i = 1, ..., r$, $j = 1, ..., M$). We will discuss the estimation problem in the rest of this section.

### 2.3.1 Model Estimation and Model Properties

The followings are some basic and general features that affect the model properties.

**(1) Models and Model Estimation**

Consider our general hybrid quasi-linear black-box model

$$y(t) = g(\varphi(t), \Theta) + e(t) \tag{2.31}$$

where

$$g(\varphi(t), \Theta) = \sum_{i=1}^{n} \bar{a}_{i,t} y(t-i) + \sum_{i=1}^{m} \bar{b}_{i,t} u(t-i) + \sum_{i=1}^{l} c_i e(t-i) \tag{2.32}$$

[4] Strictly speaking, since there exists error of nonlinear undermodeling, the optimal order might not exist. In such cases, the Akaike criteria are only used as indicator of the error of nonlinear undermodeling.

$$\begin{array}{l} \bar{a}_{i,t} = a_i + f_i(\varphi(t)) \quad (i = 1, ..., n) \\ \bar{b}_{j,t} = b_j + f_{j+n}(\varphi(t)) \quad (j = 1, ..., m) \end{array} \tag{2.33}$$

$$f_i(\varphi(t)) = \sum_{j=1}^{M} \omega_{ij} \mathcal{N}_f\left(\boldsymbol{p}_j, \varphi(t)\right) \tag{2.34}$$

$$\varphi(t) = [y(t-1) \ ... \ y(t-n) \ u(t-1) \ ... \ u(t-m)]^T \tag{2.35}$$

$$\Theta = [a_1 \ ... \ a_n \ b_1 \ ... \ b_m \ \omega_{11} \ ... \ \omega_{rM} \ c_1 \ ... \ c_l]^T \tag{2.36}$$

Assume that we are given a finite set $Z_e^N$ of measured regressor-output part:

$$Z_e^N = \{(y(t), \varphi(t)), \ t = 1, ..., N\} \tag{2.37}$$

We refer to $Z_e^N$ as the *estimation data set*, since the model parameter estimation will rely on it.

Now, a leading guideline for estimating $\Theta$ will be to minimize the error between the output of the model and the measured output using $Z_e^N$

$$V_N(\Theta, Z_e^N) = \frac{1}{N} \sum_{t=1}^{N} \|y(t) - g(\varphi(t), \Theta)\|^2 \tag{2.38}$$

$$\hat{\Theta}_N = \arg\min_{\Theta} \{V_N(\Theta, Z_e^N)\} \tag{2.39}$$

which can be performed using existing well-known identification algorithms.

### (2) Model Quality

Suppose that the actual data can be described by

$$y(t) = g_0(\varphi(t)) + e(t) \tag{2.40}$$

where $g_0$ is some unknown 'true model' and $e(t)$ is white noise. Then for the estimate of $\Theta$ based on $Z_e^N$, $\hat{\Theta}_N$, we want $g_0(\varphi(t))$ and $g(\varphi(t), \hat{\Theta}_N)$ to be 'close'.

Measure of model quality

There are many possible measures for model quality. We here use the following one:

$$\bar{V}(\Theta) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \|g_0(\varphi(t)) - g(\varphi(t), \Theta)\|^2 \tag{2.41}$$

Based on this measure, the root mean square (RMS) error is introduced as

$$\text{RMS error} = \sqrt{\frac{1}{N} \sum_{t=1}^{N} \|g_0(\varphi(t)) - g(\varphi(t), \hat{\Theta}_N)\|^2} \tag{2.42}$$

Bias error and variance error

Under reasonable conditions [69, 25]

$$\hat{\Theta}_N \to \Theta^* \tag{2.43}$$

where

$$\Theta^* = \arg\min_{\Theta} \{\bar{V}(\Theta)\} \tag{2.44}$$

With this definition of $\Theta^*$, we can decompose the total error into tow part: *bias error* and *variance error*

$$\begin{aligned} E\bar{V}(\hat{\Theta}_N) &= E\|g_0(\varphi(t)) - g(\varphi(t), \hat{\Theta}_N)\|^2 \\ &= \underbrace{E\|g_0(\varphi(t)) - g(\varphi(t), \Theta^*)\|^2}_{\text{bias error}} + \underbrace{E\|g(\varphi(t), \Theta^*) - g(\varphi(t), \hat{\Theta}_N)\|^2}_{\text{variance error}}. \end{aligned} \tag{2.45}$$

In order to obtain a 'good' model, one should try to make both bias error and variance error small.

### (3) Model Flexibility and Model Parameters

Since as the number of data $N$ tends to infinity, the variance error vanishes, a model structure should be flexible enough in order to have bias error small. For such purpose, a nonlinear black-box usually offers a large number of parameters. However, the estimation of $\Theta$ is realized using finite data set in practice, so that there inevitably exists variance error which is proportional to the number of 'active' or 'used' parameters [94, 92]

$$E\|g(\varphi(t),\Theta^*) - g(\varphi(t),\hat{\Theta}_N)\|^2 \sim \sigma^2 \frac{p_e}{N} \tag{2.46}$$

where $\sigma^2 = Ee^2(t)$ and $p_e$ is the number of 'active' or 'used' parameters. Therefore, an attention should be paid to making the number of 'active' or 'used' parameter $p_e <<$ (far smaller than) the number of 'offered' parameters dim($\Theta$).

Regularization

In the case (e.g. neural network) where the model parameters do not have physical interpretations, the effort has to be made in the estimation algorithm. The 'regularization' is one of the common and useful techniques [91]. In order to distinguish between more and less 'important' parameters, a penalty term is added to the criterion (2.38):

$$W_N(\Theta, Z_e^N) = V_N(\Theta_N, Z_e^N) + \delta\|\Theta\|^2 \tag{2.47}$$

where $\delta$ is a small number. Then the estimation of $\Theta_N$ is done by minimizing (2.47) instead of (2.38). Intuitively, the idea is that a parameter that does not influence the first term of (2.47) very much will be kept close to zero by the second term, so that only the 'important' parameters among the offered are used. However, the bias error may increase by introducing extra term $\delta\|\Theta\|^2$.

Parameter localization

In the hybrid quasi-linear black-box modeling, the model structure flexibility is realized by employing a group of NNMs, which of course brings a large number of parameters into the model. However, all those parameters which are globally active will be determined using knowledge information, so that the parameters needed to be estimated are only locally active. For example, in the case where the 'basis functions' are chosen to be convex and compact, the parameter vector $\Omega_j = [\omega_{1j} \dots w_{n+m,j}]$ of $\Theta$ in (2.36) is only active in the operating region around $j$th node. We call this technique, which lets the parameters to be estimated be only locally active, a *parameter localization*. In such a case, for each operating point, the number of active parameters will be small, so that $p_e << \dim(\Theta)$ holds.

## 2.3.2 Estimation Algorithm

The estimation of $\Theta$ based on (2.39) is a well established problem, see e.g. [70, 94]. Especially, since the hybrid quasi-linear black-box model is linear in the parameters (LIP), it can be expressed in a 'pseudo-linear' regression structure (2.14). The existing estimation techniques can be applied.

For clarity, we recall (2.14)

$$y(t) = \varphi_{NL}^T(t)\Theta + e(t) \tag{2.48}$$

where $\varphi_{NL}^T(t)$ is a regression vector consisting of both linear and nonlinear elements, given by

$$\varphi_{NL}^T(t) = \left[\varphi^T(t),\ \varphi^T(t) \otimes \varphi_{\mathcal{N}_f}^T(t),\ e(t-1), \dots, e(t-l)\right]^T \tag{2.49}$$

and where

$$\varphi_{\mathcal{N}_f}^T(t) = [\mathcal{N}_f(p_j, \varphi(t)),\ j = 1, \dots, M] \tag{2.50}$$

### (1) Least Squares Method

If (2.48) does not include the noise model (e.g. $l = 0$), the $\Theta$ can be estimated efficiently and analytically by solving the normal equations. For an estimation data set $Z_e^N$, the optimal parameter estimate is

$$\hat{\Theta}_N = \left[\sum_{t=1}^{N} \varphi_{NL}(t)\varphi_{NL}^T(t)\right]^{-1} \sum_{t=1}^{N} \varphi_{NL}(t)y(t) = R_N^{-1} \sum_{t=1}^{N} \varphi_{NL}(t)y(t) \tag{2.51}$$

provided that the inverse of the $d \times d$ ($d = \dim\Theta$) regression matrix $R_N$ exists. For numerical reasons this inverse is rarely formed, but instead the estimate is computed via approaches such as singular value decomposition (SVD), which is able to handle rank deficient regression matrices. The estimation algorithm (2.51) can easily be transformed into a recursive form. However, instead doing this we shall derive recursive algorithm for the case where (2.48) includes MA noise model.

### (2) Recursive Estimation Algorithm

If a noise model such as MA noise model is included, or the hybrid quasi-linear black-box models are employed for on-line applications such as adaptive control of nonlinear systems, the estimation should be performed in a recursive way. Based on the well developed recursive identification theory [70, 94], we will first introduce the prediction error method (PEM) generally, then apply it to the hybrid quasi-linear black-box models.

Recursive PEM algorithm

Consider the problem of estimation $\Theta$ by minimizing a criterion based on prediction error

$$\hat{\Theta}(N) = \arg\min_{\Theta} \{V_N(\Theta)\} \tag{2.52}$$

$$V_N(\Theta) = \frac{1}{2}\sum_{t=1}^{N} \varepsilon^2(t,\Theta) \tag{2.53}$$

$$\varepsilon(t,\Theta) \triangleq y(t) - \hat{y}(t|\Theta). \tag{2.54}$$

A recursive algorithm for this problem can be given as follows, see Appendix C for the details of the derivation

$$\hat{\Theta}(t) = \hat{\Theta}(t-1) + L(t)\varepsilon(t,\hat{\Theta}(t-1)) \tag{2.55}$$

$$L(t) = \frac{P(t-1)\psi(t,\hat{\Theta}(t-1))}{1 + \psi^T(t,\hat{\Theta}(t-1))P(t-1)\psi(t,\hat{\Theta}(t-1))} \tag{2.56}$$

$$P(t) = P(t-1) - \frac{P(t-1)\psi(t,\hat{\Theta}(t-1))\psi^T(t,\hat{\Theta}(t-1))P(t-1)}{1 + \psi^T(t,\hat{\Theta}(t-1))P(t-1)\psi(t,\hat{\Theta}(t-1))} \tag{2.57}$$

where $\psi(t,\hat{\Theta}(t))$ is the *negative* derivative of $\varepsilon(t,\Theta)$ with respect to $\Theta$.

In order to apply the recursive algorithm (2.55)-(2.57) to the hybrid quasi-linear black-box models, we should discuss how to determine the $\varepsilon(t,\hat{\Theta}(t-1))$ and $\psi(t,\hat{\Theta}(t-1))$ for those models. The following two methods can be considered.

Extended Least Squares (ELS) Method

Let us write (2.48) as

$$e(t) = y(t) - \varphi_{NL}^T(t)\Theta \tag{2.58}$$

If we have a sequence of estimates $\hat{\Theta}(t)$ available, it seems natural to estimate $e(t)$ by $\bar{\varepsilon}(t)$, computed according to

$$\bar{\varepsilon}(t) = y(t) - \varphi_{NL}^T(t)\hat{\Theta}(t) \tag{2.59}$$

In this way, the $\varepsilon(t,\hat{\Theta}(t-1))$ and $\psi(t,\hat{\Theta}(t-1))$ can be determined as

$$\varepsilon(t,\hat{\Theta}(t-1)) = y(t) - \varphi_{NL}^T(t)\hat{\Theta}(t-1) \tag{2.60}$$

$$\psi^T(t,\hat{\Theta}(t-1)) = -\left.\frac{\partial\varepsilon(t,\Theta)}{\partial\Theta}\right|_{\Theta=\hat{\Theta}(t-1)} = \varphi_{NL}^T(t) \tag{2.61}$$

where $\varphi_{NL}(t)$ is described by (2.49) whose elements $e(t-i)$ will be replaced by $\varepsilon(t-i,\hat{\Theta}(t-i-1))$.

**Prediction Error Method (PEM)**

According to the contents of $\psi_{NL}(t)$ in (2.49), (2.48) can be decomposed as

$$y(t) = \varphi^T(t)\theta + \varphi^T(t)\otimes\varphi_{\mathcal{N}_f}^T(t)\Theta_f + C(q^{-1},\Theta)e(t) \tag{2.62}$$

where

$$\Theta = [\theta^T,\ \Theta_f^T,\ c_1\ ...\ c_l]^T \tag{2.63}$$

$$C(q^{-1},\Theta) = 1 + c_1q^{-1} + ... + c_lq^{-l} \tag{2.64}$$

and $\Theta_f$ is the parameters associated with the $\varphi(t)\otimes\varphi_{\mathcal{N}_f}(t)$ part of $\varphi_{NL}(t)$, defined as

$$\Theta_f = [\omega_{11}\ ...\ \omega_{rM}]^T. \tag{2.65}$$

Then for (2.62), the prediction of $y(t)$ can be computed recursively

$$\hat{y}(t|\Theta) = \frac{1}{C(q^{-1},\Theta)}\left[\varphi^T(t)\theta + \varphi^T(t)\otimes\varphi_{\mathcal{N}_f}^T(t)\Theta_f + (C(q^{-1},\Theta)-1)y(t)\right] \tag{2.66}$$

using data up to $t-1$ with given $\Theta$, in which the initial conditions can be taken to zero. We can also evaluate the *prediction error*

$$\varepsilon(t,\Theta) \triangleq y(t) - \hat{y}(t|\Theta) \tag{2.67}$$

according to the model parameter $\Theta$.

With (2.67), we can write (2.66) as

$$C(q^{-1},\Theta)\varepsilon(t,\Theta) = y(t) - \varphi^T(t)\theta - \varphi^T(t)\otimes\varphi_{\mathcal{N}_f}^T(t)\Theta_f \tag{2.68}$$

$\partial\varepsilon(t,\Theta)/\partial\Theta$ can thus be computed as

$$C(q^{-1},\Theta)\frac{\partial\varepsilon(t,\Theta)}{\partial\theta} = -\varphi^T(t)\theta \tag{2.69}$$

$$C(q^{-1},\Theta)\frac{\partial\varepsilon(t,\Theta)}{\partial\Theta_f} = -\varphi^T(t)\theta + \varphi^T(t)\otimes\varphi_{\mathcal{N}_f}^T(t) \tag{2.70}$$

$$C(q^{-1},\Theta)\frac{\partial\varepsilon(t,\Theta)}{\partial c_i} = -\varepsilon(t-i,\Theta). \tag{2.71}$$

It therefore follows that $\varepsilon(t,\hat{\Theta}(t-1))$ and $\psi(t,\hat{\Theta}(t-1))$ are determined as

$$\varepsilon(t,\hat{\Theta}(t-1)) = y(t) - \varphi_{NL}^T(t|\hat{\Theta}(t-1))\hat{\Theta}(t-1) \tag{2.72}$$

$$\psi(t,\hat{\Theta}(t-1)) = \frac{1}{C(q^{-1},\hat{\Theta}(t-1))}\varphi_{NL}^T(t|\hat{\Theta}(t-1)) \tag{2.73}$$

where $\varphi_{NL}(t|\hat{\Theta}(t-1))$ is $\varphi_{NL}(t)$ whose elements $e(t-i)$ are replaced by $\varepsilon(t-i,\hat{\Theta}(t-1))$.

### (3) Local Minimum and Over-fitting

Local minimum and over-fitting are two difficult problems associated with the identification of nonlinear black-box models. The reason is that a nonlinear black-box model usually offers a large number of parameters and is often nonlinear in those parameters. Using 'regularization' technique, several authors state that the over-fitting problem can be solved [91]. But it seems that there is so far no easy solution for local minimum problem [92], since a hybrid identification method using genetic algorithm does not seem to be very efficient when the number of parameters is large.

For the hybrid quasi-linear black-box models, local minimum and over-fitting do not seem to be problems. Since the models are linear in the parameters (LIP) to be estimated, local minimum problem appears only when a noise model is employed for the system disturbance. We will show that this problem can be solved using our proposed hybrid identification algorithm [38], see also Chapter 3. On the other hand, since the parameters in the hybrid quasi-linear black-box are only locally active, the over-fitting does not appear to be a big problem. In particular, when the identification is done via a multi-resolution approach, i.e., the identification is started with fewer parameters, while the number of parameter increases only when higher resolution (modeling accuracy) is required, the over-fitting problem vanishes.

## 2.3.3 Implementation of the Estimation Algorithm

In contrast to the case of linear system identification, when a recursive algorithm is implemented to identify nonlinear systems, the following two problems arise: (1) how to improve the convergence property; (2) how to estimate the separate noise model. We will show that these problems can be solved in the identification of the hybrid quasi-linear black-box models.

### (1) Two-Step Estimation of Hybrid Quasi-Linear Black-Box Models

When the estimation is done using the optimization-based algorithm (2.55)-(2.57), a better initial value will improve the convergence property very much. This is particularly true when the number of parameters to be estimated is large and larger than that may be really needed. In the hybrid quasi-ARMAX model, the number of the parameter is $\dim\{\Theta\} = n + m + l + (n + m) \times M$, which is much larger than that of a linear model. Moreover, $\theta$ in the $\Theta$ is obviously not really needed in the sense of describing the system. Therefore, we will implement the estimation in the following two steps.

Step 1: Estimation of linear approximation

Assume the parameters associated with the nonlinear terms of coefficients in the hybrid quasi-ARMAX model to be zero, i.e., $\Theta_f = [\omega_{vj},\ v = 1, ..., r;\ j = 1, ..., M]^T = 0$ in (2.5). Then we have

$$y(t) = \sum_{i=1}^{n} a_i y(t-i) + \sum_{i=1}^{n} b_i u(t-i) + \sum_{i=1}^{n} c_i e(t-i) + e(t) \tag{2.74}$$

Indeed, (2.74) is nothing but a linear ARMAX model. Hence the estimate of the linear approximation can be used as an initial value for the estimation of the hybrid quasi-linear black-box model. The linear estimate can be considered as a nice initial value because of the following two reasons: (1) most processes in practice can be approximated with a linear model in a reasonable accuracy; (2) the constant parameters $a_i$ and $b_i$ are not independent of the parameters $\omega_{vj}$, i.e., the roles of the parameters $a_i$ and $b_i$ can be replaced by the parameters $\omega_{vj}$ (we call $a_i$ and $b_i$ as *redundant parameters*).

It is well known that the estimation of (2.74) can be realized using a recursive PEM algorithm [70, 94]. However, it should be noticed that the criterion function in this estimation is not always unimodal because of the noise model, so that there is a risk that the optimization-based algorithm is stuck at a local minimum. Moreover, the unmodeled dynamics resulted from the linear approximation will increase the risk [58]. On the other hand, it has been found experimentally that the parameters of noise model should be estimated in this linear approximation and be fixed in the next stage where the parameters of system model are estimated, because a noise model can not be well identified if its parameters are estimated together with a large number of system model parameters. Therefore, a reliable identification algorithm is crucially needed to improve the convergence in the estimation of linear approximation. In [38], we have proposed an hybrid identification algorithm based on an effective combination of genetic algorithm and an optimization-based method. Using that algorithm, $a_i$, $b_i$, and $c_i$ can be estimated reliably. We will discuss this problem further in Chapter 3.

Step 2: Estimation with using the initial value

With the estimates of linear approximation ($\hat{a}_i$, $\hat{b}_i$, $\hat{c}_i$), we may now take $a_i = \hat{a}_i$, $b_i = \hat{b}_i$, $c_i = \hat{c}_i$ and $\omega_{vj} = 0$ as initial values, and estimate $a_i$, $b_i$, $c_i$ and $\omega_{vj}$ using the algorithm (2.55)-(2.57). However, it is found experimentally that fixing $c_i = \hat{c}_i$ in this step gives better identified noise model.

On the other hand, in some applications, a nonlinear system is preferred to be approximated using a linear model, and the error due to nonlinear undermodeling is treated as unmodeled dynamics or is modeled using a nonlinear compensator. One of the motivations for such identification is that many existing results can be applied to linear models, but rare for nonlinear models. For such applications, we just fix $a_i = \hat{a}_i$, $b_i = \hat{b}_i$, $c_i = \hat{c}_i$, and only estimate $\omega_{vj}$ using (2.55)-(2.57). Since in the hybrid quasi-ARMAX model, $a_i$ and $b_i$ are redundant parameters, i.e., they are not independent of $\omega_{vj}$, the estimation with $a_i = \hat{a}_i$, $b_i = \hat{b}_i$, $c_i = \hat{c}_i$ fixed can be approximately achieved at the global minimum of the criterion function. The estimation results will thus give an identified linear ARMAX model ($\hat{a}_i$, $\hat{b}_i$ and $\hat{c}_i$) for the best linear approximation of the system and the estimate of modeling error due to nonlinear undermodeling described by $\Delta a_{i,t}(\hat{\omega}_{vj})$ and $\Delta b_{i,t}(\hat{\omega}_{vj})$.

### (2) Estimation of the Noise Model

Generally, it is harder to obtain good noise models than system models, so that it is important to restrict the flexibility of the noise model. With assuming the model to be linear to the residuals, we simply add linear noise terms to our quasi-linear black-box models. For example, the hybrid quasi-ARMAX model can be viewed as a hybrid quasi-ARX model with a linear MA noise term added to it. Such noise models are also considered in connection to nonlinear black-box models [89]. However, in [89], J. Sjöberg concluded based on his experiment results that the separate noise models did not improve the fit substantially, hence it was more important to model the nonlinearities than to model the noise dynamics. But from our experiment results, we do not agree to his conclusion. The problem arisen here is *how to estimate the noise terms in a nonlinear model with a large number of parameters to be estimated.* We found that it is difficult, if not impossible, to estimate the noise model and the system model by minimizing the same criterion function. In this section, a two-step approach is thus suggested for such estimation, in which the noise model is first identified by approximating the system with a corresponding linear model, then the estimate of the noise model is fixed when the system model is identified.

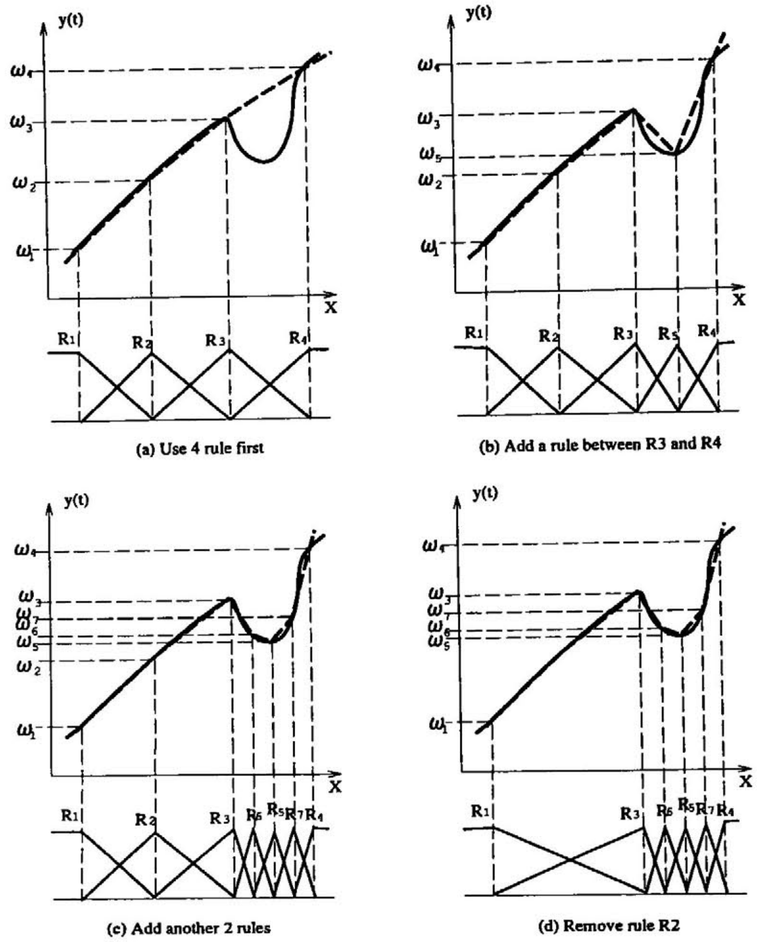
## 2.4 Intelligent Identification of the Hybrid Models

We believe that the hybrid quasi-linear black-box modeling to be distinctive not only in that it is equipped with a linear structure, flexibility and simplicity, but also in that it is able to incorporate various knowledge information during the modeling and the parameter estimation. In Section 2.2.3, we describe a strategy for determining the scale and position parameter vectors $p_j$ using knowledge information. In this section, we will discuss the possibility using knowledge information to optimize the model structure during the parameter estimation.

As mentioned earlier, in the quasi-linear black-box model, the parameters offered by the NNMs are locally active, and the overall performance of the model is obtained by employing interpolation. This localization and interpolation makes the following intelligent identification possible:

(1) If modeling error is large in the operating region near the $j$th node, the node density near the $j$th node should be increased, which can be done on-line.

(2) The identification can be realized using a multiresolution approach, in which few nodes are chosen at the beginning, the new nodes will then be added when higher resolution (modeling



(a) Use 4 rule first

(b) Add a rule between R3 and R4

(c) Add another 2 rules

(d) Remove rule R2

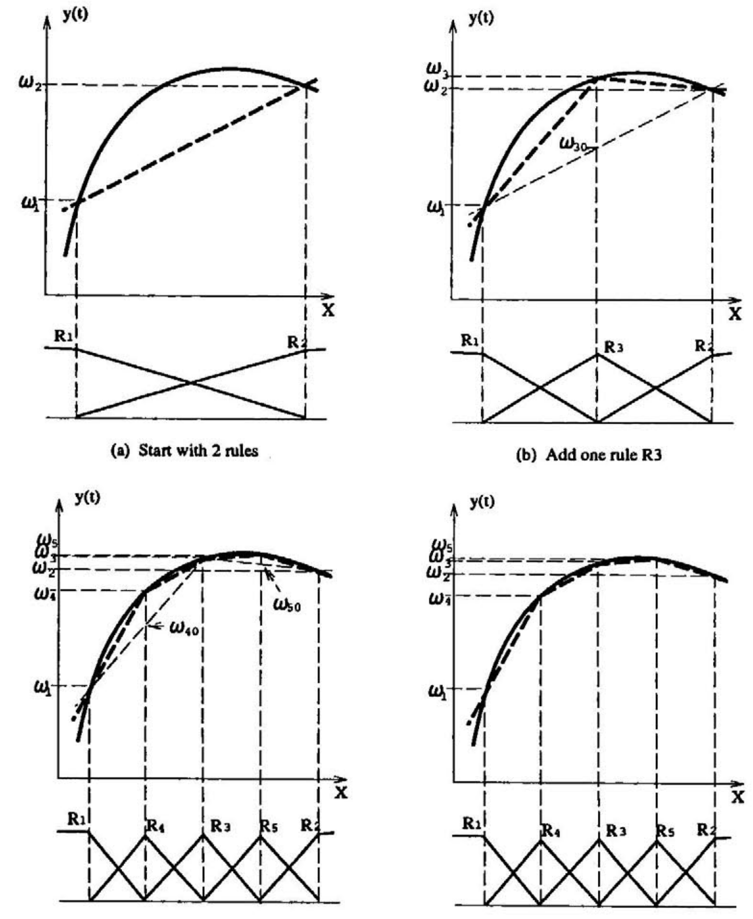
Figure 2.8: Diagrammatic representation of optimizing a fuzzy system

accuracy) is required. The parameters associated with the latter added nodes can be assigned with better initial values using the knowledge obtained during the estimation.

### 2.4.1 Self-Optimization of Model Structure

When only the information concerning the operating region $[X_{\min}, X_{\max}]$ is used, the nodes will be uniformly assigned in the hyperplane. A preferable way to optimize the model structure is to reduce the total number of the nodes automatically during the parameter estimation. Two ways may be considered: (1) start the estimation algorithm with a low node density, and then add nodes only to where they are necessitated; (2) start the estimation algorithm with a rather high node density, and then remove the nodes which are not necessitated. If the contribution of a node can be obtained from the interpolation of its neighborhood, the node can be removed. We know that when adaptive fuzzy systems are used as the NNMs, the nodes corresponds to fuzzy rules. Therefore, the existing results for optimizing fuzzy systems using knowledge information and observed data can be applied to the above optimization.

However, how to implement those results in the estimation algorithm is a problem to be solved. We here discuss a simple case to show how a fuzzy system is optimized. Consider a nonlinear system whose input-output relation is shown in Fig. 2.8(a) with solid line. A fuzzy system with 4 rules is first using to represent the system. Based on the interpolation property of a fuzzy system, we can



(a) Start with 2 rules

(b) Add one rule R3

(c) Add another two rules R4 and R5

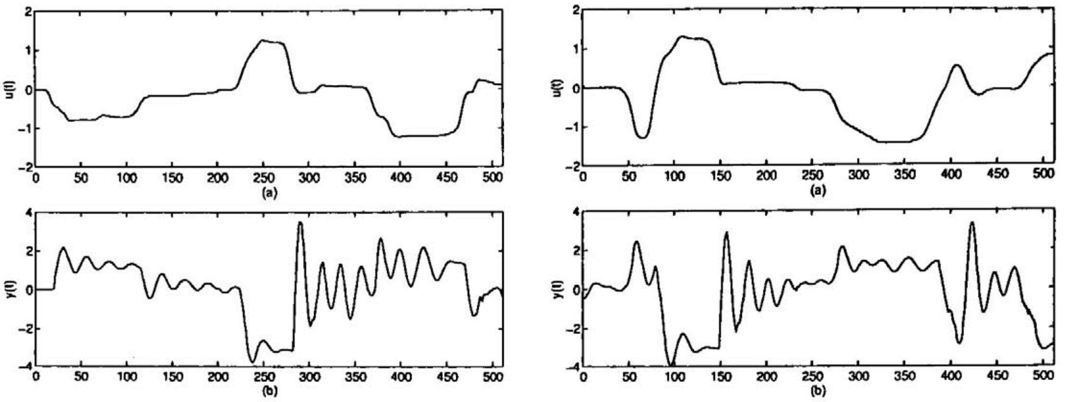
(d) Final result

Figure 2.9: An example of estimating an adaptive fuzzy system in a multiresolution way

give its output shown with dashed line. We may find that after estimation the modeling error in the operating region between $R_3$ and $R_4$ is large. Then we add a rule $R_5$ between the $R_3$ and $R_4$. The performance has been improved, see Fig. 2.8(b). However, it seems that higher modeling accuracy is needed in the operating region near the $R_5$, so that another two rule $R_6$ and $R_7$ are added, Fig. 2.8(c). On the other hand, the contribution of the $R_2$ seems can be obtained from the interpolation of $R_1$ and $R_3$, so that it can be removed. In this way, we can build an optimized fuzzy system with 6 rules, see Fig. 2.8(d).

### 2.4.2 Identification via Multiresolution Approach

Crucial points for estimating a model with large number of parameters are to improve its convergence property, to prevent being stuck at a local minimum, and to avoid being over-fitted. Since the quasi-linear black-box model is linear in the parameters, the local minimum does not seem to be a problem. It is well known that a better initial value can improve the convergence property very much. We have shown that the initial values of the parameters associated with the linear part of the model can be given by using the result of the linear approximation. We here will further show that the parameters associated with the nonlinear terms can be estimated in a multiresolution way, i.e., the estimation starts with few nodes for the NNMs, and the node density increases during the estimation. The initial values of the parameters associated with the newly added nodes can be obtained from the current



Estimation data Validation data

Figure 2.10: Real data: (a) the input $u(t)$, (b) the output $y(t)$.

estimates of the parameters associated with the neighbor nodes. In this way, all parameters associated with later added notes will have better initial values, the later the more close to their 'true' values.

Figure 2.9 shows an example of estimating an adaptive fuzzy system in a multiresolution way. The estimation starts with the adaptive fuzzy system containing two parameters to be adapted, corresponding two rules. After estimating the two parameters for a while, rule $R_3$ is added, which give the parameter $\omega_3$ to be estimated. Using the interpolation property, the initial value for $\omega_3$ can be given as $\omega_{30}$ (Fig. 2.9(b)) which is calculated from $\hat{\omega}_1$ and $\hat{\omega}_2$. Then after estimation, the rules $R_4$ and $R_5$ are added. The initial values for $\omega_4$ and $\omega_5$ can be given as $\omega_{40}$ and $\omega_{50}$ which can be determined using $\hat{\omega}_1$, $\hat{\omega}_2$ and $\hat{\omega}_3$ (Fig. 2.9(c)). Finally, we have the result shown in Fig. 2.9(d). Obviously, the estimation carried out in such multiresolution way is expected to have better convergence property and the over-fitting problem will vanish.

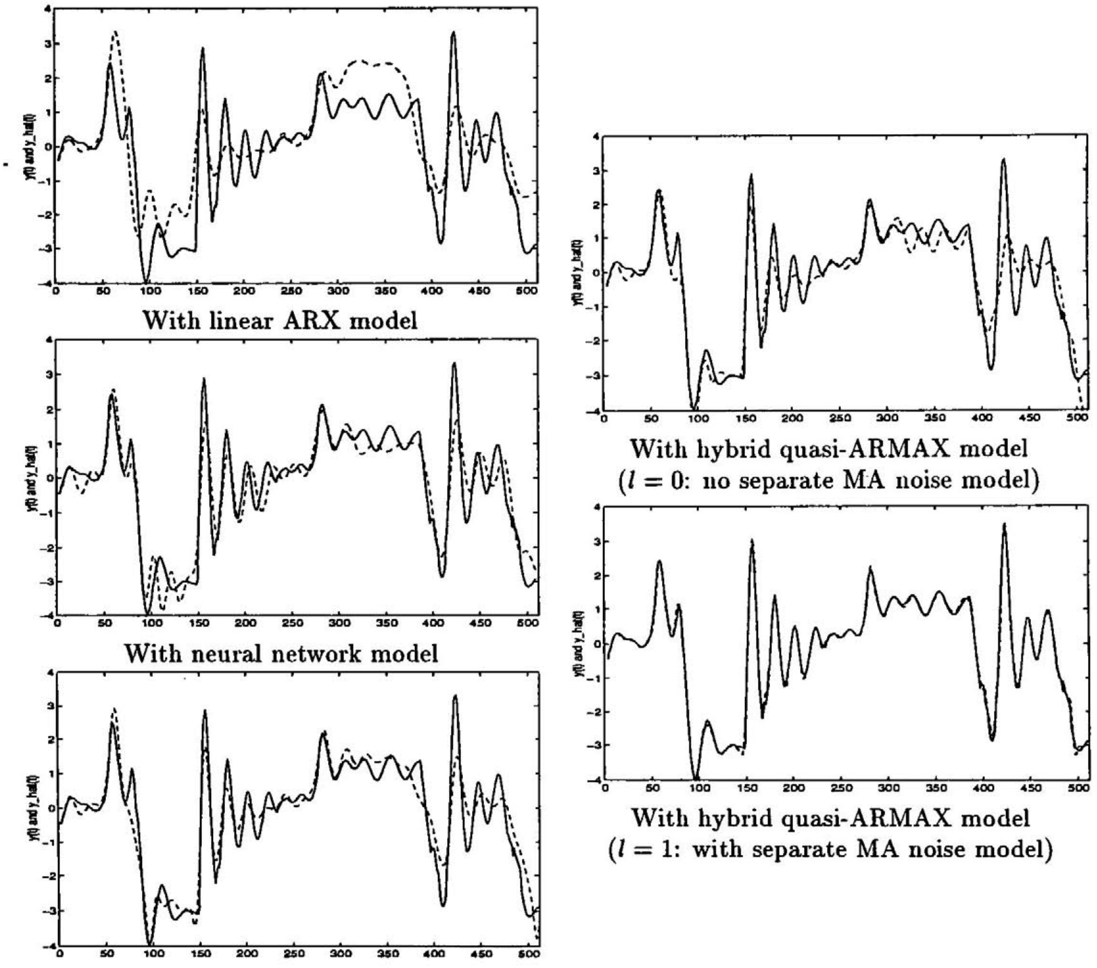
It has been well-known in the literature that wavelet transform is very suitable for multiresolution analysis. How to use the wavelet transform as the NNMs in the quasi-linear black-box model seems to be an interesting topic for further research.

## 2.5 Experimental Studies

In this section, we will apply the hybrid quasi-linear black-box model to identify a variety of real systems and simulated systems. We have argued that the proposed hybrid quasi-linear black-box has linear structure, flexibility and simplicity. The model simplicity is clearly shown in its estimation algorithm. The usefulness of the linear structure will be shown in Chapter 5 and 6 by applying it to fault detection and control design of nonlinear systems. We here will show its flexibility and its generalization ability by applying it to identify a variety of nonlinear systems and comparing the results with those using Neural Networks and Wavelet Networks which are known to have good flexibility. For such purposes, some of the systems chosen in this section are well known in the literature, where they have been used to test nonlinear black box models such as Neural Networks, Wavelet Network and Hinging Hyperplanes. We can easily compare our results with those using nonlinear black-box models.

### 2.5.1 Modeling A Hydraulic Robot Actuator

The position of a robot arm is controlled by a hydraulic actuator. The oil pressure in the actuator is controlled by the size of the valve opening through which the oil flows into the actuator. The position of the robot arm is then a function of the oil pressure. Let us denote by u(t) and y(t) the position of the valve and the oil pressure at time $t$, respectively. A sample of 1024 pairs of $\{y(t), u(t)\}$ was



With linear ARX model

With neural network model

With wavelet network model

With hybrid quasi-ARMAX model
($l = 0$: no separate MA noise model)

With hybrid quasi-ARMAX model
($l = 1$: with separate MA noise model)

Figure 2.11: Simulation of identified models on validation data. The solid line shows the true oil pressure and the dashed line the simulated model output.

registered[5]. We divide it into two equal parts for estimating and for validating our model, respectively. The estimation data are depicted in Fig. 2.10 (left). and the validation data in Fig.2.10 (right).

### (1) Using a linear model

Following the principle of 'try simple things first', we try to use a linear ARX model to identify the system. A reasonable modeling result has obtained with $n = 3$ and $m = 2$, that is, the regression vector $\varphi(t) = [y(t-1)\ y(t-2)\ y(t-3)\ u(t-1)\ u(t-2)]^T$. Figure 2.11 (left-upper) shows the result of a simulation with the obtained linear model on validation data, which gives a root mean square (RMS) error of 1.0160. The result is not very impressive.

### (2) Using neural network and wavelet network models

The problem of modeling the hydraulic robot actuator has been discussed comprehensively by Sweden and France groups. J. Sjöberg modeled the system using network network models [90, 89], Q. Zhang

[5] The data were taken from public ftp domain. We gratefully acknowledge Linköping University for providing the data.

using wavelet network models [7], and P. Pucar using hinging hyperplane models [86].

For the sake of easy comparison, similar simulations have been carried out using Matlab Neural Network toolbox and the package of Zhang (1993) [116]. First, a NARX model based on an one-hidden-layer sigmoid neural network with 10 hidden units, 5 input units and one output unit is considered. This gives a model with 71 parameters. The identified neural network NARX model is simulated on the validation data. The result is shown in Fig. 2.11 (left-middle), which gives a RMS error of 0.617. Our result is a little worse than the one obtained by J. Sjöberg, see Fig. 9 in [92], where the RMS error is 0.467. The reason we think is that we have over-trained the model. J. Sjöberg et al. have developed an algorithm using regularization to solve the over-training problem [91]. Then, another NARX model based on a wavelet network [117] is considered to model the hydraulic actuator in a similar way, with the same regressors. The wavelet function used is $\psi(\varphi) = (d - \varphi^T\varphi)e^{-\varphi^T\varphi/2}$, with $d = \dim\varphi$. Since this identification is realized using the package of Zhang, we obtained the same result as the one obtained by Q. Zhang, see Fig. 10 in [92]. The result of the identified wavelet network model simulated on the validation data is shown in Fig. 2.11 (left-lower), which gives a RMS error of 0.5285.

In [92], Sjoberg et al. reported that they have also identified the hydraulic robot actuator using several other nonlinear black-box models with various nonlinear structures. The best model they obtained simulated on the validation data gives a RMS error of 0.328, referred to [92] for details.

### (3) Using hybrid quasi-ARMAX models

Now we will use the hybrid quasi-ARMAX model described by (2.5) to identify the hydraulic robot actuator. For easy comparison, we do not consider the noise model first, i.e., $n = 3$, $m = 2$, $l = 0$ are chosen for the regression vector. From the estimation data shown in Fig. 2.10, we choose $X_{\min}$=[-4 -4 -4 -2 -2] and $X_{\max}$=[4 4 4 2 2]. Since no other useful information available, we choose $n_1 = n_4 = 4$, $n_2 = n_3 = n_5 = 1$ using Hint B, which gives $M = 16$. The model obtained thus has 85 parameters to be estimated. After estimating the model using the estimation data for 2048 steps, the simulation of the model on the validation data is shown in Fig. 2.11 (right-upper), which gives a RMS error of 0.5445. Comparing this result with those using neural networks and wavelet networks, we can see that the hybrid quasi-ARMAX model has obtained a compatible result.

In [89], J. Sjoberg reported that for this example, the separate noise models did not improve the fit substantially. Since a noise model has to be identified from the residual system model, it is difficult to identify it if the system model is nonlinear in the parameters to be estimated. As mentioned earlier, the hybrid quasi-ARMAX model is linear in the parameters and includes a MA noise model naturally, so that the noise model can be identified easily. Next, we use a hybrid quasi-ARMAX model with $n = 3$, $m = 2$, $l = 1$ and $M = 16$ to identify the system in a similar way. Figure 2.11 (right-lower) shows the simulation of the model on the validation data. The RMS error is 0.1360, which is better than the best results in [89]. We think that the main reason for the superior performance of the proposed model is that it has better property in dealing with correlated noise.

## 2.5.2 Modeling A Boiler Plant

As shown in Fig. 2.12, the water level in the boiler drum (Drum level) is controlled by the water flows into the drum, and the pressure (Drum pressure) is controlled by the heater which is controlled by the fuel flows into the heater. The stream output (Stream flow) is kept as constant. Therefore, this is a two-input-two-output system, in which the drum pressure $y_1(t)$ and the drum level $y_2(t)$ are a function of the fuel flow $u_1(t)$ and the water flow $u_2(t)$, while the fluctuation of the stream flow can be treated as system disturbance.

### (1) The data

450 pairs of $\{u_1(t),\ u_2(t),\ y_1(t),\ y_2(t)\}$ were registered as estimation data, and 290 pairs as validation data. For the estimation data, the system was excited by adding pseudo-random binary sequences (PRBS) with amplitudes [−5%, +5%] to the $u_1(t)$, $u_2(t)$. The estimation data are depicted in Fig. 2.13.

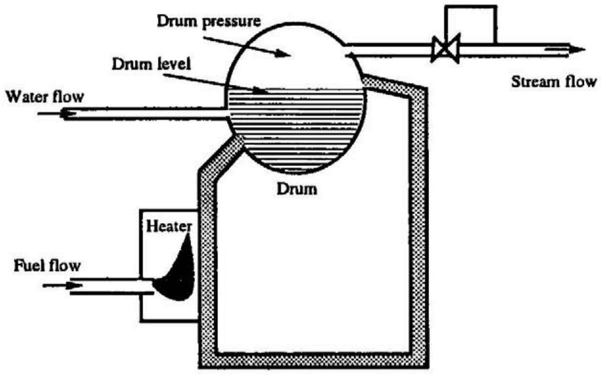


Figure 2.12: A boiler plant.

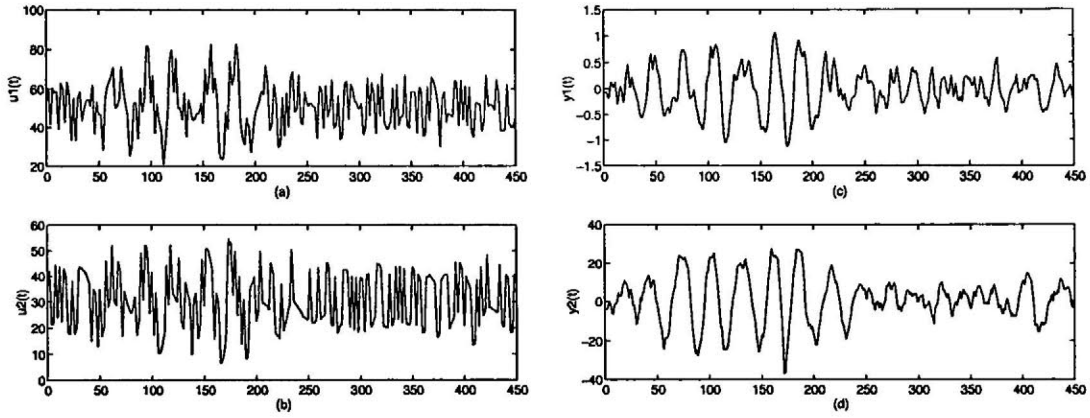


Figure 2.13: Estimation data: the inputs $u_1(t)$, $u_2(t)$ and the output $y_1(t)$, $y_2(t)$.

### (2) Using linear ARMAX models

Let us divide the two-input-two-output system into two two-input-one-output systems. Then use two ARMAX models to identify the systems. The two-input-one-output ARMAX model is described as

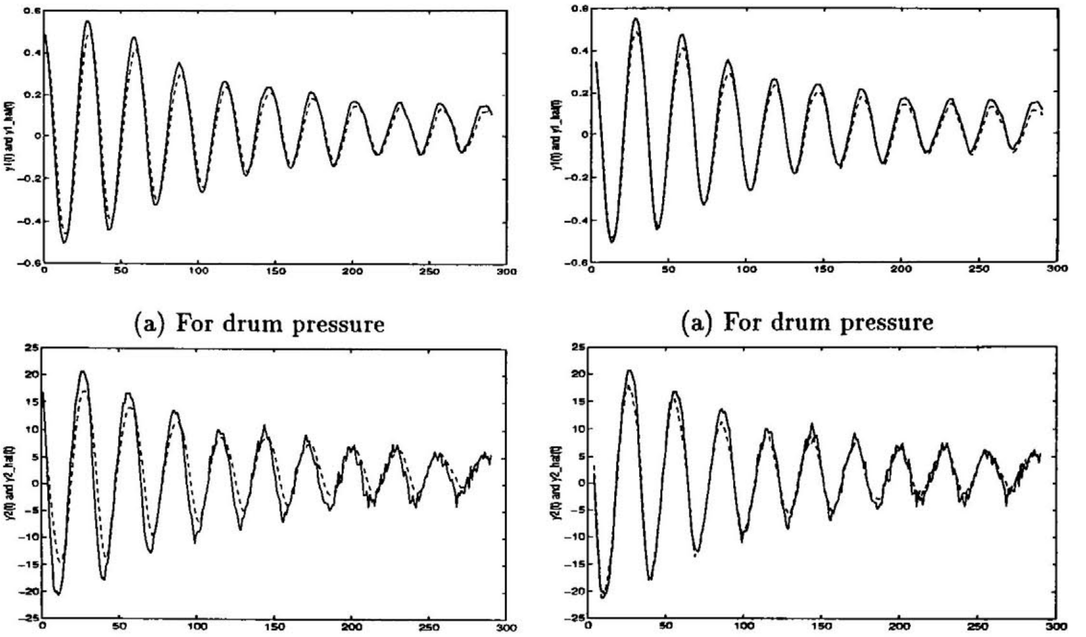
$$\begin{aligned}(1+a_1q^{-1}+...+a_nq^{-n})y(t) &= (b_{11}q^{-1}+...+b_{1m_1}q^{-m_1})u_1(t)\\ &+ (b_{21}q^{-1}+...+b_{2m_2}q^{-m_2})u_2(t)\\ &+ (1+c_1q^{-1}+...+c_lq^{-l})e(t) \qquad (2.75)\end{aligned}$$

Various ARMAX models have been tried. Reasonable results have obtained by using ARMAX models with $n=m_1=m_2=2$, $l=1$ for the drum pressure $y_1(t)$ part, and $n=3$, $m_1=m_2=2$, $l=1$ for the drum level $y_2(t)$ part. The results of the identified models simulated on the validation are shown in Fig. 2.14 (left). The RMS errors for the drum pressure is 0.0480 and for the drum level 3.2528.

### (3) Using hybrid quasi-ARMAX models

Now we use the hybrid quasi-ARMAX models to identify the systems. We first modify the SISO hybrid quasi-ARMAX model described in (2.5) to a two-input-one-output one

$$(1+\bar{a}_{1,t}q^{-1}+...+\bar{a}_{n,t}q^{-n})y(t) \doteq (\bar{b}_{11,t}q^{-1}+...+\bar{b}_{1m_1,t}q^{-m_1})u_1(t)$$



(a) For drum pressure

(a) For drum pressure

(b) For drum level
With linear ARMAX models

(b) For drum level
With hybrid quasi-ARMAX models

Figure 2.14: Simulation of the identified models on validation data. The solid line shows the true measurements and the dashed line the simulated model outputs.

$$
\begin{aligned}
&+ \quad (\bar{b}_{21,t}q^{-1} + ... + \bar{b}_{2m_2,t}q^{-m_2})u_2(t) \\
&+ \quad (1 + c_1q^{-1} + ... + c_lq^{-l})e(t)
\end{aligned}
\qquad (2.76)
$$

where

$$
\left\{
\begin{array}{ll}
\bar{a}_{i,t} = a_i + f_{i,t} & i = 1, ..., n \\
\bar{b}_{1j,t} = b_{1j} + f_{j+n,t} & j = 1, ..., m_1 \\
\bar{b}_{2k,t} = b_{2k} + f_{k+n+m_1,t} & k = 1, ..., m_2
\end{array}
\right.
\qquad (2.77)
$$

where $f_{i,t}$ $(i = 1, ..., n + m_1 + m_2)$ are realized using the NNMs.

Based on the results of linear approximation, the hybrid quasi-ARMAX models with $n = m_1 = m_2 = 2$, $l = 1$ is chosen for the drum pressure $y_1(t)$ part, with $n = 3$, $m_1 = m_2 = 2$, $l = 1$ for the drum level $y_2(t)$ part. For determining $p_j$, $X_{\min}$=[-1.1 -1.1 20 20 6 6] and $X_{\max}$=[1.1 1.1 85 85 55 55] are chosen for the case of modeling the drum pressure, while $X_{\min}$=[-30 -30 -30 20 20 6 6] and $X_{\max}$=[30 30 30 85 85 55 55] for modeling the drum level. Since no other useful information available, using Hint B $n_1 = n_3 = n_5 = 3$, $n_2 = n_4 = n_6 = 1$ and $n_1 = n_4 = n_6 = 3$, $n_2 = n_3 = n_5 = n_7 = 1$ are chosen for the two cases, respectively. Next, using Hint C to remove some nodes, we obtain $M = 18$ for both cases. After performing the estimation for 4500 steps, the simulations of the identified models on the validation data are shown in Fig. 2.14 (right), which give the RMS errors 0.0332 for the drum pressure and 1.6072 for the drum level. We can see that the model performances have been improved by using the hybrid quasi-ARMAX models.

## 2.5.3 Modeling A De-NOx Device

As shown in Fig. 2.15, in the de-NOx device, $NH_3$ is used to de-oxidize the NOx, so that the output of NOx to the open air can be reduced. Let us denote by $u_1(t)$, $u_2(t)$ and $y(t)$ the NOx in, the $NH_3$ flow and the NOx out at time $t$, respectively. A sample of 2000 pairs of $\{u_1(t),\ u_2(t),\ y(t)\}$ was registered as estimation data, and 4500 pairs as validation data. The estimation data are depicted in Fig. 2.16

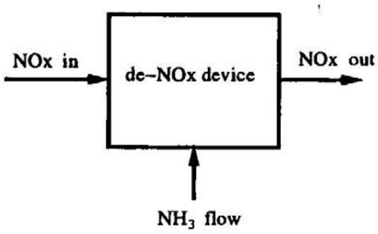


Figure 2.15: A de-NOx device.

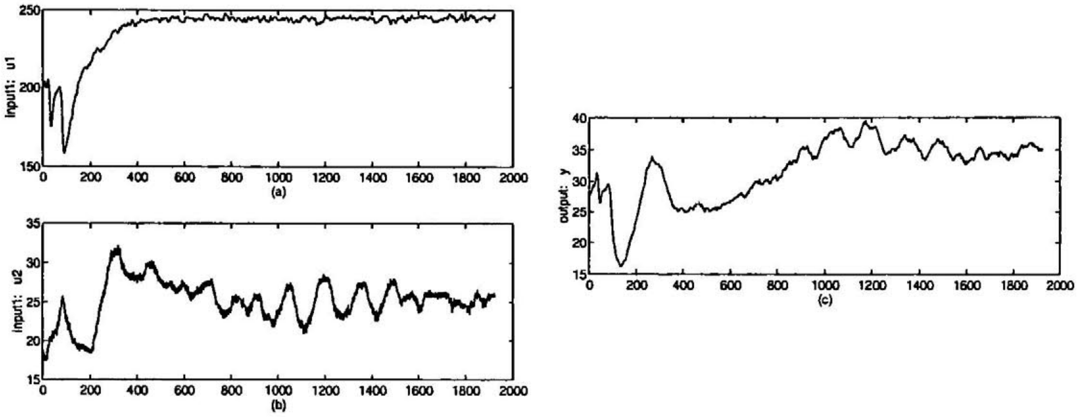


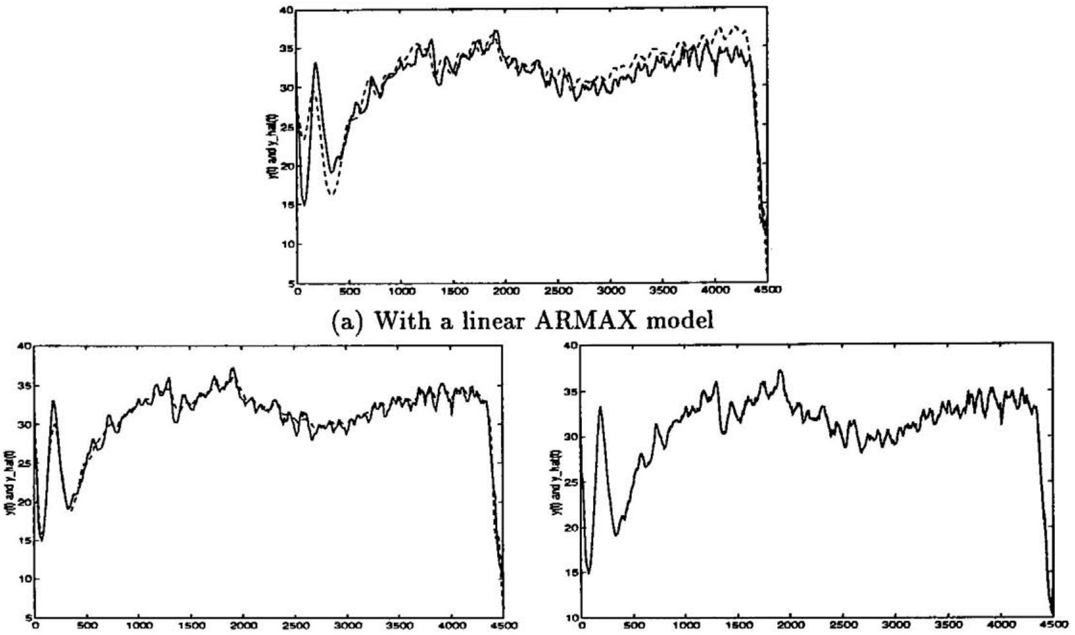
Figure 2.16: Estimation data: the inputs $u_1(t)$, $u_2(t)$ and the output $y(t)$.

We first use a two-input-one-output linear ARMAX model with $n = m_1 = m_2 = 2$, $l = 1$ and the two-input-one-output hybrid quasi-ARMAX model (2.76) with $n = m_1 = m_2 = 2$, $l = 1$, $M = 16$ to identify the system, respectively. For the hybrid quasi-ARMAX model, we choose $X_{\min}$=[16 16 150 150 17 17] and $X_{\max}$=[40 40 250 250 33 33] based on the estimation data shown in Fig. 2.16. Since changing $m_1$ in the identifications using linear ARMAX models did not affect the modeling accuracy, we may consider that the system is linear with respect to $u_1(t-i)$. Then based on Hints A and B, $n_1 = n_5 = 4$, $n_2 = n_3 = n_4 = n_6 = 1$ is chosen, which gives $M = 16$. After estimating the models for 4000 steps, the simulations of the identified models on the validation data are shown in Fig. 2.17(a) and (b), in which the RMS errors for the linear ARMAX model is 2.0232 and for the hybrid quasi-ARMAX model 1.0289. We can see that the hybrid quasi-ARMAX model has better performance, but do not improve the fit substantially. Next we increase the orders of the linear ARMAX model with $n = 5$, $m_1 = 2$, $m_2 = 5$, $l = 1$. The result is shown in Fig. 2.17(c), which give a RMS error of 0.2464. The result shows that the de-NOx device does not contain nonlinearity. However, the results obtained this modeling shows that the NNMs employed in the hybrid quasi-ARMAX model can represent not only nonlinear behaviors of dynamic systems but also the high order linear behaviors.

### 2.5.4 Modeling A Mathematical System

We have modeled three real processes successfully. However in practice, it is very difficult to excite the nonlinearity of a real process, so that the above three real processes do not seem to have strong nonlinearity. Here, we borrow a mathematical system from [76], which contains rather high order nonlinearity. The system is governed by

$$y(t) = f[y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)] \tag{2.78}$$



(a) With a linear ARMAX model

(b) With a hybrid quasi-ARMAX model

(c) With a high order linear ARMAX model

Figure 2.17: Simulation of the identified models on validation data. The solid line shows the true measurements and the dashed line the simulated model outputs.

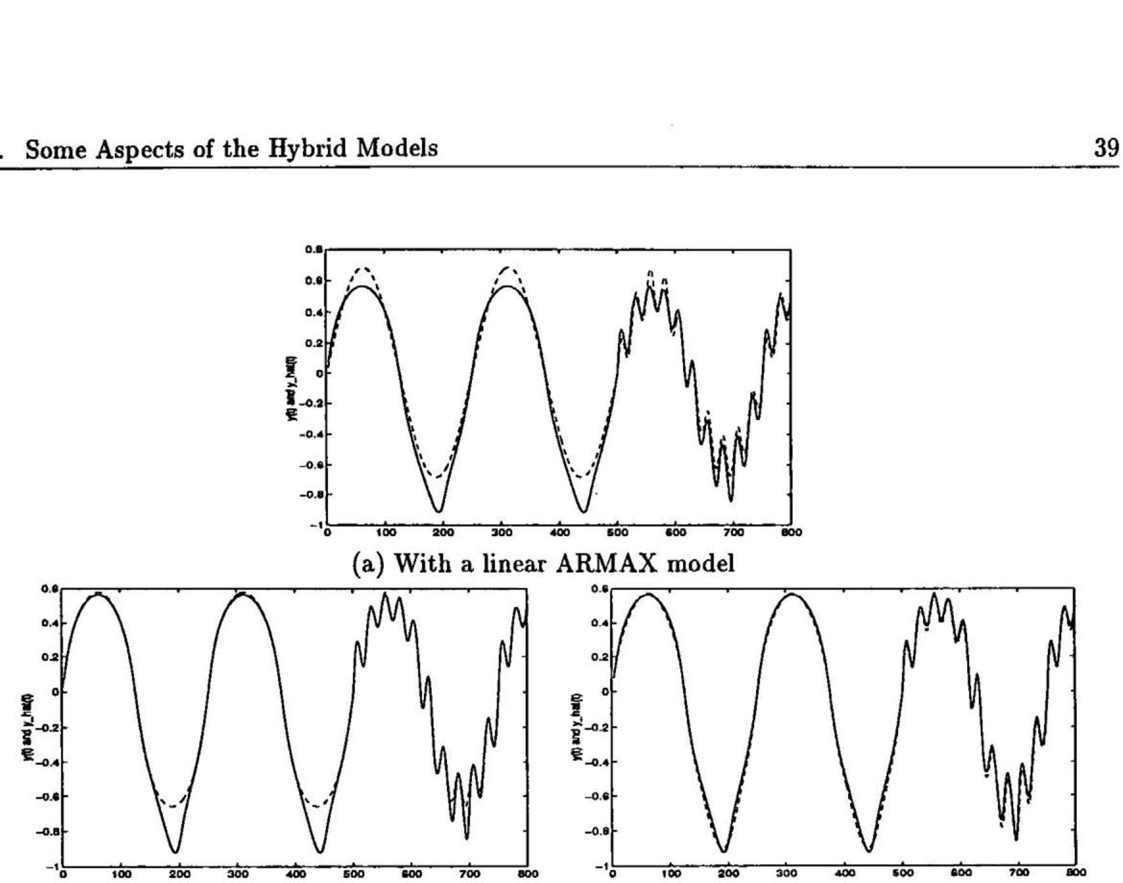
where

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2}$$

Estimation data are sampled when system is excited using random input uniformly distributed in the interval [-1, 1], while validation data are sampled from system using an input $u(t) = \sin(2\pi t/250)$ for $t \leq 500$ and $u(t) = 0.8\sin(2\pi t/250) + 0.2\sin(2\pi t/25)$ for $t > 500$.

First, an ARMAX model with $n = 3$, $m = 2$, $l = 0$ is used to identify the system. The simulation of ARX model on the validation data is shown in Fig. 2.18(a), which gives a RMS error of 0.0866. The result is not very impressive, which is not surprising since the system contains high order nonlinearity. Next, a four-layer neural network of $\mathcal{N}_{5,20,10,1}$, which contains 341 parameters to be estimated, is used to identify the system. Similar to K.S. Narendra in [76], we trained the model for 100,000 steps using Matlab Neural Network toolbox. The simulation of the model on the validation data is shown in Fig. 2.18(b). The RMS error is 0.0678, which is compatible to the result shown in [76], but it is not very impressive too. The reason we think is that the identification algorithm has been stuck at a local minimum. Finally, we use the hybrid quasi-ARMAX model with $n = 3$, $m = 2$, $l = 0$ to identify the system. $X_{\min}$=[-1 -1 -1 -1 -1] and $X_{\max}$=[1 1 1 1 1] are chosen based on the information obtained from the estimation data. Since the system is linear with respect to $u(t - i)$, we choose $n_1 = n_2 = n_3 = 3$, $n_4 = n_5 = 1$ using Hint A. $M = 18$ is then obtained by removing some nodes using Hint C. We thus obtain a model with 95 parameters to be estimated. After estimating the model for 5000 steps, the simulation of the model on the validation data is shown in Fig. 2.18(c). The RMS error is as small as 0.0270. We can see that the hybrid quasi-ARMAX model represents the system very well. We think that the reason for the better performance of the proposed model is that our estimator could find the global minimum.

Now we use the fuzzy inference based multi-model (2.27) to identify the system. Such identification is equivalent to identifying the hybrid quasi-ARMAX model discussed above with $a_i = 0$ and $b_i = 0$ fixed. The identification has been done using the multi-models with 1 rule, 3 rules, 9 rules and 18 rules



(a) With a linear ARMAX model

(b) With a neural network model

(c) With a hybrid quasi-ARMAX model

Figure 2.18: Simulation of the identified models on validation data. The solid line shows the true measurements and the dashed line the simulated model outputs.

respectively. The simulations of the identified models on the validation data are shown in Fig. 2.19, which give the RMS errors of 0.0866, 0.0748, 0.0700 and 0.0288, respectively. The results obtained in this identification shows that it is possible to identify a system in a multiresolution way.

## 2.6 Some Aspects of the Hybrid Models

We have shown that the hybrid quasi-ARMAX model has a linear ARMAX structure, flexibility and simplicity. In this section, we will discuss its relations with some existing linear and nonlinear models and its possible applications.

### 2.6.1 Relations with the Existing Linear and Nonlinear Models

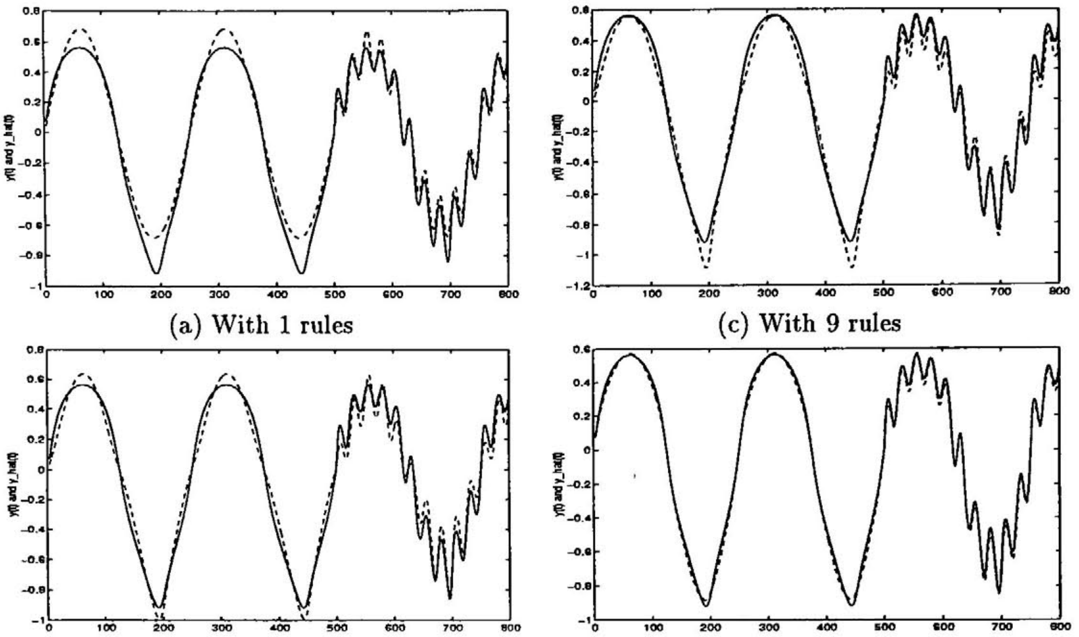
In order to understand the hybrid quasi-ARMAX model well, we will compare it with various existing linear and nonlinear models and find out their relations.

#### (1) Relations with linear ARMAX model

As described by (2.79), a linear ARMAX model has an ARMAX structure with constant coefficients

$$y(t) = \sum_{i=1}^{n} a_i y(t-i) + \sum_{i=1}^{m} b_i u(t-i) + \sum_{i=1}^{l} c_i e(t-i) + e(t). \tag{2.79}$$

Therefore, a hybrid quasi-ARMAX model can be considered to be a generalized ARMAX model that has an ARMAX structure with coefficients which are functions of input-output variables. From a viewpoint of local linearization, we may consider the hybrid quasi-ARMAX model as a scheme to realize local linearization for all operation points. Furthermore, the scheme may also be closely related with *variable structure techniques* for identification and control.



(a) With 1 rules

(c) With 9 rules

(b) With 3 rules

(d) With 18 rules

Figure 2.19: Simulation of the fuzzy inference based multi-models on validation data. The solid line shows the true measurements and the dashed line the simulated model outputs.

### (2) Relations with modeling scheme based on interpolation theory

A practical modeling scheme for nonlinear systems is based on interpolation theory [45, 119]. In such scheme, the operating region is first divided into several parts. Then rather simple (for instance, linear) models are employed as local models to approximate the system in each part of operating region (multi-model approach). The total performance of the modeling is finally obtained based on interpolation theory.

On the other hand, from the expression (2.11), we know that the hybrid quasi-ARMAX model is equivalent to a hybrid model combining a linear ARMAX model and a multi-ARX-model consisting of several local linear ARX models with employing interpolation using the 'basis functions' in the NNM. In the multi-ARX-model part, the operating region is divided into $M$ parts and $M$ local ARX models has been employed for those parts. Therefore, the hybrid quasi-ARMAX model combines the properties of the modeling scheme based on interpolation theory.

### (3) Relations with semi-physical model

Semi-physical modeling is one of approaches using grey-box type models, in which physical insight is used to suggest certain nonlinear combinations of measured data signal. These new signals are then subjected to model structures of black box character [68, 67]. For example,

$$y(t) = \theta^T \varphi_{\rm sp}(t) \tag{2.80}$$

$$\theta^T = [\theta_1\ \theta_2\ \theta_3\ \theta_4] \tag{2.81}$$

$$\varphi_{\rm sp}(t) = [y(t-1)\ y(t-2)\ u^2(t-1)\ u^2(t-2)]^T \tag{2.82}$$

where the regression vector (2.82) is assumed to be suggested using physical insight. The basic idea of semi-physical modeling is to construct a desired predictor of (2.80), where the elements of $\varphi(t)$ can be linear as well as nonlinear. As stated in [67], a drawback of the semi-physical modeling procedure is the computational complexity which requires both symbolic and numerical computations.

Obviously, our objective to achieve the model *simplicity* in the hybrid quasi-ARMAX modeling is the same as that of semi-physical modeling. By comparing our third expression (2.14) of the hybrid quasi-ARMAX model with the semi-physical model (2.80), we may find that the hybrid quasi-ARMAX modeling scheme is a special type of semi-physical modeling approach. The regression vector $\varphi_{NL}(t)$ in our case is determined using knowledge information by employing nonlinear nonparametric modeling, e.g., adaptive fuzzy modeling,

$$\varphi_{NL}(t) = [\varphi^T(t),\ \varphi^T(t) \otimes \varphi^T_{\mathcal{N}_f}(t)]^T \tag{2.83}$$

where $\varphi^T_{\mathcal{N}_f}(t) = [\mathcal{N}_f(\boldsymbol{p}_j, \varphi(t)),\ j = 1, ..., M]$. We may expect that less knowledge information and less computational complexity are required in our case.

### (4) Relations with nonlinear nonparametric models (NNMs)

A hybrid quasi-ARMAX model consists of a group of certain NNMs. As shown in Fig 2.2, it can be seen as a specially constructed *associative memory networks*, which consists of two hidden layers: the first layer (next to the input layer) with weights determined by a set of simplified NNMs. The second layer with weights simply taking the time delayed value of the system input and output. As will be shown in the followed chapters, it is more suitable for control design and system analysis than a traditional neural network. Therefore, the hybrid quasi-ARMAX model may be considered as a NNM which is specially constructed by introducing better compromises to the trade-off among *structural linearity*, *flexibility* and *simplicity*.

## 2.6.2 Applications of the Hybrid Model

The hybrid quasi-linear black-box models have been shown to have several distinctive features: *structural linearity*, *flexibility* and *simplicity*. In this section, we discuss several possible application examples briefly, some of which will be further discussed in Chapter 4, Chapter 5 and Chapter 6.

### 'Simulation' or 'Prediction' Applications

In the experimental studies of Section 2.5, we have already shown that the proposed models are suitable for such applications. After identifying the models using estimation data, we simulated the model on validation data to test the model's qualities. We have shown that the proposed models have nice 'simulation' and 'prediction' ability.

### Adaptive Control of Nonlinear Systems

Taking the advantage of its flexibility, we may use the hybrid quasi-linear black-box models to replace *neural networks* in most existing neural networks based control systems. Because the proposed models are linear in the parameters to be adjusted, we may expect the control systems to have better convergence and stability properties. Furthermore, we can construct the hybrid model is linear not only in the parameters to be adjusted but also in the one-step past input variable. Then the control design of nonlinear systems can be done under the framework of linear control theory based on the linear properties. For example, the well known STR controller may be extended to nonlinear cases. We will discuss this application in Chapter 6.

### System Analysis of Nonlinear Systems

From the expression (2.11), we know that the parameters in the hybrid quasi-ARMAX model have useful physical interpretations. Using appropriate estimation approach, the estimate of the parameter $\theta$ may be used as the best linear approximation of the system. And by appropriately selecting the 'basis functions' in NNM, the parameters $\Omega_j$ describes a local linear approximation in the region around the $j$th node. These information is obviously useful for system analysis.

### Robust Fault Detection and Control

Among the parameters of the hybrid quasi-ARMAX model, $a_i$, $b_i$ and $c_i$ are related to the linear property of model, while $\omega_{vj}$'s stand for the nonlinear property. If we identify the model in such a way that the results give an interpretation that $\hat{a}_i$, $\hat{b}_i$ and $\hat{c}_i$ describes a linear ARMAX model for the best linear approximation of the system, while the $\Delta a_{i,t}(\hat{\omega}_{vj})$ and $\Delta b_{i,t}(\hat{\omega}_{vj})$ describes the error due to nonlinear undermodeling. The hybrid quasi-ARMAX model has potential application to robust fault detection and robust control for systems with nonlinearity. Fault detection of nonlinear systems using the hybrid quasi-ARMAX model will be discussed in Chapter 5.

## 2.7 Conclusions

In the toolbox for system identification techniques, one should have black-box models for nonlinear dynamical systems available. It is true that it is preferred to use physical insight to build up the nonlinear effects in a model, since this typically can be done using fewer parameters. However, such insight is not always available, and if linear approximative models are not good enough, there is no other choice than to turn to black box structures.

This topic is not at all new. The "classical" literature on the subject seems to have concentrated on global basis function expansions, such as Volterra expansions. These have apparently had limited success. The topic was really revived by the onslaught of neural network applications. Recently, some authors have suggested to use the nonlinear black-box models based on nonlinear structures such as neural networks or wavelet networks etc., see [92, 46]. However, the latter ones have resulted in an abrupt abandonment of the the highly successful linear black-box modeling methods as soon as a small amount of nonlinearity is introduced.

In this chapter, we have introduced a hybrid (linear-nonlinear) model structure. The hybrid modeling scheme can be considered as an approach to construct a model structure which is between a linear black-box model structure and a certain nonlinear black-box model structure. By finding a better compromise to the trade-off between the model flexibility and the model simplicity, we have obtained a hybrid model equipped with a linear structure, flexibility and simplicity. It is easier to do nonlinear system identification with the new model structure, since it has better convergence properties and better properties for dealing with correlated noise. The effectiveness has been confirmed through numerical simulations. In the following chapters, it will be shown that the hybrid models are useful in several applications (fault detection, control design, etc.).

One of the distinctive features of the new model structure is that both observed data and knowledge information can be used in the parameter estimation. We have suggested some strategies and hints for incorporating knowledge information. In order to make the hybrid modeling scheme less heuristically dependent, further investigations are however needed to develop an algorithm for incorporating knowledge information in the modeling or during the parameter estimation.

# Chapter 3

# A Hybrid Method Using Genetic Algorithm for Identification

## 3.1 Introduction

In many model-based approaches, it is an essential problem to build up a good model. System identification has been known as such a technique. When model structure is determined, the main task of the identification is to estimate model parameters, which is usually determined as the global minimum point of the loss function that is not guaranteed to be unimodal. There have been proposed many identification algorithms, most of which are optimization-based methods [94, 26, 70]. However, it should be noticed that there is a potential risk that an optimization-based algorithm is stuck at a local minimum, which may result in a poorly identified model. Such a risk will further increase in the presence of unmodeled dynamics [54].

On the other hand, in the hybrid quasi-linear black-box modeling and identification approach (see Chapter 2), one usually benefits by implementing the identification in a hierarchical way, i.e., the system is first identified by using a corresponding linear model and the estimate of the linear approximation is then used as prior knowledge in the second stage identification. This is true because of the following two reasons: (1) the parameters of the hybrid quasi-linear black-box model corresponding to the linear part are not all independent of the parameters of nonlinear part. This allows to model the linear characteristics first and thus improve the convergence property; (2) it is found experimentally that a better noise model can be obtained in the first stage than in the second stage [36, 37]. There inevitably exists large unmodeled dynamics in the first identification stage. It therefore is crucial to develop a system identification algorithm which may find global minimum reliably.

There is no easy solution to the multimodality problem. It is usually well used effort to spend some time to come up with a good initial value where to start the iterations [92]. This however is not a realistic option in such cases where little prior knowledge is available. A better option is usually to search for a good initial value or to estimate the parameters directly using global searching techniques such as random search and genetic algorithm [88, 90, 51]. However, since the searchings based on these techniques are rather time-consuming, it is difficult to implement the techniques in the practical identification algorithms. In this chapter, an efficient system identification algorithm, which may find "global minimum" [1] of multimodal loss function reliably, is developed on the basis of an *effective combination* of a Genetic Algorithm (GA) and an optimization-based method.

GA is a probabilistic search algorithm based on a simple mechanics which models on genetic processes occurring in nature [24, 14]. Recently, GA has attracted much interest in system control and identification community [51, 114, 100], because it has high ability for global optimization. It is well known that GA is rather time-consuming algorithm, some work should therefore be done to improve its efficiency. There have been proposed several hybrid GAs by combining GA with certain

[1] In this paper, the term "global minimum" is used in the sense of engineering. This means that "global minimum" includes the allowable points around the true global minimum.

local searching method such as optimization-based method or simulated annealing [47, 73]. However, few of these hybrid GAs can be applied efficiently to system identification. In this chapter, we propose a Non-Standard GA (NSGA) which is suitable for system identification. In the NSGA, a new GA operator named as *development* is introduced to improve its convergence property, which is inspired by the fact that living begins adapt themselves to their environment and may be performed by using an optimization-based method. In this way, the NSGA evolves not only in a probabilistic manner (e.g. crossover, mutation) but also in a non-probabilistic manner (e.g. development). Combining this NSGA with optimization-based method, we can perform a hybrid identification, in which a nice initial value is first searched using the NSGA and then the identification is continued by an optimization-based method. The hybrid identification algorithm constructed in this way combines the reliability properties of the GAs with the accuracy of optimization-based method, while requiring a computation time only slightly higher than the latter. Furthermore, it is possible to estimate the parameters of noise model, which is difficult for the existing hybrid GA approaches.

The chapter is organized as follows: Section 3.2 describes the problem to be solved. Section 3.3 discusses briefly the optimization-based approaches, in which prediction error method (PEM) is used as an example. In Section 3.4, we propose an Non-Standard GA by introducing a new GA operator named as *development*. The hybrid identification algorithm is described in Section 3.5. Several numerical simulations are carried out to test the effectiveness of the proposed algorithm in Section 3.6. Finally, Section 3.7 is devoted to discussions and conclusions.

## 3.2 Problem Description

Let us consider the problem of estimating a model for a dynamic system on the basis of the observation of an $N$ point input-output data sequence $Z^N = [\{u_t\}, \{y_t\}]$. For simplicity we assume that the observed data $Z^N$ is generated by a SISO system. The system is considered to be a general one, which may be linear or nonlinear, time-invariant or time-variant and even with time-delay.

When identifying the system, we assume that the parametric model is a linear time-invariant ARMAX model

$$A(q^{-1},\theta)y(t) = B(q^{-1},\theta)u(t) + C(q^{-1},\theta)e(t) \tag{3.1}$$

with

$$\begin{aligned} A(q^{-1},\theta) &= 1 + a_1q^{-1} + ... + a_nq^{-n} \\ B(q^{-1},\theta) &= b_1q^{-1} + ... + b_mq^{-m} \\ C(q^{-1},\theta) &= 1 + c_1q^{-1} + ... + c_lq^{-l} \end{aligned} \tag{3.2}$$

where $\theta = [a_1 \ ... \ a_n \ b_1 \ ... \ b_m \ c_1 \ ... \ c_l]^T$ is parameter vector [2], $y(t)$ is the output at time $t$ $(t = 1, 2, ...)$, $u(t)$ the input and $e(t)$ the white noise with zero mean.

In order to estimate the model parameter $\theta$, let us introduce a regression vector

$$\begin{aligned} \varphi(t,\theta) = [&-y(t-1) \ \dots \ -y(t-n) \\ &u(t-1) \ \dots \ u(t-m) \ \varepsilon(t-1,\hat{\theta}) \ \dots \ \varepsilon(t-l,\hat{\theta})]^T. \end{aligned} \tag{3.3}$$

Then we can rewrite (3.1) into the linear regression from

$$y(t) = \varphi(t,\theta)^T\theta + e(t). \tag{3.4}$$

$\varepsilon(t,\hat{\theta})$ in $\varphi(t,\hat{\theta})$ denotes prediction error defined by

$$\varepsilon(t,\hat{\theta}) = y(t) - \hat{y}(t,\hat{\theta}) = y(t) - \varphi(t,\hat{\theta})^T\hat{\theta} \tag{3.5}$$

For this identification problem, most numerical schemes select $\theta = \hat{\theta}$ so that

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} \|y(t) - \varphi(t,\theta)^T\theta\| \tag{3.6}$$

[2] In this chapter, the notation $\theta$ denotes $\theta_c$ in Chapter 2 for simplicity.

is minimized for some norm $\| * \|$, that is

$$\hat{\theta} = \arg\min_{\theta} \{V_N(\theta)\}. \tag{3.7}$$

A typical choice is a quadratic norm

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^{N} |y(t) - \varphi(t,\theta)^T \theta|^2. \tag{3.8}$$

However, it should be noted that the loss function (3.8) is multimodal because there exists a noise model $C(q^{-1})e(t)$ in (3.1).

## 3.3 Optimization-Based Methods

An optimization-based numerical search is the most commonly-used method to estimate the model parameters $\theta$. Without loss of generality, we will use Prediction Error Method (PEM) in our discussion. It is well known that minimizing (3.8) with respect to $\theta$ can be carried out by a recursive PEM algorithm [70]

$$\hat{\theta}(t) = \hat{\theta}(t-1) + L(t)\varepsilon(t, \hat{\theta}(t-1)) \tag{3.9}$$

$$L(t) = \frac{P(t-1)\psi(t,\hat{\theta}(t-1))}{1 + \Gamma(t)\psi(t,\hat{\theta}(t-1))} \tag{3.10}$$

$$P(t) = P(t-1) - L(t)\Gamma(t) \tag{3.11}$$

where $\Gamma(t) = \psi^T(t, \hat{\theta}(t-1))P(t-1)$ and $\psi(t, \hat{\theta}(t-1))$ is defined by

$$\psi(t, \hat{\theta}(t-1)) = \left( \frac{\partial \varepsilon(t,\theta)}{\partial \theta} \right)^T \Bigg|_{\theta = \hat{\theta}(t-1)} = \frac{\varphi(t, \hat{\theta}(t-1))}{C(q^{-1}, \hat{\theta}(t-1))}$$

The derivation of the algorithm can be found in Section 2.5 of Chapter 2. It should be point out that because the loss function is not unimodal, the algorithm (3.9)-(3.11) has a potential risk to get caught at a local minimum. The risk has been found to be increased when the modeling contains unmodeled dynamics. In order to find the global minimum reliably, we shall develop a hybrid identification algorithm by combining GA with the PEM, in which an NSGA is introduced to search for a better initial value.

## 3.4 Non-Standard Genetic Algorithm (NSGA)

Genetic Algorithms, which model on the genetic processes occurring in nature, are adaptive methods that may be used to solve search and optimization problem [5]. They work with a population of individuals (also known as chromosome), which represent possible solutions to a given problem. Each individual is assigned a fitness score according to how good a solution to the problem it is. The highly fit individuals are given opportunities to reproduce, by cross breeding with individuals in the population. The least fit of the members of the population are less likely to get selected for reproduction, and so die out. Over several generations, individuals tend to be identity and the algorithm is converged.

Figure 3.1 shows the principle of the random search via crossover and mutation in a conventional GA. The search based on this kind of "probabilistic" searching rules has been demonstrated to be effective for solving the multimodality problem. However, since individual developmental process is disallowed in a conventional GA because of premature problem, its convergence speed is rather low. Furthermore, the conventional GA can not be applied to solve the multimodal problems resulted from noise models because the fitness is difficult, if not impossible, to calculate. In order to solve this problem and to improve the convergence properties of GA, some improvements should be made for

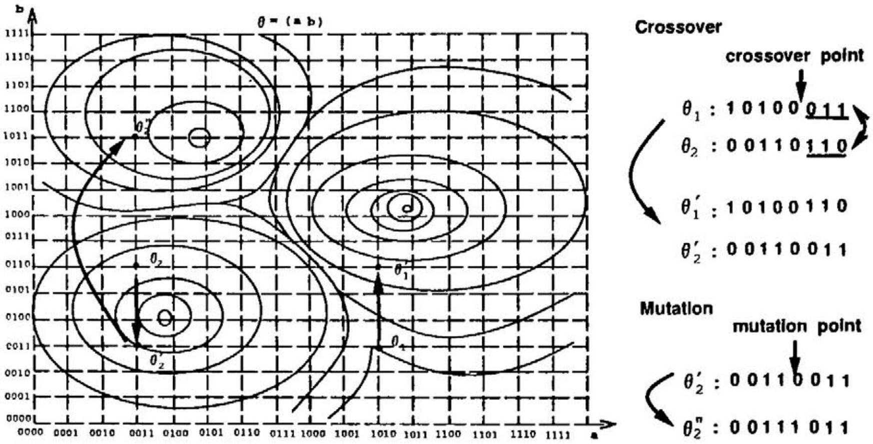


Figure 3.1: Crossover and mutation in a conventional GA

the GA. Considering the peculiarity of system identification, and the fact that living beings adapt themselves to their environment, we introduce a limited development to improve the convergence property of GA and also to make it possible to calculate the fitness of individual. The improvement will be realized by introducing a new GA operator named as *individual development* or *development* and performing it by the optimization-based method. That is, before the fitness is evaluated, each individual is allowed to develop itself via optimal rule with a limited degree. We will name the hybrid GA introduced in this way as Non-Standard GA (NSGA). The NSGA evolves in a manner of "probabilistic ⇒ optimal ⇒ probabilistic ⇒ optimal ⇒ ...", which combines the advantages of both GA and optimization-based method. It provides a better compromise to the trade-off among the reliability, accuracy and computation time.

Figure 3.2 show the scheme and the structure of the Non-Standard Genetic Algorithm (NSGA). There two difference between the NSGA and a conventional GA. One is that the NSGA contains a new operator called *development* which is based on optimal rule. The second is that the NSGA employs a real coded GA, in which the chromosome vector is a vector of floating point numbers instead string of bits. We will discuss the NSGA follows in details.

### 3.4.1 Representation of the Solutions

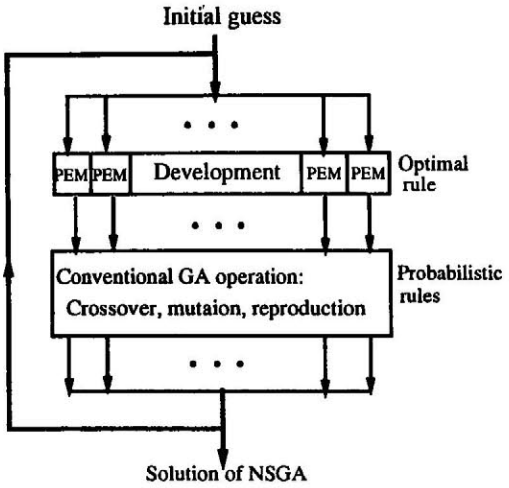
To solve a problem with GA, potential solutions have first to be encoded as chromosome. Generally, chromosome has been string of bits, but also other alphabets can be used [33]. For an ARMAX model described by (3.1), it would seem particularly natural to represent the genes directly as real parameters. Then a chromosome vector is a vector of floating point numbers. This is known as Real Coded GA.

$$C^t = (a_1, ..., a_n, b_1, ..., b_m, c_1, ..., c_l) \tag{3.12}$$

### 3.4.2 Development and Fitness Evaluation

In a genetic algorithm, selection is done based on the fitness of individual. A fitness function should be chosen that the fitness of an individual indicates the "goodness" of the individual which represents a solution to the problem. In the case of the PEM, the fitness function could simply be the inversion of the variance of prediction error or the loss function (3.8).

Generally, there is no individual developmental process in a conventional GA. In our NSGA, we have introduced a limited development via optimal rule to each individual. This can improve the convergence of the GA. The *degree* of the development should be chosen carefully according to each problem to be solved in order to keep the reliability.



(a) Schematic representation of the NSGA

```
t:=0;
initpopulation P(t);
develop P(t);
evaluate P(t);
while not done do
    t:=t+1
    P':=selectparents P(t);
    cross P'(t)
    mutate P'(t);
    develop P'(t);
    evaluate P'(t);
    P:=survive P, P'(t);
od
```

(b) Structure of the NSGA

Figure 3.2: The Non-Standard Genetic Algorithm (NSGA)

In the NSGA, *fitness evaluation* and *development* are done in one function described as (3.13). By giving the old individual ($old\_indi$), the degree of development ($m$) and input-output data $Z^m$, the function will return developed individual ($new\_indi$) and the fitness ($fit$).

$$[fit, new\_indi] = \text{RPEM}(Z^m, old\_indi, m) \tag{3.13}$$

where RPEM is the recursive PEM described by (3.9)-(3.11), and the fitness is calculated by inverting the variance of the variance of prediction error

$$fit = \frac{1}{\beta + \frac{1}{m}\sum_{t=1}^{m} \varepsilon^2(t, \hat{\theta}(t-1))} \tag{3.14}$$

where $\beta$ is an appropriate constant, which is introduced to change the shape of fitness function.

### 3.4.3 Crossover and Mutation

In the case of Real Coded GA, each of the randomly chosen positions is naturally located between two genes. Usually, two strategies, Simple Crossover and Max-Min-Arithmetical Crossover can be used [33].

Suppose two chromosomes of $C_a^t = (c_1, ..., c_k, ...c_H)$ and $C_b^t = (\bar{c}_1, ..., \bar{c}_k, ...\bar{c}_H)$ to be crossed. For Simple Crossover, two offsprings will be obtained:

$$\begin{array}{l} C_a^{t+1} = (c_1, ..., c_{k-1}, \bar{c}_k, ...\bar{c}_H) \\ C_b^{t+1} = (\bar{c}_1, ..., \bar{c}_{k-1}, c_k, ...c_H) \end{array} \tag{3.15}$$

where the crossover point (between $k - 1$ and $k$) is determined randomly. And for Max-Min-Arithmetical Crossover, four offsprings will be generated:

$$\begin{array}{l} C_1^{t+1} = \alpha C_a^t + (1-\alpha)C_b^t \\ C_2^{t+1} = \alpha C_b^t + (1-\alpha)C_a^t \\ C_3^{t+1} = (\min\{c_1, \bar{c}_1\}, ..., \min\{c_H, \bar{c}_H\}) \\ C_4^{t+1} = (\max\{c_1, \bar{c}_1\}, ..., \max\{c_H, \bar{c}_H\}) \end{array} \tag{3.16}$$

where $\alpha \in [0, 1]$ is a random value.

Crossover is not usually applied to all pairs of individuals selected for mating. A random choice is made, where the likelihood of crossover being applied is typically between 0.6 and 1.0.

Unlike in a standard GA, the mutation in a Real Coded GA is applied to each child individually. It randomly alters each gene with a small probability. A non-uniform stage can be used [33].

Suppose chromosome $C^t = (c_1, ..., c_k, ...c_H)$ is selected to be mutated, the result chromosome will be $C^{t+1} = (c_1, ..., \tilde{c}_k, ...c_H)$, with

$$\tilde{c}_k = \begin{cases} c_k + \Delta(t, c_{\max} - c_k) & \text{if } r = 0 \\ c_k - \Delta(t, c_k - c_{\min}) & \text{if } r = 1 \end{cases} \tag{3.17}$$

where $r$ is a randomly generated binary digit, $c_k \in [c_{\min}, c_{\max}]$, and the function $\Delta(t, y)$ returns a value in the range [0, y] such that the probability of $\Delta(t, y)$ being close to 0 increases as $t$ increases. This property causes that the operator makes a uniform search into the initial space when $t$ is small, and very locally at later stages.

### 3.4.4 Reproduction

During the reproductive phase of the GA, the individuals of the next generation should be selected based on the rule of natural selection: The highly fit individuals are given opportunities to reproduce, by cross breeding with individuals in the population, while the least fit of the individuals are less likely to get selected for reproduction, and so die out.

Usually reproduction is carried out as a two-stage process. It starts with the *current population*. Selection is applied to the current population to create an *intermediate population*. Then the *next population* is created from the *intermediate population*. Following two strategies can be chosen.

In one strategy, the individuals of *intermediate population* are generated from selected parents which give their offsprings proportional to their fitness. Then pairs of individuals are chosen randomly as parents for mating by crossover and give their offsprings to form the *next population*.

In the other strategy, *intermediate population* is generated by including the individuals of *current population* and their offsprings generated by applying Max-Min-Arithmetical crossover. The fitness of each individual of the *intermediate population* is evaluated. Then good individuals of the *intermediate population* are chosen to form the *next population*. This method is known as truncation(or extinctive) selection.

## 3.5 Hybrid Identification Algorithm

A hybrid identification algorithm based on GA and optimization-based method can usually be described as a two-stage process, in which a better initial value is first searched using GA, and then the identification is continued by optimization-based method. We here will follow this idea to construct a hybrid identification algorithm. The NSGA discussed in the previous section is used for searching a better initial value. Such a hybrid identification algorithm can be shown in Fig. 3.3 (left).

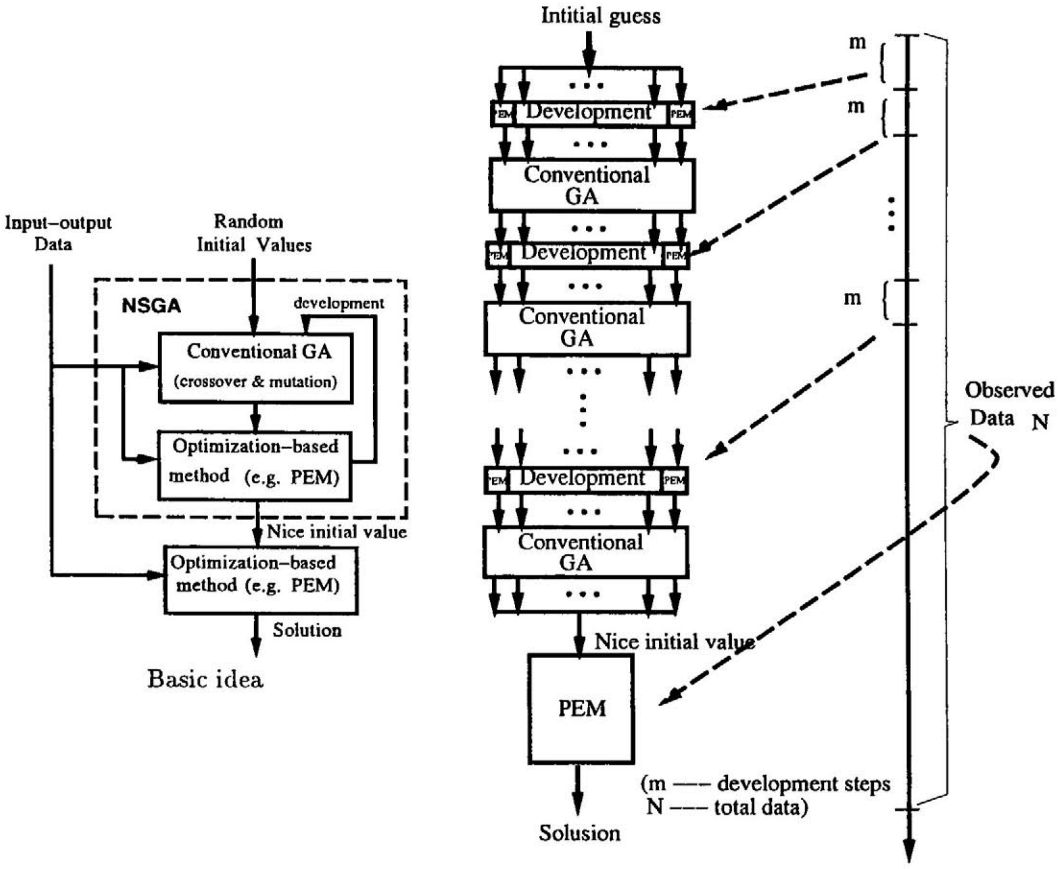
### 3.5.1 Identification Algorithm

The hybrid identification algorithm can be summarized in the following steps:

- Step 1: Initialize parameters of the NSGA
  Specify population size, crossover probability, mutation probability, development step and maximum generation with appropriate values.

- Step 2: Generate the first generation
  If there is no *a priori* knowledge, individuals in the first generation are usually given to be random values.

- Step 3: Search with the optimal rule and evaluate the fitness
  Develop each individual with limited degree based on optimal rule (3.9)-(3.11) and evaluate the fitness of each individual via (3.14).



Basic idea

On-line implementation

Figure 3.3: The hybrid identification algorithm

- Step 4: Search with the probabilistic rules
  Apply crossover operator (3.15) or (3.16) and mutation operator (3.17) to individuals of population and then reproduce offsprings.

- Step 5: Reproduce and select new generation based on the fitness

- Step 6: Determine whether the algorithm is converged or maximum generation is reached. If the answer is 'no', go to step 3 otherwise go to next step.

- Step 7: Identify model parameters
  Average individuals in the final generation to obtain a nice initial value and then continue the identification using (3.9)-(3.11)

### 3.5.2 On-line Implementation of the Algorithm

The algorithm will be implemented on-line. Figure 3.3 (right) shows an example of such implementation. For the case where the specifications of the NSGA are that population = 15, development step = 30, actual generations = 6, the computation time for the NSGA is approximately equivalent to that of a 2700-step PEM algorithm. Since with the initial value searched by the NSGA, the number of the identification step in the second stage by PEM is usually far smaller than that of a PEM with random initial value, the computation time for the hybrid identification algorithm is not increased significantly.

## 3.6 Numerical Simulations

In this section, some numerical simulations are carried out to demonstrate the effectiveness of the proposed hybrid identification algorithm. Since we mainly want to show the reliability of the hybrid identification to find global minimum, the simulations will be carried out as Monte Carlo tests. The reliability will be compared with the conventional optimization-based method (prediction error method (PEM)) which has risk to be stuck at a local minimum.

In each simulation, first, 2000 input-output data sets are sampled from the system; Then Monte Carlo tests with 300 trials are conducted by using both the proposed hybrid method and the PEM, in which the initial values of each trial are given randomly. Considering the fact that the value of $\hat{\sigma}^2 = \sum_{t=1}^{N} \varepsilon(t)^2/N$ becomes larger when an algorithm is stuck at a local minimum, we show the results in histogram of $\hat{\sigma}^2$. It should be noticed that the reliability of the proposed approach is demonstrated by the concentrated histograms of prediction errors obtained via Monte Carlo test, while the distributed histograms of prediction error obtained via Monte Carlo test of PEM with random initial value are mainly used to show the multimodality of loss function.

In these simulations, when PEM is used, the identification begins with a randomly given initial value and continues for 2000 steps; When the proposed hybrid method is used, a better initial value is searched using the NSGA, where the parameters of the NSGA used are *population= 15, crossover probability=0.8, mutation probability=0.1, development step=30, maximum generation=10.* Then the identification is continued using the PEM for 500 steps.

*Example 1*: Consider a second order SISO system described by

$$G(s) = \frac{k_0}{s(T_0 s + 1) + k_0} \tag{3.18}$$

where $T_0 = 0.74$, $k_0 = 3$. After being sampled with a period of $0.25sec$, the discrete-time system is corrupted by a non-white noise sequence $\{\nu(t)\}$:

$$\nu(t) = e(t) + c_1 e(t-1) + c_2 e(t-2) \tag{3.19}$$

where $c_1 = -1.2$, $c_2 = 0.5$, and $e(t) \in N(0,\ 0.1)$ is white Gaussian noise. The input of the test is pseudo-random binary sequence (PRBS).

When the system is identified, a second order ARMAX models described by (3.1) is used as model, where $n = m = l = 2$. Since there exists a noise model in (3.1), the loss function is multimodal hence there is a risk that an optimization-based method such as the PEM is stuck at a local minimum. Figure 3.4 (left-upper) shows the results of Monte Carlo tests in histogram, where x-axis denotes $\hat{\sigma}^2$ and y-axis denotes numbers(probabilities), (a) is the result using proposed method and (b) is the result using the PEM.

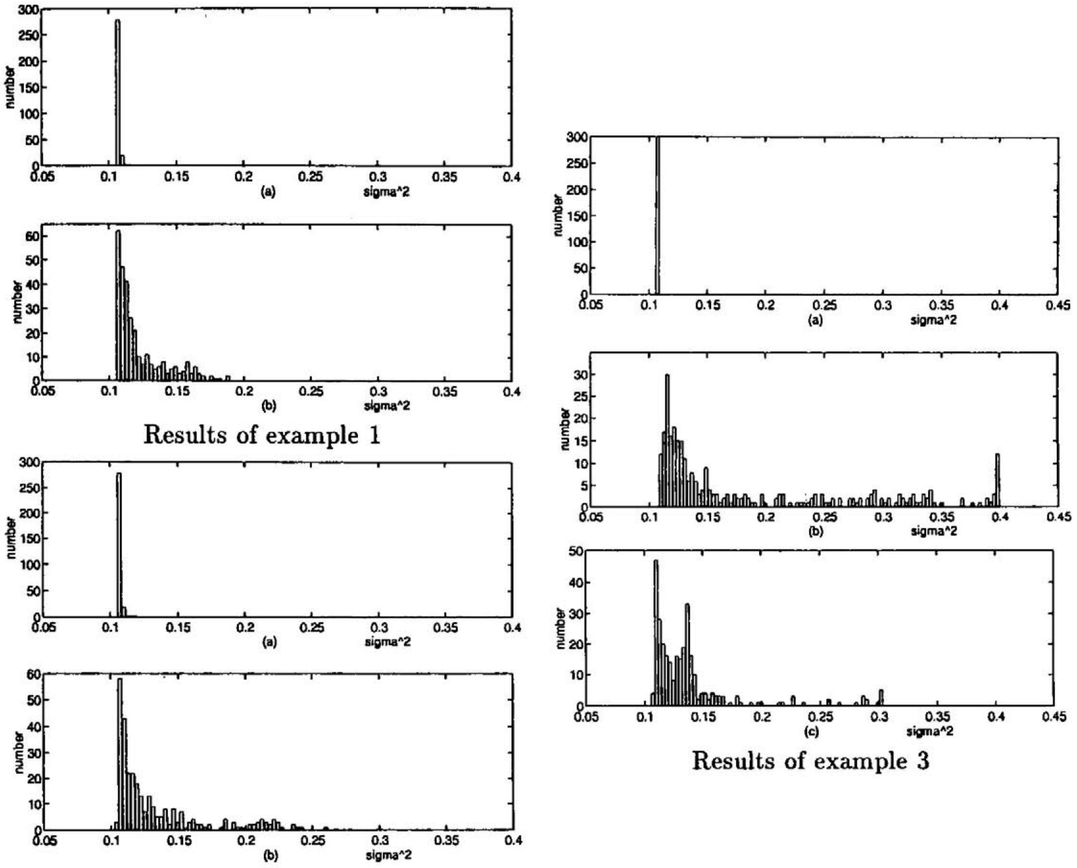
*Example 2*: In this example, we consider a discrete-time SISO system governed by

$$\begin{aligned} y(t) &= (1.5 - 0.01y(t-1))y(t-1) - 0.7y(t-2) \\ &+ 0.1u(t-1) + 0.1u(t-2) + \nu(t) \end{aligned} \tag{3.20}$$

where $\nu(t)$ is a non-white noise described by (3.19) and $u(t)$ is a PRBS.

The model used to identified this system is the same as that of example 1. The results of Monte Carlo tests are shown in figure 3.4 (left-lower) where (a) is the result using proposed method and (b) is the result using the PEM. Since in this example, the nonlinearity of system caused unmodeled dynamics, it seems that the probability of an optimization-based method being stuck at local minima was increased. However, our proposed method did not get caught in all the trials.

In the *Example 1* and *Example 2*, we have shown that the proposed algorithm is very *efficient*, which can find the "global minimum" *reliably* (see concentrated histograms Fig. 3.4 (left) (a) ) using short data sequence (500) and reasonable computing time (about 3200 steps of PEM) in multimodality problems (see distributed histograms Fig. 3.4 (left) (b)). On the other hand, the proposed method is basically an optimization-based method (PEM), while only the initial value is searched by the NSGA. In next example, we will compare it with those using initial values given in different ways.



Results of example 1

Results of example 3

Results of example 2

Figure 3.4: Histograms of $\hat{\sigma}^2$ (a) by using the proposed method; (b) by using PEM with random initial value; (c) by using PEM with LS estimations as initial value.

*Example 3*: a third order SISO system governed by (3.21) is considered as true system, in which a time delay of $0.25sec$ is added to increase the unmodeled dynamics.

$$G(s) = \frac{ke^{-\tau s}}{s(T_0 s + 1)(T_1 s + 1) + k} \tag{3.21}$$

where $k = 2.5$, $T_0 = 0.1$, $T_1 = 0.5$, $\tau = 0.25sec$.

The sampling period and the noise sequence for the discrete-time system and the identification model used are the same as those of example 1.

In the Monte Carlo tests, each of trials is carried out by PEM with 2000 steps using an initial value (a) searched by using the NSGA; (b) given randomly; (c) $a_i$ and $b_i$ searched by LS while $c_i$ given randomly.

Figure 3.4 (right) shows the results of Monte Carlo tests. We can see that the proposed method can achieve "global minimum" reliably, while other two methods can not. This example again confirms the effectiveness of the proposed method.

Table 3.1 shows mean values of the identified parameters and their standard variations in the Monte Carlo tests of example 3. We can see that using the proposed method, the standard variations of identified model parameters are very small.

Table 3.1: Mean values and Standard variations

| | PEM with random initial value | | PEM with LS estimation as initial value | | the proposed method | |
|---|---|---|---|---|---|---|
| | Mean Value | Standard Variation | Mean Value | Standard Variation | Mean Value | Standard Variation |
| $a_1$ | -1.3163 | 0.4444 | -1.4972 | 0.1451 | -1.5544 | 0.00003 |
| $a_2$ | 0.6123 | 0.3360 | 0.7585 | 0.1086 | 0.7707 | 0.00001 |
| $b_1$ | -0.0919 | 0.0864 | -0.1008 | 0.0444 | -0.0777 | 0.00004 |
| $b_2$ | 0.2126 | 0.1072 | 0.2325 | 0.0585 | 0.2638 | 0.00002 |
| $c_1$ | -0.6442 | 0.7078 | -0.9871 | 0.3365 | -1.2410 | 0.00026 |
| $c_2$ | 0.3438 | 0.1669 | 0.4082 | 0.1512 | 0.5082 | 0.00027 |

## 3.7 Discusions and Conclusions

It has been shown that when the loss function is not unimodal, there is a potential risk that an optimization-based identification algorithm is stuck at a local minimum. The risk has been found to be increased in the case of presence of unmodeled dynamics. This usually results in a poorly identified model. It therefore is crucial to develop an identification algorithm which may find global minimum reliably.

Many GA based hybrid methods have been proposed to solve the local minimum problems. From the viewpoint of system identification, there are two problems in the existing hybrid GAs: (1) few of them can be applied efficiently to system identification; (2) they usually can not solve the local minimum problems resulted from noise models. In this chapter, we have first proposed a Non-Standard GA by introducing a new GA operator named as *development* which is performed by an optimization-based method. The NSGA is typically suitable for system identification. By using the NSGA to search a better initial value for an optimization-based algorithm, we have then developed a hybrid identification algorithm. The effectiveness of the algorithm has been confirmed through numerical simulations. The proposed hybrid methods combines the reliability of GAs with the accuracy of optimization-based methods, while requiring a computation time only slightly higher than the latter. Furthermore, it can be used to solve the local minimum problem resulted from noise model.

In our current research, we have applied the hybrid identification algorithm to fault detection and nonlinear system identification [54, 36], and found it very effective and useful. We thus believe that the reliably identification algorithm will find wide applications in the practical cases.

It is known in the literature that a GA-based identification method can be applied not only to solve multimodality problem, but also to increase the robustness against disturbance. In our future research, we will investigate the robustness against disturbance. Furthermore, investigations are also needed to improve the efficiency and the effectiveness when the algorithm is applied to the case where there are a rather large number of parameters to be estimated.

# Chapter 4

# KDI-Based Robust Fault Detection of Dynamic Systems

## 4.1 Introduction

Model-based fault detection approach has received an increasing attention recently [20, 41] because of that it is portable and the model parameters provide useful information for fault analysis. One of the model-based methods is based on parameter estimation. The parameter estimation approach is based on the assumption that the faults are reflected in the physical system parameters such as friction, mass, viscosity, capacitance, inductance, etc. The basic idea of the detection method is that the parameters of the actual process are repeatedly estimated on-line using well known parameter estimation methods and the results are compared with the parameters of the reference model obtained initially under fault-free conditions. Any substantial discrepancy indicates a change in the process and may be interpreted as a fault [41, 87, 26, 61]. A typical parameter estimation approach is the KDI-based fault detection scheme proposed by K. Kumamaru and coworkers [61, 60], which use Kullback discrimination information as an index for model discrimination. In Appendix D, we give a detail review of the KDI-based fault detection scheme and point out that there were two important assumptions in the scheme, one of which is that there is no unmodeled dynamics in the modeling. However, in practical applications, model uncertainties are inevitable for the following reasons: 1) the system of interest is usually unknown to some extent. An exact mathematical modeling of the system is impossible; 2) a low complex (for instance, low order, linear) model is often used for convenience, since a high order or nonlinear model may make fault analysis difficult. Therefore, it is crucial and highly motivated to relax the assumption, i.e., to extend the scheme to the case where there exists model uncertainty.

Recently, it attracts much interest in the challenging problem to increase the robustness of a model-based fault detection and isolation (FDI) methods with respect to model uncertainties, see e.g. [20, 107]. The robust FDI methods proposed so far are mainly realized by the following way: first assume that the fault and the model uncertainty are mutually independent extra input signals of the system, then develop a residual generation scheme which is sensitive to fault but insensitive to the model uncertainty. A typical such scheme is proposed by Frank and Wunnenberg (1989) [22], Patton and Kangethe (1989) [84] based on observer design approaches, in which the robustness is achieved by appropriately design the observers. The most attracting feature of their approaches is that under certain ideal conditions, a full decoupling can be reached, which achieves invariance between different fault effects or between the effects of faults and of unknown inputs (i.e. disturbance, uncertainty, ...) independent of the fault modes. In their approaches, however, the distribution matrices of fault and unknown inputs should be known in the state-space representation. This is sometimes infeasible, especially in the case where an input-output model has to be used.

In this chapter, we propose a robust input-output model-based fault detection scheme using robust identification techniques, which is obtained by extending the KDI-based fault detection scheme to the

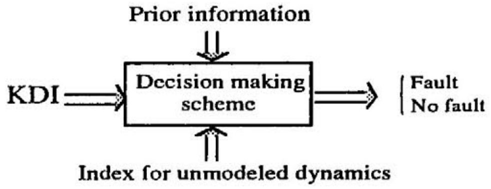


Figure 4.1: Robust fault detection by using decision making scheme

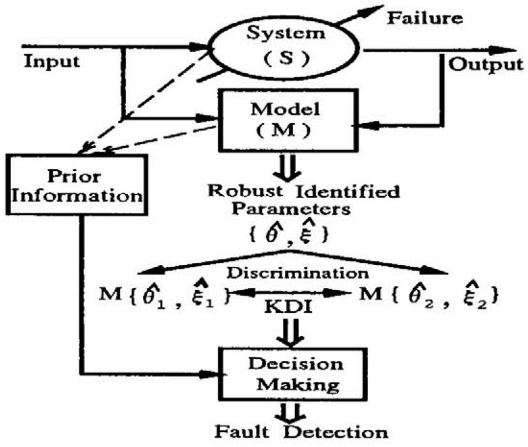

Failure
System (S)
Input
Output
Model (M)
Prior Information
Robust Identified Parameters
$\{\hat{\theta}, \hat{\xi}\}$
Discrimination
$M\{\hat{\theta}_1, \hat{\xi}_1\}$
KDI
$M\{\hat{\theta}_2, \hat{\xi}_2\}$
Decision Making
Fault Detection

Figure 4.2: The basic structure of KDI-based robust fault detection system

case where there exists model uncertainty [55]. It is reasonable to assume that in parameter estimation the effects of modeling error and of system fault are different, because obviously the former may have random properties. Based on this assumption, if we can derive an index for evaluating unmodeled dynamics, a robust fault detection can be achieved by developing a decision making scheme so that the fault detection is carried out on the basis of the index of KDI, the index of unmodeled dynamics and other information (variance of disturbance, etc.), see Fig. 4.1

It follows that the basic structure of KDI-based robust fault detection system (FDS) can be shown in Fig. 4.2. It contains three principal parts:

### (1) Robust Identification

In this stage, a robust identification is carried out so that the monitored system is described by an identified mathematical model and the modeling error is quantified and given as some forms. In the next section, we will suggest two robust identification methods for this purpose.

### (2) Calculation of the KDI

The identified model will be compared with the reference model obtained initially under fault-free conditions. This model discrimination is executed via the KDI. The KDI, a sensitive fault detection index, is analyzed so that its evaluation for finite but fairly large data sets can be done in a feasible way, in which the descriptions of unmodeled dynamics are incorporated. Furthermore, several indexes for evaluating the unmodeled dynamics are also derived based on the results of the KDI analysis. We will discuss these in Section 4.3.

### (3) Fault Decision Scheme

Substantial discrepancy of the identified models indicates a change in the system of interest which may be interpreted as a fault. However, the index of the KDI is sensitive to the modeling error. In order to realize the robustness, fault decision making schemes are developed so that the fault detection can be carried out on the basis of the KDI, the indexes of unmodeled dynamics and other information about the system. We will discuss this issue in Section 4.4.

This chapter is organized as follows: Section 4.2 suggests two robust identification methods based on or modified from the existing robust identification methods. In Section 4.3, we analyze the KDI into a feasible form, in which the description of unmodeled dynamics is incorporated. Furthermore, several indexes for evaluating unmodeled dynamics are introduced. In Section 4.4, fault detection making schemes are developed for realization of robustness on the basis of the KDI, the indexes of unmodeled dynamics and other information about the system. Several numerical simulations are carried out to test the effectiveness of the proposed robust fault detection schemes in Section 4.5. Finally, Section 4.6 is devoted to discussions and conclusions.

## 4.2 Robust Identificaiton

In the literature, 'robust identification' basically means the identification for robust control design. Since in robust control system the robustness is considered in the sense of stability, it is required that the modeling error is characterized as a quantity with a suitable 'bound'. On the other hand, it seems that there rarely exist publications discussing robust identification for system analysis such as fault detection. In this chapter, we will develop a robust fault detection scheme based on robust identification methods. Since in a fault detection system, the robustness is considered in the sense of detecting accuracy, the size of the modeling error will play a more important role than the 'bound'. Because of this difference, the results of existing robust identification for control design may not be able to be applied directly to the robust fault detection design. Hence some modifications are needed.

In Appendix E, we briefly review the existing robust identification methods with emphasis on Goodwin's stochastic embedding (*soft bound*) approach [25] and Wahlberg's *hard bound* approach [105]. Considering the particularity of KDI-based fault detection scheme, we will suggest two robust identification methods for fault detection design based on or modified from the existing robust identification methods.

### 4.2.1 Robust Identification with Soft Bound Error Description

In the stochastic embedding approach, modeling error is characterized as a random quantity with *soft bound.* The stochastic property of the unmodeled dynamics can easily be incorporated into the KDI. We here will employ the stochastic embedding approach to realize a robust identification for fault detection.

Consider a discrete-time linear SISO ARMAX system described by:

$$\begin{array}{rl} \mathcal{S}: & A_0(q^{-1})y(t) = B_0(q^{-1})u(t) + C_0(q^{-1})e(t) \\ & e(t) \in N(0, \sigma_0^2) \end{array} \tag{4.1}$$

where $A_0(q^{-1})$, $B_0(q^{-1})$ and $C_0(q^{-1})$ are scalar polynomials in the backward shift operator $q^{-1}$ with appropriate orders. And assume that $A_0$, $B_0$ and $C_0$ are described by

$$\begin{array}{l} A_0(q^{-1}) = A(q^{-1}, \theta) + A_\triangle(q^{-1}) \\ B_0(q^{-1}) = B(q^{-1}, \theta) + B_\triangle(q^{-1}) \\ C_0(q^{-1}) = C(q^{-1}, \xi) \end{array} \tag{4.2}$$

where $A_\triangle(q^{-1})$ and $B_\triangle(q^{-1})$ denote unmodeled dynamics. The $A(q^{-1}, \theta)$, $B(q^{-1}, \theta)$ and $C(q^{-1}, \xi)$ are scalar polynomials in the backward shift operator $q^{-1}$ with the orders $n$, $m$ and $l$, respectively.

$$\begin{array}{l} A(q^{-1}, \theta) = 1 + a_1 q^{-1} + ... + a_n q^{-n} \\ B(q^{-1}, \theta) = b_1 q^{-1} + ... + b_m q^{-m} \\ C(q^{-1}, \xi) = 1 + c_1 q^{-1} + ... + c_l q^{-l}. \end{array} \tag{4.3}$$

Now according to the basic idea of Goodwin's Stochastic Embedding Approach [25], a stochastic prior model may be introduced in order to define the distribution of unmodeled dynamics.

$$A_\triangle(q^{-1}) = \sum_{i=1}^{L} \eta_{ai} q^{-i} \qquad B_\triangle(q^{-1}) = \sum_{i=1}^{M} \eta_{bi} q^{-i} \tag{4.4}$$

$$\eta = [\eta_{a1} \ \cdots \ \eta_{aL} \ \eta_{b1} \ \cdots \ \eta_{bM}]^T \in N(0, C_\eta) \tag{4.5}$$

where $\eta_{ai}$ $(i = 1, ..., L)$ and $\eta_{bi}$ $(i = 1, ..., M)$ are assumed to be mutually independent.

### (1) Parametric Model

It follows from above assumptions that the parametric model is given by

$$\mathcal{M}(\theta, \xi): \ A(q^{-1}, \theta) y(t) = B(q^{-1}, \theta) u(t) + \omega(t) \tag{4.6}$$

$$\omega(t) = \psi(t)\eta + C(q^{-1}, \xi) e(t) \tag{4.7}$$

where

$$\begin{aligned} \theta &= [a_1 \ ... \ a_n \ b_1 \ ... \ b_m]^T \\ \xi &= [\mathrm{diag}\{C_\eta\} \ \sigma^2 \ c_1 \ \dots \ c_l]^T \end{aligned} \tag{4.8}$$

$$\psi(t) = [-y(t-1) \ \dots \ -y(t-L) \ u(t-1) \ \dots \ u(t-M)].$$

Introduce the following notations

$$G(q^{-1}, \Theta) = B(q^{-1}, \theta)/A(q^{-1}, \theta) \tag{4.9}$$

$$\Delta G(q^{-1}) u(t) = \psi(t)\eta / A(q^{-1}, \theta) \tag{4.10}$$

$$H(q^{-1}, \Theta) = C(q^{-1}, \xi)/A(q^{-1}, \theta). \tag{4.11}$$

Then the model (4.6) and (4.7) can be expressed as

$$y(t) = G(q^{-1}, \Theta) u(t) + \Delta G(q^{-1}) u(t) + H(q^{-1}, \Theta) e(t) \tag{4.12}$$

where $\Theta = [\theta^T \ \ \xi^T]^T$.

### (2) Estimation of $\theta$ and $\xi$

Since the term $\omega(t)$ in (4.6) is a Gaussian correlated noise, it may be treated as an MA noise process. In this way, the vector $\theta$ can be estimated by using the prediction error method (PEM), while the vector $\xi$ can be estimated by applying the maximum likelihood (ML) method in (4.7) with the data $\hat{\omega}(t)$ calculated from $\hat{\theta}$.

In order to identify $C_\eta$, let us introduce the following parameterization similar to [25]:

$$E\{\eta_{ai}^2\} = \alpha_a (\lambda_a)^i \qquad E\{\eta_{bi}^2\} = \alpha_b (\lambda_b)^i. \tag{4.13}$$

The parameter $\xi$ thus becomes

$$\xi = [\alpha_a, \ \lambda_a, \ \alpha_b, \ \lambda_b, \ \sigma^2, \ c_1, \ \dots, \ c_l]^T \tag{4.14}$$

Now introduce a vector defined by

$$W = [\hat{\omega}(1) \ \hat{\omega}(2) \ \dots \ \hat{\omega}(N)]^T \tag{4.15}$$

Then the corresponding likelihood function $p(W|\xi)$ is subject to $N(0, \Sigma)$ because of the assumptions of (4.1) and (4.5). The estimation of $\xi$ can be obtained by maximizing the log-likelihood function $l(W|\xi)$

$$\hat{\xi} = \arg \mathrm{Max}\{l(W|\xi)\} \tag{4.16}$$

where

$$l(W|\xi) = -\frac{1}{2}\log \det\Sigma - \frac{1}{2}W^T\Sigma^{-1}W + \text{const} \tag{4.17}$$

with

$$\begin{aligned} \Sigma &= \Psi C_\eta \Psi^T + E\{VV^T\} \\ V &= [\nu_1\ \nu_2\ \ldots\ \nu_N]^T \\ \Psi &= [\psi(1)^T\ \psi(2)^T\ \ldots\ \psi(N)^T]^T \end{aligned} \tag{4.18}$$

and $\nu_t = C(q^{-1}, \xi)e(t)$.

When optimization-based methods such as PEM and ML are used, the parameter estimates are determined as the global optimal point of criterion function. However, there is a potential risk that an optimization based method is stuck at a local optimum when the criterion function is not unimodal [94]. Moreover, the risk has been found to increase in the presence of unmodeled dynamics [54]. This problem can be solved by using our proposed hybrid identification method which consists of an effective combination of a genetic algorithm (GA) and an optimization-based method. The hybrid identification is executed by the optimization-based method using a nice initial value searched by a Non-Standard GA (NSGA). The NSGA evolves not only in probabilistic manner (e.g. crossover and mutation) but also in non-probabilistic manner (e.g. development), see Chapter 3 or [38] for more details.

### 4.2.2 Robust Identification with Direct Error Description

In the case of system analysis using black-box model, a comparatively simple model is occasionally used for convenience. For example, a linear ARMAX model is used to deal with a system containing nonlinearity. Two reasons may be considered for the use of simplification: (1) it is easy to realize a repeated estimation on-line for a simple model; (2) the parameters in a simple model may contain more physical interpretation for fault analysis. The unmodeled dynamics resulted from such a simple model will contain some characteristics, which can no longer be described by the stationary stochastic process model. This fact motivates us to develop an alternative approach for describing the unmodeled dynamics.

Let us first make a comparison for Goodwin's *soft bound* approach (SBA) and Wahlberg's *hard bound* approach (HBA). For a model given by

$$y(t) = G(q^{-1}, \theta)u(t) + \Delta G(q^{-1}, \xi)u(t) + v(t), \tag{4.19}$$

their difference can be described as

| | $G(q^{-1}, \theta)$ | $\Delta G(q^{-1}, \xi)$ | $v(t)$ |
|---|---|---|---|
| SBA | $G(q^{-1}, \theta)$ | stochastic description | stochastic description |
| HBA | $G(q^{-1}, \theta)$ | deterministic description | deterministic description |

Another one which is closely related with Wahlberg's *hard bound* approach is the model reduction approach proposed by E.W. Bai in [2]. In Bai's approach, the order of the estimate model is first set to be high enough so we can assume that the estimate is close to the true plant. Then by model reduction, a low-order approximation of the estimate is obtained. The difference between the estimate model and its low-order approximation may be considered as an estimate of uncertainty. In order to apply these two approaches to robust fault detection, the following modifications are required:

- A stochastic description for the disturbance $v(t)$ should be assumed in favor of the KDI-based fault detection scheme.
- Usually, $\theta$ and $\xi$ are estimated by optimizing the same criterion function. In favor of repeatedly estimating $\theta$ on-line for parameter estimation based fault detection, $\theta$ and $\xi$ should be estimated by optimizing separate criterion functions. In order to make such estimation possible, $\theta$ is required to be a *redundant parameter vector* [1]. For this purpose, hierarchical models will be constructed.

[1] *Redundant parameter vector*: Let $\theta$ and $\xi$ be two parameter vectors of the model $\mathcal{M}\{\theta, \xi\}$. The vector $\theta$ is said to be a *redundant parameter vector* if for any given $\theta$, $\xi$ and $\theta_0$, there exists a $\xi'$ that $\mathcal{M}\{\theta, \xi\} \equiv \mathcal{M}\{\theta_0, \xi'\}$ holds.

### (1) Hierarchical ARMAX Modeling

Let us consider a system described by (4.1). Assume that $A_0(q^{-1})$, $B_0(q^{-1})$ and $C_0(q^{-1})$ are described by

$$\begin{aligned} A_0(q^{-1}) &= A(q^{-1},\theta) + A_\triangle(q^{-1}) \\ B_0(q^{-1}) &= B(q^{-1},\theta) + B_\triangle(q^{-1}) \\ C_0(q^{-1}) &= C(q^{-1},\theta) + C_\triangle(q^{-1}) \end{aligned} \tag{4.20}$$

where $A_\triangle(q^{-1})$, $B_\triangle(q^{-1})$ and $C_\triangle(q^{-1})$ denote modeling uncertainty. The $A(q^{-1},\theta)$, $B(q^{-1},\theta)$ and $C(q^{-1},\theta)$ are scalar polynomials in the backward shift operator $q^{-1}$ with the order $n$, $m$ and $l$, respective

$$\begin{aligned} A(q^{-1},\theta) &= 1 + a_1 q^{-1} + ... + a_n q^{-n} \\ B(q^{-1},\theta) &= b_1 q^{-1} + ... + b_m q^{-m} \\ C(q^{-1},\theta) &= 1 + c_1 q^{-1} + ... + c_l q^{-l}. \end{aligned} \tag{4.21}$$

In the stochastic embedding approach, the modeling uncertainty is described by using stationary stochastic process model. Following the idea of *hard bound* approach, we describe the modeling uncertainty using deterministic model, e.g., ARMAX model with order $n_d$, $m_d$ and $l$

$$\begin{aligned} A_\triangle(q^{-1},\xi) &= 1 + \delta_{a_1} q^{-1} + ... + \delta a_{n_d} q^{-n_d} \\ B_\triangle(q^{-1},\xi) &= \delta_{b_1} q^{-1} + ... + \delta_{b_{m_d}} q^{-m_d} \\ C_\triangle(q^{-1},\xi) &= 1 + \delta_{c_1} q^{-1} + ... + \delta_{c_l} q^{-l}. \end{aligned} \tag{4.22}$$

Here the order $(n_d,\ m_d)$ will be chosen large enough to describe the modeling uncertainty so well that the error can be taken account into the noise model. And $n \leq n_d$ and $m \leq m_d$ will be chosen so that the model of system dynamics is of low order.

Introduce parameter vectors $\theta$ and $\xi$ defined as

$$\begin{aligned} \theta &= [a_1\ ...\ a_n\ b_1\ ...\ b_m\ c_1\ ...\ c_l]^T \\ \xi &= [\delta_{a_1}\ ...\ \delta_{a_{n_d}}\ \delta_{b_1}\ ...\ \delta_{b_{m_d}}\ \delta_{c_1}\ ...\ \delta_{c_l}]^T. \end{aligned} \tag{4.23}$$

Then it is easy to show that the vector $\theta$ is a redundant parameter vector for the model $\mathcal{M}\{\theta,\xi\}$. Furthermore, introducing the following notations

$$G = B(q^{-1},\theta)/A(q^{-1},\theta) \quad H = (C(q^{-1},\theta) + C_\triangle(q^{-1},\xi))/A(q^{-1},\theta) \tag{4.24}$$

$$\triangle G_t = \frac{-A_\triangle(q^{-1},\xi)y(t) + B_\triangle(q^{-1},\xi)u(t)}{A(q^{-1},\theta)u(t)} \tag{4.25}$$

we have parameterized model of the form

$$\mathcal{M}\{\theta,\xi\}: \quad y(t) = Gu(t) + \triangle G_t u(t) + He(t) \tag{4.26}$$

Since $\theta$ is a redundant parameter vector of $\mathcal{M}\{\theta,\xi\}$, we may assume that $\theta$ is parameter vector of a simple model $\mathcal{M}\{\theta\}$

$$\mathcal{M}_s\{\theta\}: \quad A(q^{-1},\theta)y(t) = B(q^{-1},\theta)u(t) + C(q^{-1},\theta)e(t). \tag{4.27}$$

Then $\mathcal{M}_s\{\theta,\xi\}$ and $\mathcal{M}\{\theta\}$ form a hierarchical model. We will call the robust identification based on the hierarchical modeling as a *direct description* approach.

Figure 4.3 shows another interpretation of the *direct description* approach. The system is represented by a *complex model* $\mathcal{M}\{\theta,\xi\}$ with desired accuracy and the *complex model* includes a *simple model* $\mathcal{M}_s\{\theta\}$ as its submodel. The *simple model* $\mathcal{M}_s\{\theta\}$ embedded in the *complex model* $\mathcal{M}\{\theta,\xi\}$ will be identified such that its parameter vector $\theta$ describes the main charateristics of the system. In this way, the KDI-based robust fault detection can be applied to the simple model, in which the unmodeled dynamics is directly estimated as the difference of the *complex model* and the *simple model.*

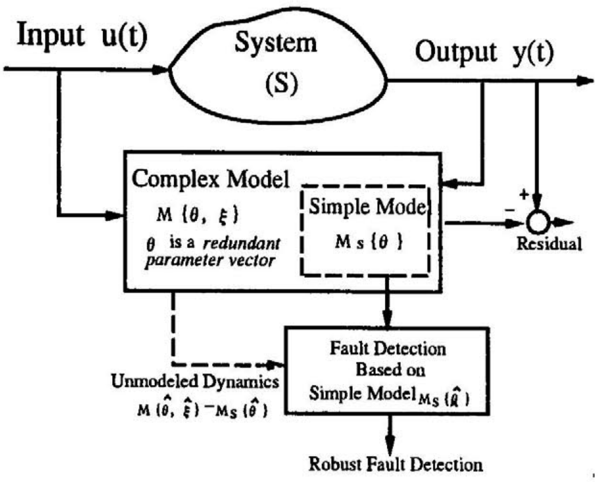


Figure 4.3: A *direct error description* approach based on hierarchical modeling

### (2) Hierarchical Hybrid Quasi-ARMAX Modeling

Let us consider a system described by

$$\mathcal{S}: \quad \begin{array}{c} \bar{A}_0(q^{-1},t)y(t) = \bar{B}_0(q^{-1},t)u(t) + C_0(q^{-1})e(t) \\ e(t) \in N(0,\sigma^2) \end{array} \tag{4.28}$$

where $y(t)$ is the output at time $t$ ($t = 1,2,...$), $u(t)$ the input and $e(t)$ the white Gaussian noise. $\bar{A}_0(q^{-1},t)$, $\bar{B}_0(q^{-1},t)$ and $C_0(q^{-1})$ are scalar polynomial functions in the backward shift operator $q^{-1}$ with appropriate orders. The index $t$ in $\bar{A}_0(q^{-1},t)$ and $\bar{B}_0(q^{-1},t)$ means that the coefficients of the polynomial functions may be functions of input-output variables, corresponding to the case where the system contains nonlinearity. When identifying the system, we assume that the $\bar{A}_0(q^{-1},t)$, $\bar{B}_0(q^{-1},t)$ and $C_0(q^{-1})$ are parameterized as

$$\begin{array}{l} \bar{A}_0(q^{-1},t) = A(q^{-1},\theta) + A_\triangle(q^{-1},t,\xi) \\ \bar{B}_0(q^{-1},t) = B(q^{-1},\theta) + B_\triangle(q^{-1},t,\xi) \\ C_0(q^{-1}) = C(q^{-1},\theta) + C_\triangle(q^{-1},\xi) \end{array} \tag{4.29}$$

where $A_\triangle(q^{-1},t,\xi)$, $B_\triangle(q^{-1},t,\xi)$ and $C_\triangle(q^{-1},\xi)$ denote modeling errors resulted from system uncertainty (including nonlinearity), and $A(q^{-1},\theta)$, $B(q^{-1},\theta)$, $C(q^{-1},\theta)$ are scalar polynomial functions described by (4.21). Since the modeling errors may be resulted from system nonlinearity, we will introduce a group of certain NNMs to describe them based on the basic idea of hybrid quasi-ARMAX modeling approach (see Chapter 2). We then further assume that $A_\triangle(q^{-1},t,\xi)$, $B_\triangle(q^{-1},t,\xi)$ and $C_\triangle(q^{-1},\xi)$ are scalar polynomials in $q^{-1}$ with the same orders as $A(q^{-1},\theta)$, $B(q^{-1},\theta)$, $C(q^{-1},\theta)$, respectively

$$\begin{array}{l} A_\triangle(q^{-1},t,\xi) = f_{1,t}q^{-1} + ... + f_{n,t}q^{-n} \\ B_\triangle(q^{-1},t,\xi) = f_{(n+1),t}q^{-1} + \cdots + f_{(n+m),t}q^{-m} \\ C_\triangle(q^{-1},\xi) = \delta_{c_1}q^{-1} + ... + \delta_{c_l}q^{-l} \end{array} \tag{4.30}$$

where $\delta_{c_i}$'s are constant parameters, while $f_{i,t}$'s are nonlinear functions of input-output variables. Now according to the idea of hybrid quasi-ARMAX modeling [37], $f_{i,t}$ will be realized using certain NNMs described by

$$f_{i,t} = \sum_{j=1}^{M} \omega_{ij}\mathcal{N}_f(p_j,\varphi(t)) \quad (i = 1,...,r;\ r = n+m) \tag{4.31}$$

where $\mathcal{N}_f(x)$ is the 'basis function', $\omega_{ij}$'s are the coordinate parameters to be estimated, $\boldsymbol{p}_j$'s are the scale and position parameter vector to be determined using knowledge information, and $\varphi(t) = [y(t-1) \ldots y(t-n)\ u(t-1) \ldots u(t-m)]^T$ is the regression vector.

Introduce parameter vectors $\theta$ and $\xi$ defined as

$$\begin{aligned} \theta &= [a_1 \ldots a_n\ b_1 \ldots b_m\ c_1 \ldots c_l]^T \\ \xi &= [\omega_{11} \ldots \omega_{n1} \ldots \omega_{r1} \ldots \omega_{rM}\ \delta_{c_1} \ldots \delta_{c_l}]^T. \end{aligned} \tag{4.32}$$

Then it can be shown that the vector $\theta$ is a redundant parameter vector of the model $\mathcal{M}\{\theta, \xi\}$. Furthermore, introducing the following notations

$$G = B(q^{-1},\theta)/A(q^{-1},\theta) \quad H = (C(q^{-1},\theta) + C_{\triangle}(q^{-1},t,\xi))/A(q^{-1},\theta) \tag{4.33}$$

$$\triangle G_t = \frac{-A_{\triangle}(q^{-1},t,\xi)y(t) + B_{\triangle}(q^{-1},t,\xi)u(t)}{A(q^{-1},\theta)u(t)} \tag{4.34}$$

we may have a model of the form

$$\mathcal{M}\{\theta,\xi\} \quad y(t) = Gu(t) + \triangle G_t u(t) + He(t) \tag{4.35}$$

Note that (4.12) and (4.26) or (4.35) are similar in the forms. However, $\triangle G(q^{-1})$ in (4.12) is a rational function in the backward shift operator $q^{-1}$, while $\triangle G_t$ in (4.26) or (4.35) is only a function in time $t$. This means that the former describes the unmodeled dynamics in the sense of model structure, while the latter describes the modeling error in the sense of model output.

### (2) Estimation of $\theta$ and $\xi$

The $\theta$ and $\xi$ are two parameter vectors of the *complex model* $\mathcal{M}\{\theta,\xi\}$. As discussed above, the identification should be done such that the estimates $\hat{\theta}$, $\hat{\xi}$ have the interpretation that $\hat{\theta}$ describes the linear ARMAX model, while $\hat{\xi}$ describes the unmodeled dynamics.

Note the fact that the vector $\theta$ is a *redundant parameter vector* of the *complex model* $\mathcal{M}\{\theta,\xi\}$, that is, for any given $\theta$, $\xi$ and $\theta_0$, there exists a $\xi'$ that $\mathcal{M}\{\theta,\xi\} \equiv \mathcal{M}\{\theta_0,\xi'\}$ holds. Therefore, the identification can be implemented in the following two steps.

**Step 1:** *Estimation of $\theta$*

Let $\xi = 0$, then the models (4.26) and (4.35) become

$$A(q^{-1},\theta)y(t) = B(q^{-1},\theta)u(t) + C(q^{-1},\theta)e(t). \tag{4.36}$$

which is the same as $\mathcal{M}\{\theta\}$. The estimate $\hat{\theta}$ can thus be obtained by matching this linear ARMAX model to the input-output data of system.

**Step 2:** *Estimation of $\xi$*

Set $\theta = \hat{\theta}$ as constant, and use $\xi = 0$ as initial value. The vector $\xi$ is estimated by applying prediction error method (PEM) to the model (4.26) or (4.35).

Although the models (4.26), (4.35) and (4.36) are linear in the parameters to be estimated, the criteria in the identifications are not always unimodal because there exists MA noise model $C(q^{-1},\theta)e(t)$. A hybrid identification method using genetic algorithm [38] may be employed, if necessary, to prevent the estimations to be stuck at local minima.

## 4.3 KDI Analysis

As summarized in Appendix D, Kumamaru and his coworkers have developed a fault detection scheme based on the use of KDI for model discrimination [93, 61, 60]. However, the scheme was developed only for linear model without model uncertainty. In this section, we will generalize the scheme to the models containing model uncertainties.

Assume that the data from system are available from two distinct time intervals $I_1$ and $I_2$, see Fig. 4.4, in which $N_i$ $(i = 1, 2)$ denotes the number of data points in the interval $I_i$, while $X_i$ denotes the

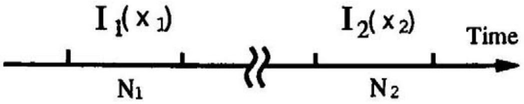


Figure 4.4: Two distinct time intervals

experimental condition. We estimate the unknown parameter vectors specifying the model and the error description, $\{\theta, \xi\}$, using the data obtained from the two distinct intervals $I_1$ and $I_2$, and denote the estimates by $\{\hat{\theta}_1, \hat{\xi}_1\}$ and $\{\hat{\theta}_2, \hat{\xi}_2\}$ respectively. The KDI is then used to detect the distortion of identified models $\mathcal{M}\{\hat{\theta}_1, \hat{\xi}_1\}$ and $\mathcal{M}\{\hat{\theta}_2, \hat{\xi}_2\}$.

Applying the KDI to the likelihood functions of identified models, $p(Y_{k+1}|\hat{\theta}_i, \hat{\xi}_i, U_k)$ $(i = 1,\ 2)$, we get

$$I_{k+1}[1,2] = \int p(Y_{k+1}|\hat{\theta}_1, \hat{\xi}_1, U_k) \log \frac{p(Y_{k+1}|\hat{\theta}_1, \hat{\xi}_1, U_k)}{p(Y_{k+1}|\hat{\theta}_2, \hat{\xi}_2, U_k)} dY_{k+1} \tag{4.37}$$

where $U_k = [u(1)\ u(2)\ \ldots\ u(k)]^T$ and $Y_{k+1} = [y(1)\ y(2)\ \ldots\ y(k+1)]^T$ are the input-output data sets from the interval $I_1$.

Based on the Bayesian rule

$$p(Y_{k+1}|\hat{\theta}_i, \hat{\xi}_i, U_k) = p(y(k+1)|\hat{\theta}_i, \hat{\xi}_i, Y_k, U_k) p(Y_k|\hat{\theta}_i, \hat{\xi}_i, U_k), \tag{4.38}$$

we can express (4.37) as

$$I_{k+1}[1,2] = E_{1k} J_{k+1} + I_k[1,2] \tag{4.39}$$

where

$$E_{1k} J_{k+1} = \int p(Y_k|\hat{\theta}_1, \hat{\xi}_1, U_k) J_{k+1} dY_k \tag{4.40}$$

$$J_{k+1} = \int p(y(k+1)|\hat{\theta}_1, \hat{\xi}_1, Y_k, U_k) \log \frac{p(y(k+1)|\hat{\theta}_1, \hat{\xi}_1, Y_k, U_k)}{p(y(k+1)|\hat{\theta}_2, \hat{\xi}_2, Y_k, U_k)} dy(k+1) \tag{4.41}$$

Assume that the likelihood functions $p(Y_{k+1}|\hat{\theta}_i, \hat{\xi}_i, U_k)$ are Gaussian distributed. Then the (4.40) can be further developed into an explicit form, by which the KDI can be evaluated in a feasible way for finite but fairly large data set [61]. However the developed algorithms will be different depending on the modeling error descriptions.

### 4.3.1 A Soft Bound Error Description Case

For simplicity, introduce the notations

$$G_i \sim G_i(q^{-1}, \hat{\Theta}_i), \quad H_i \sim H_i(q^{-1}, \hat{\Theta}_i), \quad Q_{i,k} \sim \psi(k)/A(q^{-1}, \hat{\theta}_i) \quad i = 1, 2 \tag{4.42}$$

Then we can transform the model (4.12) into the form

$$H_i^{-1} y(k) = H_i^{-1} G_i u(k) + H_i^{-1} Q_{i,k} \eta_i + e(k) \tag{4.43}$$

Considering that both $\eta$ and $e(k)$ are Gaussian stochastic process, we assume that

$$\nu_{i,k} \triangleq H_i^{-1} Q_{i,k} \eta_i + e(k) \tag{4.44}$$

and since $\eta$ and $e(k)$ are assumed to be independent, we get the prediction error $\hat{\Sigma}_i(k)$ given by

$$\begin{aligned} \hat{\Sigma}_i(k) &= E\left\{\nu_{i,k}^2\right\} \\ &= E\left\{H_i^{-1} Q_{i,k} \eta_i \eta_i{}^T Q_{i,k}^T H_i^{-1}\right\} + \hat{\sigma}_i^2 \\ &= H_i^{-1} Q_{i,k} \hat{C}_{\eta_i} Q_{i,k}^T H_i^{-1} + \hat{\sigma}_i^2 \end{aligned} \tag{4.45}$$

and the one-step prediction $m_{i,k} = \hat{y}(k+1|k)$ given by

$$m_{i,k} = (1 - H_i^{-1})y(k+1) + H_i^{-1}G_i u(k+1). \tag{4.46}$$

It therefore follows that the likelihood functions of the identified models in this case is given as

$$p(y(k+1)|\hat{\theta}_i, \hat{\xi}_i, Y_k, U_k) \in N(m_{i,k}, \hat{\Sigma}_i(k+1)), \quad i = 1,\ 2 \tag{4.47}$$

Now using (4.47) in (4.41), we can get

$$\begin{aligned} J_{k+1} &= \frac{1}{2}\left(\hat{\Sigma}_1(k+1)/\hat{\Sigma}_2(k+1) - 1\right) \\ &\quad -\frac{1}{2}\log\left(\hat{\Sigma}_1(k+1)/\hat{\Sigma}_2(k+1)\right) + \frac{1}{2}\left\|m_{1,k} - m_{2,k}\right\|^2_{\hat{\Sigma}_2(k+1)} \end{aligned} \tag{4.48}$$

Assume then that the system in the time interval $I_1$ is under normal mode. Hence the model $\mathcal{M}\{\hat{\theta}_1, \hat{\xi}_1\}$ describes the system, that is

$$y(k+1) = G_1 u(k+1) + Q_{1,k+1}\eta_1 + H_1 e(k+1) \tag{4.49}$$

Further assume that the system operates in open loop. Then $u(k+1)$, $\eta_i$ and $e(k+1)$ can be assumed to be mutually independent.

Now using (4.49) in (4.46), we will have

$$\begin{aligned} & E_{1k}\left\|m_{1,k} - m_{2,k}\right\|^2_{\hat{\Sigma}_2(k+1)} \\ = \; & E_{1k}\left\|H_2^{-1}(G_1 - G_2)u(k+1) + (H_2^{-1}H_1 - 1)e(k+1)\right. \\ & \qquad\qquad \left. +(H_2^{-1} - H_1^{-1})Q_{1,k+1}\eta_1\right\|^2_{\hat{\Sigma}_2(k+1)} \\ = \; & \left\|H_2^{-1}(G_1 - G_2)u(k+1)\right\|^2_{\hat{\Sigma}_2(k+1)} + E_{1k}\left\|(H_2^{-1}H_1 - 1)e(k+1)\right\|^2_{\hat{\Sigma}_2(k+1)} \\ & \qquad\qquad +E_{1k}\left\|(H_2^{-1} - H_1^{-1})Q_{1,k+1}\eta_1\right\|^2_{\hat{\Sigma}_2(k+1)} \end{aligned} \tag{4.50}$$

With (4.50), we can thus express (4.40) as

$$\begin{aligned} E_{1k}J_{k+1} &= \frac{1}{2}\left(\hat{\Sigma}_1(k+1)/\hat{\Sigma}_2(k+1) - 1\right) - \frac{1}{2}\log\left(\hat{\Sigma}_1(k+1)/\hat{\Sigma}_2(k+1)\right) \\ &\quad +\frac{1}{2}\left\|H_2^{-1}(G_1 - G_2)u(k+1)\right\|^2_{\hat{\Sigma}_2(k+1)} \\ &\quad +\frac{1}{2}E_{1k}\left\|(H_2^{-1}H_1 - 1)e(k+1)\right\|^2_{\hat{\Sigma}_2(k+1)} \\ &\quad +\frac{1}{2}E_{1k}\left\|(H_2^{-1} - H_1^{-1})Q_{1,k+1}\eta_1\right\|^2_{\hat{\Sigma}_2(k+1)} \end{aligned} \tag{4.51}$$

⟨ *Summary* ⟩: For finite but fairly large data sets $N_1$, the KDI can therefore be calculated as

$$I_{N_1}[1,2] = \sum_{k=0}^{N_1-1} E_{ik}J_{k+1} + I_0[1,2] = \sum_{j=1}^{4} I_{N_1}^{(j)}[1,2] \tag{4.52}$$

where $I_0[1,2]$ is neglected and $I_{N_1}^{(j)}[1,2]$'s are given as:

$$I_{N_1}^{(1)}[1,2] = \frac{N_1}{2}\left(\hat{\Sigma}_1(k+1)/\hat{\Sigma}_2(k+1) - 1\right) - \frac{N_1}{2}\log\left(\hat{\Sigma}_1(k+1)/\hat{\Sigma}_2(k+1)\right) \tag{4.53}$$

$$I_{N_1}^{(2)}[1,2] = \frac{1}{2}\sum_{k=0}^{N_1-1}\left\|H_2^{-1}(G_1 - G_2)u(k+1)\right\|^2_{\hat{\Sigma}_2(k+1)} \tag{4.54}$$

$$I_{N_1}^{(3)}[1,2] = \frac{1}{2}\sum_{k=0}^{N_1-1} E_{1k}\left\|(H_2^{-1}H_1 - 1)e(k+1)\right\|^2_{\hat{\Sigma}_2(k+1)} \tag{4.55}$$

$$I_{N_1}^{(4)}[1,2] = \frac{1}{2}\sum_{k=0}^{N_1-1} E_{1k}\left\|(H_2^{-1} - H_1^{-1})Q_{1,k+1}\eta_1\right\|^2_{\hat{\Sigma}_2(k+1)} \tag{4.56}$$

Here (4.55) and (4.56) can be further expressed as:

$$\begin{aligned} I_{N_1}^{(3)}[1,2] = \frac{1}{2}\sum_{k=0}^{N_1-1} \Big\{ & \hat{\Sigma}_2^{-1}(k+1)\frac{1}{2\pi i} \\ & \oint \left(H_2^{-1}(z)H_1(z) - 1\right)\hat{\sigma}_1^2\left(H_1(z^{-1})H_2^{-1}(z^{-1}) - 1\right)\frac{dz}{z}\Big\} \end{aligned} \tag{4.57}$$

$$\begin{aligned} I_{N_1}^{(4)}[1,2] = \frac{1}{2}\sum_{k=0}^{N_1-1} \Big\{ & \hat{\Sigma}_2^{-1}(k+1)\frac{1}{2\pi i} \\ & \oint (H_2^{-1}(z) - H_1^{-1}(z))Q_{1,k+1}\hat{C}_{\eta 1}Q_{1,k+1}^T(H_2^{-1}(z^{-1}) - H_1^{-1}(z^{-1}))\frac{dz}{z}\Big\} \end{aligned} \tag{4.58}$$

Note that the extra term (4.58) is induced from unmodeled dynamics and is related mainly to the difference between $H_1$ and $H_2$. Moreover, in this case the unmodeled dynamics affects all terms in (4.53)-(4.58).

### 4.3.2 A Direct Error Description Case

Similar to the *soft bound* error description case, let us first convert the model (4.26) into the form

$$H_i^{-1}y(k) = H_i^{-1}G_iu(k) + H_i^{-1}\Delta G_{i,k}u(k) + e(k) \tag{4.59}$$

where

$$\begin{array}{ll} G_i \sim G(q^{-1},\hat{\theta}_i) & H_i \sim H(q^{-1},\hat{\theta}_i) \\ \Delta G_{i,k} \sim \Delta G_k/\theta = \hat{\theta}_i, \xi = \hat{\xi}_i & i = 1,2 \end{array} \tag{4.60}$$

From (4.59), we have the one-step prediction $m_{i,k} = \hat{y}(k+1|k)$ given as

$$m_{i,k} = (1 - H_i^{-1})y(k+1) + H_i^{-1}G_iu(k+1) + H_i^{-1}\Delta G_{i,k+1}u(k+1) \tag{4.61}$$

The likelihood function in this case is

$$p(y(k+1)|\hat{\theta}_i, \hat{\xi}_i, Y_k, U_k) \in N(m_{i,k}, \hat{\sigma}_i^2), \quad i = 1,\ 2. \tag{4.62}$$

Then using (4.62) in (4.41), we get

$$J_{k+1} = \frac{1}{2}\left((\hat{\sigma}_1^2/\hat{\sigma}_2^2 - 1) - \log(\hat{\sigma}_1^2/\hat{\sigma}_2^2)\right) + \frac{1}{2}\left\|m_{1,k} - m_{2,k}\right\|^2_{\hat{\sigma}_2^2} \tag{4.63}$$

Assume

$$y(k+1) = G_1u(k+1) + \Delta G_{1,k+1}u(k+1) + H_1e(k+1) \tag{4.64}$$

Then using (4.64) in (4.61) and noting that $u(k)$ and $e(k)$ are independent, we can approximately have

$$\begin{aligned} & E_{1k}\left\|m_{1,k} - m_{2,k}\right\|^2_{\hat{\sigma}_2^2} \\ & \doteq \left\|H_2^{-1}(G_1 - G_2)u(k+1)\right\|^2_{\hat{\sigma}_2^2} + E_{1k}\left\|(H_2^{-1}H_1 - 1)e(k+1)\right\|^2_{\hat{\sigma}_2^2} \\ & \quad +H_2^{-1}(2G_1 - 2G_2 + \Delta G_{1,k+1} - \Delta G_{2,k+1})u(k+1) \\ & \qquad\qquad H_2^{-1}(\Delta G_{1,k+1} - \Delta G_{2,k+1})u(k+1)/\hat{\sigma}_2^2 \end{aligned} \tag{4.65}$$

In the derivation of (4.65), for simplicity the terms $\Delta G_{i,k+1}$ $(i = 1, 2)$ are treated to be constant with respect to the operation of $E_{1k}$. Hence (4.40) can be expressed as

$$\begin{aligned} E_{1k}J_{k+1} &= \frac{1}{2}\left((\hat{\sigma}_1^2/\hat{\sigma}_2^2 - 1) - \log(\hat{\sigma}_1^2/\hat{\sigma}_2^2)\right) \\ &+\frac{1}{2}\left\|H_2^{-1}(G_1 - G_2)u(k+1)\right\|_{\hat{\sigma}_2^2}^2 \\ &+\frac{1}{2}E_{1k}\left\|(H_2^{-1}H_1 - 1)e(k+1)\right\|_{\hat{\sigma}_2^2}^2 \\ &+\frac{1}{2}H_2^{-1}(2G_1 - 2G_2 + \Delta G_{1,k+1} - \Delta G_{2,k+1})u(k+1) \\ &\qquad H_2^{-1}(\Delta G_{1,k+1} - \Delta G_{2,k+1})u(k+1)/\hat{\sigma}_2^2 \end{aligned} \tag{4.66}$$

⟨ *Summary* ⟩: Similar to (4.52), the KDI can be evaluated as

$$I_{N_1}[1,2] = \sum_{j=1}^{4} I_{N_1}^{(j)}[1,2] \tag{4.67}$$

where $I_{N_1}^{(j)}[1,2]$'s are given as

$$I_{N_1}^{(1)}[1,2] = \frac{N_1}{2}\left[(\hat{\sigma}_1^2/\hat{\sigma}_2^2 - 1) - \log(\hat{\sigma}_1^2/\hat{\sigma}_2^2)\right] \tag{4.68}$$

$$I_{N_1}^{(2)}[1,2] = \frac{1}{2}\sum_{k=0}^{N_1-1}\|H_2^{-1}(G_1 - G_2)u(k+1)\|_{\hat{\sigma}_2^{-2}}^2 \tag{4.69}$$

$$I_{N_1}^{(3)}[1,2] = \frac{1}{2}\sum_{k=0}^{N_1-1} E_{1k}\left\|(H_2^{-1}H_1 - 1)e(k+1)\right\|_{\hat{\sigma}_2^2}^2 \tag{4.70}$$

$$\begin{aligned} I_{N_1}^{(4)}[1,2] = \frac{1}{2}\sum_{k=0}^{N_1-1} & H_2^{-1}(2G_1 - 2G_2 + \Delta G_{1,k+1} - \Delta G_{2,k+1})u(k+1) \\ & H_2^{-1}(\Delta G_{1,k+1} - \Delta G_{2,k+1})u(k+1)/\hat{\sigma}_2^2. \end{aligned} \tag{4.71}$$

The equation (4.70) can be further expressed as

$$I_{N_1}^{(3)}[1,2] = \frac{N_1}{2}\left\{\frac{\hat{\sigma}_2^{-2}}{2\pi i}\oint\left(H_2^{-1}(z)H_1(z) - 1\right)\hat{\sigma}_1^2\left(H_1(z^{-1})H_2^{-1}(z^{-1}) - 1\right)\frac{dz}{z}\right\} \tag{4.72}$$

We notice that the first three terms (4.68)-(4.71) are the same as (D.10)-(D.12) of the idea case. Furthermore, the contribution of the unmodeled dynamics $\Delta G_{i,k+1}$ gives a new term (4.71), and it does not affect the other terms. We may therefore use the first three terms as fault detection indexes in the thresholding approach and use the fourth term for evaluating the effect of the modeling error on the KDI.

### Asymptotic Properties

As summarized in Appendix D, in the ideal case where there is no model uncertainty, the asymptotic distribution of analyzed KDI terms is $\chi^2$-distributed, which provides useful information for selecting threshold value in the fault detection application. However, in the case where there is model uncertainty, analysis for the asymptotic properties becomes difficult. Further research is needed for this open problem.

### 4.3.3 Indexes for Evaluating Modeling Error

As shown in Fig. 4.1, in order to achieve a robust fault detection by using a decision making scheme, indexes for evaluating modeling error are very important. We here will introduce several such indexes based on the results of the KDI analysis.

**(1) Indexes of unmodeled dynamics**

Let us first take a look at the results of the KDI analysis in a *soft bound* error description case. From (4.45), we can see that the prediction error $\hat{\Sigma}_i(k)$ includes two parts. Besides the contribution of disturbance $\hat{\sigma}_i^2$, it contains a part contributed from unmodeled dynamics

$$\Delta\hat{\Sigma}_i(k) = H_i^{-1} Q_{i,k} \hat{C}_{\eta_i} Q_{i,k}^T H_i^{-1} \tag{4.73}$$

This extra part is a function of input-output variables. We introduce the following index for unmodeled dynamics by averaging the term $\Delta\hat{\Sigma}_i(k)$ over the time interval $I_i$, from which $\{\hat{\theta}_i, \hat{\xi}_i\}$ is estimated

$$I_{C_\eta}[i] = \frac{1}{N_i}\sum_{k=1}^{N} \Delta\hat{\Sigma}_i(k) = \frac{1}{N_i}\sum_{k=1}^{N_i}\left(H_i^{-1} Q_{i,k} \hat{C}_{\eta i} Q_{i,k}^T H_i^{-1}\right). \tag{4.74}$$

From the expressions of $H_i$ and $Q_{i,k}$, we know that this index only depends on the unmodeled dynamics and is not related with $\hat{\theta}_i$.

Next we inspect the results of the KDI analysis in the *direct* error description case, and introduce two indexes of unmodeled dynamics. The first one is based on the fourth term of the KDI, (4.71). Since the unmodeled dynamics $\Delta G_{i,k}$ is rather sensitive to the noise realization, $I_{N_1}^{(4)}[1,2]$ is too sensitive to noise when it is directly used as an index of unmodeled dynamics. In order to derive a less sensitive index, we consider the case where the fault detection is implemented on-line. The KDIs (4.67)-(4.71) are calculated on-line for each estimated $\hat{\theta}_2(t)$ and $\hat{\xi}_2(t)$, namely we compute $I_{N_1}[1,2:t]$, $I_{N_1}^{(1)}[1,2:t]$, $I_{N_1}^{(2)}[1,2:t]$, $I_{N_1}^{(3)}[1,2:t]$, $I_{N_1}^{(4)}[1,2:t]$ at time $t$. Then the mean value of $I_{N_1}^{(4)}[1,2:t]$ in a moving window of length $\mathcal{W}$ can be used as an index of the unmodeled dynamics

$$I_{\Delta G}^{(1)}(t) = \frac{1}{\mathcal{W}}\sum_{j=t-\mathcal{W}}^{t} I_{N_1}^{(4)}[1,2:j] \tag{4.75}$$

Since in the direct error description approach, the unmodeled dynamics $\Delta G_t$ is identified directly. It is reasonable to use it as an index of unmodeled dynamics. Because the $\Delta G_t$ is a function of the past input-output of system, we introduce this second index for unmodeled dynamics using the mean value of $|\Delta G_k(\hat{\theta}_2(t), \hat{\xi}_2(t))|$; $k = 0, ..., N_1 - 1$ calculated from the input-output data sets in the internal $I_1$.

$$I_{\Delta G}^{(2)}(t) = \frac{1}{N_1}\sum_{k=0}^{N_1-1} |\Delta G_k(\hat{\theta}_2(t), \hat{\xi}_2(t))| \tag{4.76}$$

**(2) Interpretations of the indexes**

The three indexes for unmodeled dynamics have different interpretation:

- $I_{C_\eta}$ denotes the contribution of modeling error on the prediction error. Since in the soft bound approach, the modeling error is treated in the same way as the disturbance, the effect of unmodeled dynamics can be evaluated by its contribution to the prediction error. The index depends on the unmodeled dynamics only and is not related with $\theta$.

- $I_{\Delta G}^{(1)}$ expresses the effect of unmodeled dynamics on the KDI. Since the fault detection is realized based on the KDI, this index of unmodeled dynamics is very useful. It depends on both the unmodeled dynamics and the difference of the estimates $\hat{\theta}_i$ $(i = 1, 2)$.

- $I_{\Delta G}^{(2)}$ describes the size of unmodeled dynamics directly.

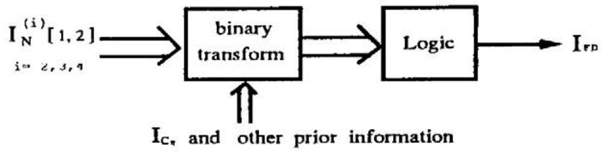


Figure 4.5: An example of robust fault detection scheme

## 4.4 Fault Decision Scheme

In the ideal case where no unmodeled dynamics exists, the fault detection can be executed by applying the thresholding approach to the KDI, see (D.17). In the presence of unmodeled dynamics, the identified model parameters may change depending on the noise realization, resulting in a fluctuation of the calculated KDI. When the unmodeled dynamics and the noise are large, the fluctuation might be so large that the method of simply applying the thresholding approach to the KDI becomes infeasible. Therefore, in order to realize a robust fault detection, robust decision making schemes should be developed so that the indexes of the KDI and the indexes of unmodeled dynamics can be used effectively.

### Scheme 1

It should be noticed that the identified model parameters have different sensitivity to the noise. For example, it is found that the fluctuations of $\hat{a}_i$ are typically not so large as those of $\hat{b}_i$. On the other hand, the KDI consists of four different terms, each of which has its particular relationship with the model parameters. Therefore, if a scheme can be developed to use the information of the four KDI terms and the index of unmodeled dynamics effectively, it is possible to achieve a robust fault detection. As a choice for such scheme, Neural Network approach can be considered. However, only a simple logic is used here as an example to show the possibility.

Considering that $I_{N_1}^{(1)}[1,2]$, which expresses the difference between the prediction errors, is not related directly to the system failure, we only use $I_{N_1}^{(j)}[1,2]$ $(j = 2,3,4)$ to develop the robust decision making scheme. Introduce a binary transform function as followings

$$f_j = \mathcal{T}_j\left(I_{N_1}^{(j)}[1,2]\right) = \begin{cases} 1 & \text{when } I_{N_1}^{(j)} > h_j \\ 0 & \text{when } I_{N_1}^{(j)} < h_j \end{cases} \tag{4.77}$$

$$j = 2,3,4$$

where $h_j$'s $(j = 2,3,4)$ are thresholds which may be determined by using the index of unmodeled dynamics $I_{C_\eta}$ and other prior information (variance of disturbance, etc.) (see Fig. 4.5), which is though still an open problem to be solved. Then the fault detection index $I_{FD}$ can be obtained by applying a logic to the $f_i$'s

$$I_{FD} = \mathcal{D}(f_i,\ i = 2,3,4) \tag{4.78}$$

where $\mathcal{D}$ may simply be a majority voting rule.

### Scheme 2

Let us consider the case where the fault detection is implemented on-line. The KDI $I_{N_1}[1,2:t]$ will be calculated for each identified $\hat{\theta}_2(t)$ and $\hat{\xi}_2(t)$. Then a robust decision making scheme may be obtained based on statistical test of $I_{N_1}[1,2:t]$. Because the fourth term $I_{N_1}^{(4)}[1,2:t]$ is is rather sensitive to noise realization, we will construct the fault detection index by averaging the sum of first three terms of the KDI

$$\bar{I}_{N_1}[1,2:t] = \frac{1}{\mathcal{L}} \sum_{i=t-\mathcal{L}}^{t} \left(I_{N_1}^{(1)}[1,2:i] + I_{N_1}^{(2)}[1,2:i] + I_{N_1}^{(3)}[1,2:i]\right) \tag{4.79}$$

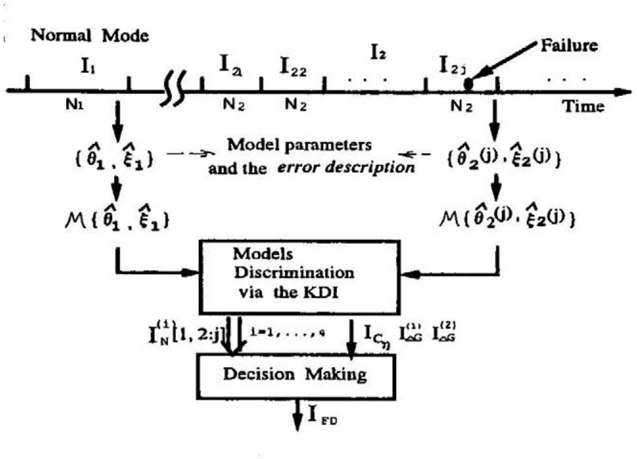

Normal Mode
$I_1$
$I_{21}$
$I_{22}$
$I_2$
$I_{2j}$
Failure
$N_1$
$N_2$
$N_2$
$N_2$
Time
$\{\hat{\theta}_1, \hat{\xi}_1\}$
Model parameters
and the *error description*
$\{\hat{\theta}_2(j), \hat{\xi}_2(j)\}$
$\mathcal{M}\{\hat{\theta}_1, \hat{\xi}_1\}$
$\mathcal{M}\{\hat{\theta}_2(j), \hat{\xi}_2(j)\}$
Models Discrimination via the KDI
$I_N^{(i)}[1, 2{:}j]$
$i=1,\ldots,q$
$I_{C_\eta}$ $I_{\Delta G}^{(1)}$ $I_{\Delta G}^{(2)}$
Decision Making
$I_{FD}$

Figure 4.6: Implementation of the FDS in a batch way

and execute fault detection by the following thresholding approach

$$\bar{I}_{N_1}[1, 2:t] \gtrless \eta \Longrightarrow \begin{cases} \text{fault} \\ \text{no fault} \end{cases} \tag{4.80}$$

In order to realize robustness of the fault detection, the window $\mathcal{L}$ and the threshold $\eta$ may be appropriately determined based on the evaluated indexes of unmodeled dynamics and by considering other information about the system such as variance of disturbance $\hat{\sigma}^2$. For instance, when the calculations $I_{\Delta G}^{(1)}(t)$ and $I_{\Delta G}^{(2)}(t)$ indicate that the modeling error is rather large, $\mathcal{L}$ shall be chosen larger while $\eta$ be taken to be smaller. This however is an intuitive strategy for the determination of $\mathcal{L}$ and $\eta$. Further investigation should be done in order to realize robustness in the model-based fault detection using the evaluation of modeling error.

### Fault Decision

It should be pointed out that the decisions made by (D.17), (4.78) or (4.80) are only for system parameter changes. Strictly speaking, all system parameter changes do not mean the occurrence of a fault. For example in an adaptive control system, some parameter variations can appear naturally. It therefore is important to make fault decision after a system change has been detected.

Usually, fault decision can be realized based on relationships between model parameter $\theta$ and physical parameter $\mathcal{P}$, $\theta = f(\mathcal{P})$. However, the function $f(\cdot)$ is usually unknown and may be so complex that it is impossible to express $\mathcal{P} = f^{-1}(\theta)$ using an explicit function. To solve this problem, some knowledge-based approaches have been proposed, in which $f^{-1}(\cdot)$ is described using Fuzzy Inference or Neural Network approach, see e.g. [59, 58].

## 4.5 Numerical Simulations

In this section, we will carry out several numerical simulations to demonstrate the effectiveness of the methods discussed in the previous sections.

### 4.5.1 Implementation of the FDS

Figure 4.6 and 4.7 show the implementations of the fault detection system (FDS) for a batch fault detection and an on-line fault detection, respectively. All of the methods discussed in the previous sections except the *soft bound* approach can be easily implemented in an on-line way. However, when a *soft bound* approach is used, the identification can only be implemented in a limited on-line way,

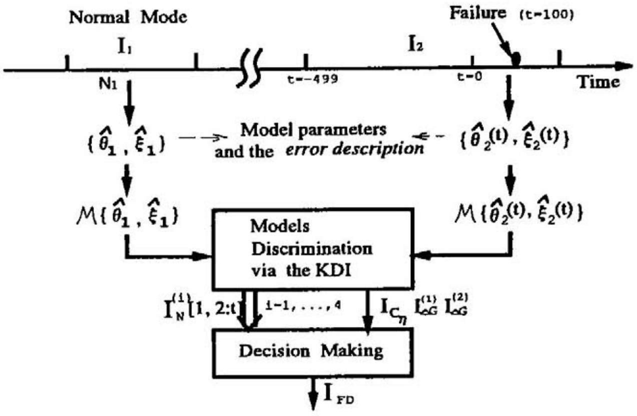


Figure 4.7: Implementation of the FDS in an on-line way

i.e, for each estimated $\hat{\theta}$, the vector $\xi$ is estimated using a batch identification method, because only a batch algorithm is so far available for the estimation of the *soft bound* of the unmodeled dynamics. Since when the orders of a model is determined for time-invariant system, the unmodeled dynamics does not change via time, it does need to be estimated repeatedly on-line. Therefore in the robust fault detection using parameter estimation, the system model are repeatedly estimated on-line in practice. But in the simulations, we will estimate the unmodeled dynamics for each estimate $\hat{\theta}$.

The data from the system are available from two distinct intervals $I_1$ and $I_2$. It is assumed that the system is under normal mode in the interval $I_1$, and is to be monitored in the interval $I_2$. Now from $I_1$ the unknown parameter vector $\{\theta, \xi\}$ is estimated as $\{\hat{\theta}_1, \hat{\xi}_1\}$, and from $I_2$ it is estimated as $\{\hat{\theta}_2(t), \hat{\xi}_2(t)\}$. If a fault occurs in the interval $I_2$, the effect will be reflected as a difference between the identified models $\mathcal{M}\{\hat{\theta}_1, \hat{\xi}_1\}$ and $\mathcal{M}\{\hat{\theta}_2(t), \hat{\xi}_2(t)\}$. The discrimination of two identified models is performed via the KDI and the robust fault detection will be realized using the fault decision schemes.

### 4.5.2 Example 1

In practical application, an identification is realized using finite data sets. The noise in the data will cause a variation between two identified models. When there exists no unmodeled dynamics, the variation is usually very small. In the case of presence of unmodeled dynamics, the identification becomes sensitive to noise realization. As the unmodeled dynamics increases, the variation may become so large that a model discrimination based FDS may become infeasible. The first example is designed to show how unmodeled dynamics affects KDI-based FDS.

The third order SISO system described by (4.81) is considered as a true system

$$G(s) = \frac{ke^{-\tau s}}{s(T_0 s + 1)(T_1 s + 1) + k} \tag{4.81}$$

where $k = 2.5$, $T_0 = 0.1$, $T_1 = 0.5$. After being sampled with a period of 0.25, the system is corrupted by a white Gaussian noise sequence $e(t) \in N(0, \sigma_0^2)$. The input of the system is PRBS (pseudo-random binary sequence).

We will apply the KDI-based robust FDS with *soft bound* error description to this example. The model used to identify the system is therefore described by (4.6) and (4.7) where the degrees of polynomials $A$, $B$, and $C$ are chosen to be $n = 2$, $m = 2$ and $l = 0$, respectively.

We will implement this example in a batch way. The number of data sets in the interval $I_1$ is $N_1 = 1000$, while $N_s \times N_2 = 200 \times 1000$ in the interval $I_2$. The interval $I_2$ is divided into $N_s$ equal parts labeled by $I_{2j}$, $(j = 1, ..., N_s)$, see Fig. 4.6. The fault mode due to physical parameter jump change, in which the physical parameters changed from their normal value to $k = 0.5$, $T_0 = 0.05$, and $T_1 = 0.2$, is assumed to occur in the 100th part of the interval $I_1$.

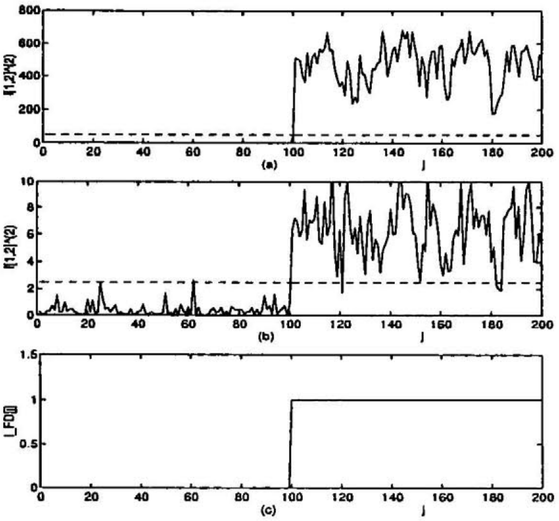


Figure 4.8: The results of case 1 (a) the second term of KDI for $\sigma_0^2 = 0.01$; (b) the second term of KDI for $\sigma_0^2 = 1.00$; (c) the output of the robust decision making scheme

By using the methods described in Section 4.3.2, from $I_1$ and $I_{2j}$ the unknown parameter vector $\{\theta, \xi\}$ can be estimated as $\{\hat{\theta}_1, \hat{\xi}_1\}$ and $\{\hat{\theta}_2(j), \hat{\xi}_2(j)\}$ respectively. The terms of the KDI, $I_{N_1}^{(i)}[1, 2 : j]$, $(i = 1, ..., 4;\ j = 1, ..., 200)$ and the KDI, $I_{N_1}[1, 2j]$ can be calculated by using (4.52)-(4.58). Finally, the fault detection index $I_{FD}$ can calculated by using (4.77) and (4.78).

[Case 1, $\tau = 0$]: This is a case where the unmodeled dynamics is small. The simulations are carried out for various value of noise variance $\sigma_0^2$ ranging from 0.01 to 1.00. The simulation results of this case are shown in Fig. 4.8. For simplicity, we only show $I_{FD}$ and the second term of KDI which expresses the difference between system models $G_1$ and $G_2$.

Figure 4.8(a) and (b) show the outputs of the second term of KDI, $I_{N_1}^{(2)}[1, 2 : j]$ for $\sigma_0^2 = 0.01$ and $\sigma_0^2 = 1.0$ respectively. We can see that the fluctuation of the KDI increases as the noise increases. the fluctuation of the KDI might become so large that the thresholding approach becomes infeasible. Figure 4.8(c) shows the output of the robust decision making scheme $I_{FD}$ calculated by using (4.77) and (4.78) where $h_2$, $h_3$, and $h_4$ were determined based on the index of unmodeled dynamics $I_{C_\eta}$ and other prior information such as the variance of the noise. In this case, $h_2 = 2.5$, $h_3 = 10$, and $h_4 = 0.5$ were chosen for $\sigma_0^2 = 1.0$. We can see that the decision making scheme responds the system fault correctly.

For $\sigma_0^2 = 0.01$, the estimates of the parameters $\theta$, $\xi$ and $I_{C_\eta}$ from the interval $I_1$ were

$$\begin{aligned}
\hat{\theta}_1 &= [a_1,\ a_2,\ b_1,\ b_2] \\
&= [-1.5103,\ 0.7428,\ 0.0659,\ 0.1433] \\
\hat{\xi}_1 &= [\alpha_a,\ \lambda_a,\ \alpha_b,\ \lambda_b,\ \sigma^2] \\
&= [0.0094,\ 0.1475,\ 0.0134,\ 0.0115,\ 0.0121] \\
I_{C_{\eta 1}} &= 0.0013.
\end{aligned}$$

[Case 2, $\tau = 0.25$]: This is a case where there exists rather large unmodeled dynamics since a time delay $\tau = 0.25$ was added to the system. The variance of the noise $\sigma_0^2$ simulated in this case varied from 0.001 to 0.5. The simulation results are shown in the Fig. 4.9, where $h_2 = 0.5$, $h_3 = 10$, and $h_4 = 0.5$ were chosen for $\sigma_0^2 = 0.50$. We can see that the output of the KDI is more sensitive to the noise in this case. However, the decision making scheme responded the system failure correctly again.

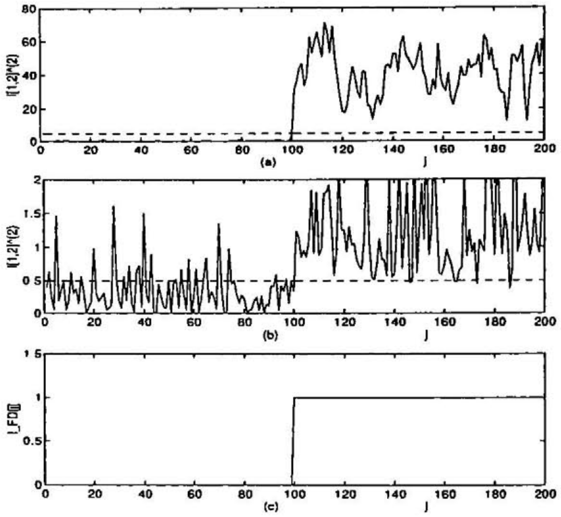


Figure 4.9: The results of case 2 (a) the second term of KDI for $\sigma_0^2 = 0.01$; (b) the second term of KDI for $\sigma_0^2 = 0.50$; (c) the output of the robust decision making scheme

For $\sigma_0^2 = 0.01$, the estimated parameters $\theta$, $\xi$ and $I_{C_\eta}$ from the interval $I_1$ in this case were

$$\begin{aligned} \hat{\theta}_1 &= [a_1,\ a_2,\ b_1,\ b_2] \\ &= [-1.5868,\ 0.8166,\ -0.0049,\ 0.0644] \\ \hat{\xi}_1 &= [\alpha_a,\ \lambda_a,\ \alpha_b,\ \lambda_b,\ \sigma^2] \\ &= [0.1379,\ 0.3498,\ 0.0092,\ 0.1358,\ 0.0121] \\ I_{C_{\eta 1}} &= 0.0341. \end{aligned}$$

Note that in this case, $I_{C_{\eta 1}}$ increased obviously, which indicates that there exists larger unmodeled dynamics. This simulation result shows that $I_{C_{\eta j}}$ defined as (4.74) is a nice index for evaluating unmodeled dynamics.

Two phenomena are obvious based on the results of this example. (1) The fluctuation of the KDI increases as the unmodeled dynamics increases (compare Fig. 4.8(a)(b) with Fig. 4.8(a)(b)). (2) The fluctuation increases as the variance of the noise increases (compare (a) and (b) in Fig. 4.8 or Fig. 4.9). Both of them obscure the performance of the KDI-based FDS.

### 4.5.3 Example 2

The example 1 also shows the KDI-based robust fault detection scheme with *soft bound* error description is effective. However, there are still two open problems to be solved. One is that an on-line algorithm is not so far available for the estimation of *soft bound.* The second is the existing batch algorithm is rather time-consuming.

The following example is designed to test the KDI-based robust fault detection schemes with *direct description* of modeling error.

The system considered is described by (4.81), where the parameters $k$, $T_0$ and $T_1$ are given the same values as those in the example 1. After being sampled with a period of 0.25, the system is corrupted by a correlated noise sequence $\{\nu_t\}$

$$\nu_t = (1 - 1.2q^{-1} + 0.5q^{-2})e(t) \tag{4.82}$$

where $e(t) \in N(0, 0.50)$ is white Gaussian noise.

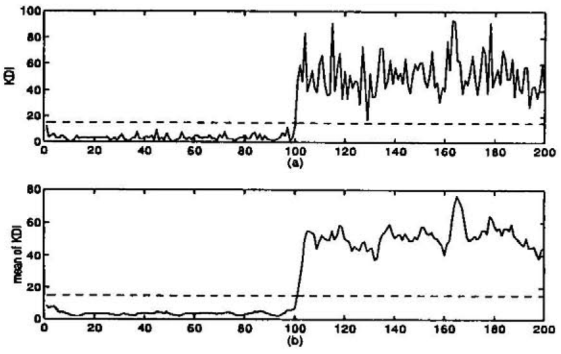


Figure 4.10: The results of example 2 when $\tau = 0$: (a) the sum of the first three terms of KDI; (b) the fault detection index $\bar{I}_N[1, t]$.

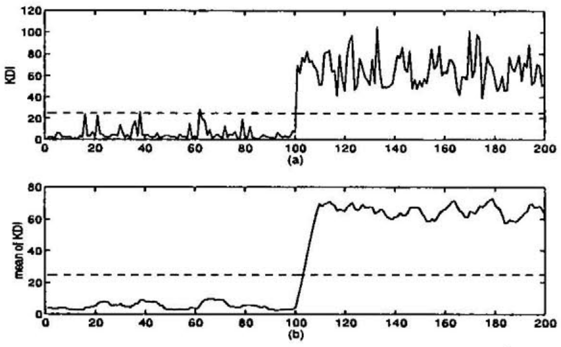


Figure 4.11: The results of example 2 when $\tau = 0.25$: (a) the sum of the first three terms of KDI; (b) the fault detection index $\bar{I}_N[1, t]$.

For identification, we use the model described by (4.26), where $n_d = 5$, $m_d = 5$, $l = 2$ are chosen for $\xi$ and $n = 2$, $m = 2$, $l = 2$ for $\theta$.

The simulation is implemented on-line, see Fig. 4.7. The interval $I_1$ contains 500 data sets, from which $\{\hat{\theta}_1, \hat{\xi}_1\}$ is estimated, while the interval $I_2$ contains 700 data sets, from which $\{\hat{\theta}_2(t), \hat{\xi}_2(t)\}$ $(t = 0, ..., 200)$ is estimated on-line. The initial values $\{\hat{\theta}_2(0), \hat{\xi}_2(0)\}$ for the on-line estimation are obtained from the first 500 data sets. The fault mode which is the same as that of example 1 is assumed to occur at $t = 100$. During the estimation, $\theta$ is updated for one step when one pair of new data observed, while $\xi$ is updated for 50 steps, for which the past data are also used.

Two simulations were done with $\tau = 0$ and $\tau = 0.25$ respectively, in which the output of KDI, $I_{N_1}[1, 2 : t]$ is calculated from (4.67)-(4.71). The output of fault detection index $\bar{I}_N[1, t]$ is calculated from (4.79) with $\mathcal{L} = 5$ for the case of $\tau = 0$ and $\mathcal{L} = 10$ for $\tau = 0.25$. And the indexes of unmodeled dynamics $I^{(1)}_{\triangle G}(t)$ and $I^{(2)}_{\triangle G}(t)$ are calculated from (4.75) with $\mathcal{W} = 60$ and (4.76), respectively. Figure 4.10 and 4.11 show the results of $I_{N_1}[1, 2 : t]$ and $\bar{I}_{N_1}[1, 2 : t]$ for the cases of $\tau = 0$ and $\tau = 0.25$. Figure 4.12 shows $I^{(1)}_{\triangle G}(t)$ and $I^{(2)}_{\triangle G}(t)$ of the two cases. The indexes for unmodeled dynamics may change when system parameter changes, so we do not show the values after the fault occurred. We can see that in the case where $\tau = 0.25$ the unmodeled dynamics is larger.

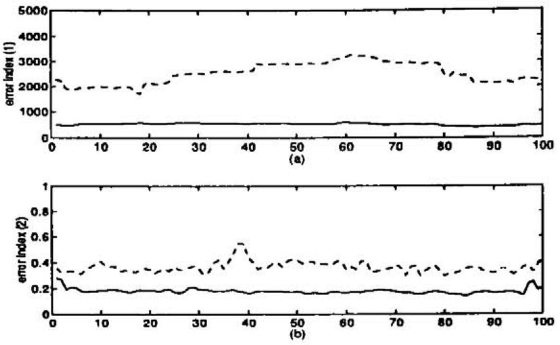


Figure 4.12: The outputs of the index for unmodeled dynamics in example 2, in which solid line is the result when $\tau = 0$ and dashed-line is the result when $\tau = 0.25$: (a) the index of unmodeled dynamics $I^{(1)}_{\triangle G}(t)$; (b) the index of unmodeled dynamics $I^{(2)}_{\triangle G}(t)$.

## 4.6 Discussions and Conclusions

Since an exact mathematical model is impossible in practice, it is a challenging problem to increase the robustness of a model-based fault detection method with respect to the model uncertainty. Typically, the KDI-based fault detection scheme was based on the assumption that the model parameterization is chosen adequately for the system. In order to make the scheme practicable, it is crucial to relax the assumption, i.e., to extend the scheme to the case where there is model uncertainty. In this chapter, we have realized such an extension by using robust identification techniques and fault decision scheme.

In the literature, the term 'robust identification' mainly means identification for robust control. There rarely exists publications discussing robust identification method for system analysis. In this chapter, we suggest two robust identification methods for fault detection by following the basic ideas of existing approaches. One is based on Goodwin's stochastic embedding approach, in which the unmodeled dynamics is characterized as a random quantity with *soft bound.* The second is modified from the *hard bound* error description approaches, in which the unmodeled dynamics is partly described directly by a deterministic model and partly taken account into the noise.

Two analyses have been made for the KDI. First, the KDI is analyzed into an explicit form with incorporating the *soft bound* error description. The analyzed KDI consists of four terms, each of which, however, contains the effect of the modeling error. Second, with incorporating the *direct description* of modeling error, the KDI is also analyzed into an explicit form consisting of four terms. In the second case, the first three terms are the same as that of ideal case, while the modeling error is only affected the fourth terms. Furthermore, three indexes for evaluating the size or the effects of unmodeled dynamics have been introduced based on the results of the KDI analysis.

Finally, two fault decision scheme have been developed for robustness realization, in which the fault detection is done on the basis of the index of KDI, the index of unmodeled dynamics and other information. The effectiveness of the proposed schemes has been confirmed through numerical simulations.

On the other hand, there are still several problems needed to solve in the future research. First, we have obtained an extra term in the KDI analysis. Theoretical analysis on the new term is needed in order to use the term efficiently. Second, Theoretical and experimental researches are needed to find out what information the individual KDI term may provide and how to use it in fault detection and fault analysis. Finally, further investigations are needed to find an more efficient algorithm for thresholding decision based on the evaluation of unmodeled dynamics.

# Chapter 5

# KDI-Based Fault Detection of Nonlinear Systems

## 5.1 Introduction

Most faults in dynamical systems can be represented as unexpected variations in system operation modes which are caused by changes in the system configuration parameters. Based on this assumption, Kumamaru and his coworker [61] have proposed a fault detection scheme using Kullback discrimination information (KDI) as the index for model discrimination. Two important assumption have so far been made in the scheme: (1) The system to be diagnosed is linear; (2) The model parameterization is chosen adequately for the system. These assumptions are never true in real applications. Therefore it is crucial and highly motivated to relax these assumptions. In Chapter 4, the assumption (2) has been relaxed by extending the fault detection scheme to a robust one, see also [55], in which the KDI is applied to discriminate the identified linear ARMAX model with model uncertainty, and the description of unmodeled dynamics is considered in the KDI analysis and the thresholding decision for robustness realization. The unmodeled dynamics has been characterized as a random quantity with *soft bound*, on the basis of Goodwin's stochastic embedding approach [25]. In this chapter, we will discuss how to relax the assumption (1), that is, how to extend the scheme to nonlinear systems.

For nonlinear systems, a linear mathematical model may not be accurate enough for model-based fault detection methods. Nonlinear models are required. Recently, many authors have suggested to use neural networks [83, 96, 106, 95, 74]. Neural networks are typically used in the two following ways in fault detection (system monitoring):

- Use neural networks as classifier, i.e., train neural networks to classify the observed data into healthy mode and faulty modes.
- Use neural networks as nonlinear black-box models of the monitored systems, i.e., train neural networks with data observed in different working modes and apply model based methods to the trained neural networks models.

These two approaches both require the availability of large amount of observed data of the monitored system, in the healthy mode and *in all the possible faulty modes*. However, it is rarely feasible to get sufficient data of all the possible faulty modes in real applications.

In this chapter, we will propose an alternative approach based on the global and local linear approximations of the monitored system. In order to describe general nonlinear systems by an input-output type model, we first propose a hybrid quasi-ARMAX model, which has a linear ARMAX structure with coefficients consisting of constant parameters and nonlinear terms of input-output variables. The nonlinear terms are represented using a group of certain nonlinear nonparametric models (NNMs). If all the NNMs used in the hybrid model have the same structure, it can be shown that the model constructed is equivalent to a hybrid model combining a linear ARMAX model and a multi-ARX-model. The multi-ARX-model consists of several local ARX models and its overall performance is obtained via an interpolation using the 'basis functions' in the NNM [37]. Furthermore, the model can be identified in such a way that its ARMAX model part gives a best linear approximation of the system, and the multi-ARX-model part gives the estimate of modeling error due to nonlinear undermodeling. With this feature, a robust fault detection scheme can be applied to the model for dealing with nonlinear systems [56]. However, in the case where the system properties of interest can not be described by a linear model, the robust fault detection like approach become infeasible. To solve this problem, we will use the hybrid quasi-ARMAX model as a multi-ARMAX-model consisting several local ARMAX models and realize the fault detection by applying the KDI to discriminate each identified local linear ARMAX model.

The chapter is organized as follows: Section 5.2 proposes a hybrid quasi-ARMAX modeling and identification scheme for general nonlinear systems. In Section 5.3, we show that the hybrid quasi-ARMAX model can be applied to fault detection in two ways: a robust fault detection like approach and a multi-model based approach. Section 5.4 discusses the details of the multi-model based fault detection scheme using the KDI to discriminate the identified local linear models. The detail of the robust fault detection like scheme is discussed in Section 5.5. Numerical simulations are carried out in Section 5.6. Finally, Section 5.7 is devoted to discussions and conclusions.

## 5.2 A Hybrid Quasi-ARMAX Modeling Scheme

In this section, we propose a hybrid quasi-ARMAX modeling scheme for identification of nonlinear systems.

### 5.2.1 Quasi-ARMAX Modeling

As the object to be diagnosed, let us consider a SISO general nonlinear ARX system described by

$$\mathcal{S}: \quad y(t) = g(\varphi(t)) + v(t) \tag{5.1}$$

$$v(t) \in N(0, \sigma_0^2) \tag{5.2}$$

$$\varphi(t) = [y(t-1) \ldots y(t-n) \; u(t-1) \ldots u(t-m)]^T \tag{5.3}$$

where $y(t)$ is the output at time $(t = 1, 2, \ldots)$, $u(t)$ the input, $\varphi(t)$ the regression vector, $v(t)$ the system disturbance, and $g(\,\cdot\,)$ the unknown continuously differentiable nonlinear function.

Based on the idea of quasi-ARMAX modeling [37], we use a quasi-ARMAX model described by (5.4) to identify the system

$$\mathcal{M}: \quad y(t) = \varphi^T(t)(\theta + \Delta\theta_t) + C(q^{-1})e(t) \tag{5.4}$$

$$e(t) \in N(0, \sigma^2) \tag{5.5}$$

where $e(t)$ is white Gaussian noise, and $\theta$, $\Delta\theta_t$, $C(q^{-1})$ are defined as

$$\left.\begin{array}{l} \theta = [a_1 \ldots a_n \; b_1 \ldots b_m]^T \\ \Delta\theta_t = [\Delta a_{1,t} \ldots \Delta a_{n,t} \; \Delta b_{1,t} \ldots \Delta b_{m,t}]^T \\ C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_l q^{-l} \end{array}\right\} \tag{5.6}$$

and where $a_i$, $b_i$ and $c_i$ are the constant parameters, while $\Delta a_{i,t}$ and $\Delta b_{i,t}$ are nonlinear functions of $\varphi(t)$, and $q^{-1}$ is backward shift operator, for instance, $q^{-1}u(t) = u(t-1)$.

### 5.2.2 Hybrid Quasi-ARMAX Model

For a certain type, for instance Kolmogorov-Gabor type (B.2), of nonlinear system, the nonlinear terms $\Delta a_{i,t}$ and $\Delta b_{i,t}$ can be expressed in explicit forms (B.6) and (B.7). However, for general nonlinear systems, these nonlinear terms may become rather complicated functions of $\varphi(t)$, and can not expressed in explicit forms. We will therefore represent them by using the NNMs as follows:

$$\left.\begin{array}{l} \Delta a_{i,t} = f_i(\varphi(t)) \quad (i = 1, \ldots, n) \\ \Delta b_{j,t} = f_{j+n}(\varphi(t)) \quad (j = 1, \ldots, m) \end{array}\right\} \tag{5.7}$$

$$f_i(\varphi(t)) = \sum_{j=1}^{M} \omega_{ij} \mathcal{N}_f\left(p_j, \varphi(t)\right) \tag{5.8}$$

where $\mathcal{N}_f\left(p_j, \varphi(t)\right)$'s are the 'basis functions', $\omega_{ij}$'s are the coordinate parameters, and $p_j$'s are the scale and position parameter vectors. Without loss of generality, the NNMs used to realize $f_i(\varphi(t))$ are assumed to have the same structure, so that the 'basis function' is independent of the index $i$. A preferable candidate for the NNMs is *adaptive fuzzy systems* (AFS) [110], which can be explicitly expressed as

$$f_i(\varphi(t)) = \frac{\sum_{j=1}^{M} \omega_{ij}(\wedge_{k=1}^{r} \mu_{A_k^j}(x_k(t)))}{\sum_{j=1}^{M}(\wedge_{k=1}^{r} \mu_{A_k^j}(x_k(t)))} \tag{5.9}$$

where $\wedge$ is the minimum operator, $M$ is the number of rules, $x_k(t)$'s are the elements of $\varphi(t)$, and $\mu_{A_k^j}$ is the membership function of fuzzy set $A_k^j$. The model described by (5.4)-(5.8) is named as *hybrid quasi-ARMAX model.*

### 5.2.3 Estimation of the Hybrid Model

It is well known that the NNMs are flexible enough to represent most reasonable systems in practice. It should be noted that there are a group of such certain NNMs in the hybrid quasi-ARMAX model. The hybrid quasi-ARMAX model becomes so flexible (complex) that it is impossible to estimate all the parameters ($a_i$, $b_i$, $c_i$, $\omega_{ij}$ and $p_j$) from observed data as usually done in a conventional system identification. In order to make the problem feasible, we will appropriately determine the parameter vectors $p_j$ specifying the 'basis functions' in the NNM by using knowledge information, and only estimate $a_i$, $b_i$, $c_i$ and $\omega_{ij}$ from observed data.

#### (1) Determining $\mathbf{p}_j$ and $M$ Using Knowledge Information

The efficient use of knowledge information for determining the parameter vector $p_j$ and the order $M$ plays a key role in the hybrid quasi-ARMAX modeling [37]. We, however, have to leave the discussion in the separate chapter, referred to Chapter 2, see also (J Hu et al., 1996) [37]. The follows are some points:

- The least prior knowledge required for determining $p_j$ is the information concerning operating region of $\varphi(t) = [x_i;\ i = 1, ..., r]^T$. That is, $[X_{\min}, X_{\max}]$ should be known for the statement that the operating region is mostly located in $X_{\min} \leq \varphi(t) \leq X_{\max}$.
- For the case where AFSs described by (5.9) are used, if the number of fuzzy sets for variable $x_i$ is denotes as $n_i$, then the number of rules is $M = \prod_{i=1}^{r} n_i$. When $\dim(\varphi(t))$ is large, $M$ will be rather large. Therefore, knowledge information obtained from observed data or the prediction error during the parameter estimation should be used to choose $n_i$ $(i = 1, ..., r)$ as small as possible, and to remove some rules by employing interpolation.

#### (2) Estimation of $a_i$, $b_i$, $c_i$ and $\omega_{ij}$

Introduce a parameter vector $\Theta$ and a regression vector $\varphi_{NL}(t)$ defined as

$$\Theta = [\theta^T,\ \omega_{11}\ ...\ \omega_{rM},\ c_1\ ...\ c_l]^T \tag{5.10}$$

$$\varphi_{NL}(t) = [\varphi^T(t),\ \varphi^T(t) \otimes \varphi_{\mathcal{N}_f}^T(t),\ e(t-1)\ ...\ e(t-1)]^T \tag{5.11}$$

where $\varphi_{\mathcal{N}_f}^T(t) = [\mathcal{N}_f(p_j, \varphi(t),\ j = 1, ..., M]$. Then we can express the hybrid quasi-ARMAX model in a 'pseudo-linear' regression structure

$$\mathcal{M}: \quad y(t) = \varphi_{NL}^T(t)\Theta + e(t). \tag{5.12}$$

It is well known that the estimation of $\Theta$ for (5.12) can be performed by using an Extended Least Square (ELS) recursive identification algorithm [70, 94].

## 5.3 Fault Detection Using The Hybrid Model

In this section, we will show that the hybrid quasi-ARMAX model is typically suitable for fault detection of nonlinear systems.

### 5.3.1 Modeling for A Robust Fault Detection Like Scheme

Using (5.8) and (5.7) in (5.4), we can get an alternative expression of the hybrid model

$$\mathcal{M}(\theta_e,\Omega): \quad y(t) = \underbrace{\varphi_e^T(t)\theta_e + e(t)}_{\text{ARMAX}} + \sum_{j=1}^{M} \underbrace{\varphi^T(t)\Omega_j}_{\text{ARX}} \mathcal{N}_f(\boldsymbol{p}_j, \varphi(t)) \tag{5.13}$$

where

$$\begin{aligned} \varphi_e(t) &= [\varphi^T(t),\ e(t-1)\ \dots\ e(t-l)]^T \\ \theta_e &= [a_1\ \dots\ a_n\ b_1\ \dots\ b_m\ c_1\ \dots\ c_l]^T \\ \Omega_j &= [\omega_{1j}\ \dots\ \omega_{rj}]^T \\ \Omega &= [\Omega_1^T\ \dots\ \Omega_M^T]^T. \end{aligned} \tag{5.14}$$

From (5.13), we notice that the hybrid quasi-ARMAX model can be considered as a hybrid model combining a linear ARMAX model with a multi-ARX-model.

For the case where the system properties of interest can be described by a linear model, the hybrid model (5.13) can be used for a best linear approximation of the system by the ARMAX term, and used for the estimation of resulting unmodeled dynamics by the multi-ARX term. Such a model matching can be realized by a hierarchical implementation of recursive identification. Then the fault detection can be performed using a robust fault detection like approach, in which the KDI is applied to discriminate the model distortion due to parameter changes and the estimate of unmodeled dynamics is used in the KDI analysis and thresholding decision for robustness realization [56].

### 5.3.2 Modeling for A Multi-Model Based Fault Detection Scheme

In the robust fault detection like scheme, fault detection is based on a global linear approximation ($\hat{\theta}$) with unmodeling dynamics. However, when the system has strong nonlinearity, a global linear approximation can not model the main characteristics of the system. That implies that the local parameters $\Omega_j$ in (5.13) may contain more useful information for fault detection than the global parameter $\theta$. Now let us remove $a_i$, $b_i$ from (5.13). We can then write (5.13) in the form

$$\begin{aligned} \mathcal{M}\{\Omega_e\}: \quad y(t) &= \sum_{j=1}^{M} \underbrace{\varphi^T(t)\Omega_j}_{\text{ARX}} \mathcal{N}_f(\boldsymbol{p}_j, \varphi(t)) + C(q^{-1})e(t) \\ &= \sum_{j=1}^{M} \underbrace{\left(\varphi_e^T(t)\Omega_{ej} + e(t)\right)}_{\text{ARMAX}} \mathcal{N}_f(\boldsymbol{p}_j, \varphi(t)) \end{aligned} \tag{5.15}$$

where $\mathcal{N}_f(\boldsymbol{p}_j, \varphi(t))$ is assumed to be normalized, i.e., $\sum_{j=1}^{M} \mathcal{N}_f(\boldsymbol{p}_j, \varphi(t)) = 1$, and the parameter vectors $\Omega_{ej}$ are defined by

$$\Omega_{ej} = [\omega_{1j}\ \dots\ \omega_{nj}\ \omega_{(n+1)j}\ \dots\ \omega_{rj}\ c_1\ \dots\ c_l]^T \tag{5.16}$$

$$\Omega_e = [\Omega_{e1}^T\ \dots\ \Omega_{eM}^T]^T. \tag{5.17}$$

When $\mathcal{N}_f(\boldsymbol{p}_j, \varphi(t))$'s are chosen to be convex and compact, the identified model (5.15) can be considered to be a multi-ARMAX-model consisting of $M$ identified local linear ARMAX models

$$\begin{aligned} \mathcal{M}\{\hat{\Omega}_{ej}\}: \quad & z_j(t) = \phi_j^T(t)\hat{\Omega}_{ej} + e(t) \\ & j = 1, \dots, M \end{aligned} \tag{5.18}$$

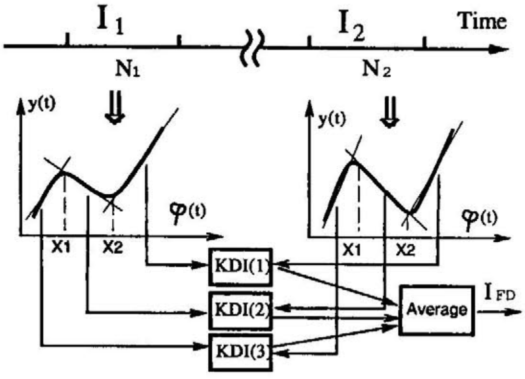


Figure 5.1: KDI-based fault detection using multi-model

where $z_j(t)$ stands for the output variable corresponding to the local model and $\phi_j(t)$ is defined by $\phi_j(t) = [z_j(t-1) \ldots z_j(t-n)\ u(t-1) \ldots u(t-m)\ e(t-1) \ldots e(t-l)]^T$. The identified local ARMAX models denote local linear approximations of the system. The overall performance of the multi-model is obtained via an interpolation using the 'basis function' $\mathcal{N}_f(x)$. We can thus consider that the estimates of the local parameter vectors $\hat{\Omega}_j$ contains useful information for fault detection. In the next section, we will propose a multi-model based fault detection scheme by applying the KDI to discriminate each identified local model (5.18) instead the multi-model (5.15).

## 5.4 A Multi-Model Based Fault Detection Scheme

Let us first consider a rather general case where the system properties of interest may not be described by a linear model. We will introduce a multi-model based fault detection scheme. The basic idea of a multi-model based fault detection can be summarized as the follows: a multi-model consisting of several local linear models with employing interpolation is used to identify the nonlinear system to be diagnosed. The identified multi-model can be interpreted as several identified local linear models. Then fault detection is performed using the KDI as index to discriminate the identified local linear models. Figure 5.1 shows an example of multi-model based fault detection scheme using the KDI as index for model discrimination, in which the multi-model consists of $M$ $(= 3)$ local linear models. As shown in Fig. 5.1, the system operating region is divided into three parts: $\varphi(t) \leq X_1$, $X_1 < \varphi(t) \leq X_2$ and $\varphi(t) > X_2$. In each local region, the system is represented by using a linear model. The global performance of the system is described by the three local linear models with employing interpolation. Next, fault detection is performed by using the KDI as index to discriminate each identified local model. The average of the KDI indexes for the three identified local linear models is used as fault detection index for the nonlinear system.

### 5.4.1 Local Linear ARMAX Models

In order to apply the KDI-based fault detection scheme, let us rewrite the local models (5.18) into the ARMAX form

$$\begin{aligned} \mathcal{M}\{\hat{\Omega}_{ej}\}: \quad & A(q^{-1},\hat{\Omega}_{ej})z_j(t) = B(q^{-1},\hat{\Omega}_{ej})u(t) + C(q^{-1},\hat{\Omega}_{ej})e(t) \\ & e(t) \in N(0,\hat{\sigma}^2) \qquad j = 1,\ldots,M \end{aligned} \tag{5.19}$$

where $z_j(t)$ is the output variable of the $j$-th local model at time $t$ $(t = 1,2,..)$, $u(t)$ the input, $e(t)$ the white Gaussian noise. $A(q^{-1},\hat{\Omega}_{ej})$, $B(q^{-1},\hat{\Omega}_{ej})$, $C(q^{-1},\hat{\Omega}_{ej})$ are scalar polynomial function in $q^{-1}$

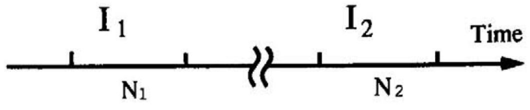


Figure 5.2: Data from two disjunct time intervals

with the orders $n$, $m$, $l$, respectively

$$\begin{aligned} A(q^{-1},\hat{\Omega}_{ej}) &= 1-\hat{\omega}_{1j}q^{-1}-...-\hat{\omega}_{nj}q^{-n} \\ B(q^{-1},\hat{\Omega}_{ej}) &= \hat{\omega}_{(n+1)j}q^{-1}+...+\hat{\omega}_{rj}q^{-m} \\ C(q^{-1},\hat{\Omega}_{ej}) &= 1+\hat{c}_1q^{-1}+...+\hat{c}_lq^{-l}. \end{aligned} \tag{5.20}$$

If we introduce the following notations

$$\begin{aligned} G(q^{-1},\hat{\Omega}_{ej}) &= B(q^{-1},\hat{\Omega}_{ej})/A(q^{-1},\hat{\Omega}_{ej}) \\ H(q^{-1},\hat{\Omega}_{ej}) &= C(q^{-1},\hat{\Omega}_{ej})/A(q^{-1},\hat{\Omega}_{ej}) \end{aligned} \tag{5.21}$$

(5.19) can be rewritten in the output error form

$$\mathcal{M}\{\Omega_{ej}\}: \qquad \begin{aligned} z_j(t) = G(q^{-1},\Omega_{ej})u(t)+H(q^{-1},\Omega_{ej})e(t) \\ j=1,...,M \end{aligned} \tag{5.22}$$

### 5.4.2 Fault Detection Index (The KDI)

The KDI, a distortion measure of two probability density functions, has been developed as an effective index to detect the fault in linear dynamic systems, see e.g. [61, 55].

Assume that data from the system are available from two disjunct time intervals $I_1$ and $I_2$, see Fig.5.2, and the system is identified by using the model described by (5.15). Using a prediction error method, we can estimate the unknown parameter vectors $\{\hat{\Omega}_{ej}\}$ using the data obtained from the two disjunct time intervals $I_1$ and $I_2$, and denote the estimates by $\{\hat{\Omega}_{e1j}\}$ and $\{\hat{\Omega}_{e2j}\}$ ($\hat{\Omega}_{1e}$ and $\hat{\Omega}_{2e}$ for (5.15)), respectively. It should be noted that so far the KDI has been applied to discriminate the identified models $\mathcal{M}\{\hat{\Omega}_{1e}\}$ and $\mathcal{M}\{\hat{\Omega}_{2e}\}$ based on the assumption that the corresponding likelihood functions obey Gaussian distribution. However, in our case such assumption can no longer be used for the model (5.15), since the model is really nonlinear one. Therefore, we will apply the model discrimination via the KDI to each identified local linear ARMAX model $\mathcal{M}\{\hat{\Omega}_{ej}\}$ described by (5.18), (5.19) or (5.23). Thus the distortion of identified local linear models $\mathcal{M}\{\hat{\Omega}_{e1j}\}$ and $\mathcal{M}\{\hat{\Omega}_{e2j}\}$ is discriminated by using the KDI.

Applying the KDI to the likelihood functions of the identified local linear models, $p(Z_{N_1}^{(j)}|\hat{\Omega}_{eij},U_{N_1-1})$ ($i=1,\ 2;\ j=1,...,M$), we have:

$$I_{N_1}[1,2](j)=\int p(Z_{N_1}^{(j)}|\hat{\Omega}_{e1j},U_{N_1-1})\log\frac{p(Z_{N_1}^{(j)}|\hat{\Omega}_{e1j},U_{N_1-1})}{p(Z_{N_1}^{(j)}|\hat{\Omega}_{e2j},U_{N_1-1}))}dZ_{N_1}^{(j)} \tag{5.23}$$

where $U_{N_1-1}=[u(1)\ u(2)\ ...\ u(N_1-1)]^T$ is the system input sequence which was used in the interval $I_1$, while $Z_{N_1}^{(j)}=[z_j(1)\ z_j(2)\ ...\ z_j(N_1)]^T$ is the output sequence variable corresponding to identified local model. Due to the assumption on Gaussian distributed disturbance in (5.19), the likelihood functions of local linear models obey Gaussian distribution. Therefore, by applying Bayes' rule, (5.23) can be analyzed into an explicit form, by which the KDI can be evaluated in a feasible way for finite but fairly large data set [61].

The result of such an analysis is:

$$I_{N_1}[1,2](j)=I_0[1,2](j)+\sum_{i=1}^{3}I_{N_1}^{(i)}[1,2](j) \tag{5.24}$$

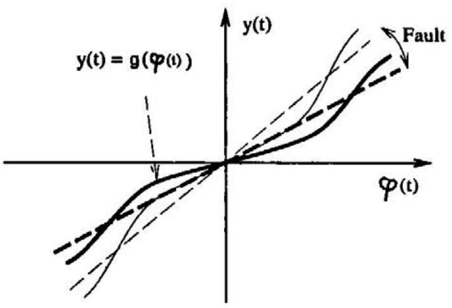


Figure 5.3: A nonlinear system whose main characteristics can be approximated using a linear model

$$j = 1, ..., M$$

where $I_0[1,2](j)$ is an initial value which might be neglected and other components are given as:

$$I_{N_1}^{(1)}[1,2](j) = \frac{N_1}{2}\left[(\hat{\sigma}_1^2/\hat{\sigma}_2^2 - 1) - \log(\hat{\sigma}_1^2/\hat{\sigma}_2^2)\right] \tag{5.25}$$

$$I_{N_1}^{(2)}[1,2](j) = \frac{1}{2}\sum_{k=0}^{N_1-1}\left\|H_2^{-1}(G_1^{(j)} - G_2^{(j)})u(k+1)\right\|_{\hat{\sigma}_2^{-2}}^2 \tag{5.26}$$

$$I_{N_1}^{(3)}[1,2](j) = \frac{N_1}{2}\left\{\frac{\hat{\sigma}_2^{-2}}{2\pi i}\oint\left((H_2^{(j)}(z))^{-1}H_1^{(j)}(z) - 1\right)\right. \\ \left.\cdot\hat{\sigma}_1^2\left(H_1^{(j)}(z^{-1})(H_2^{(j)}(z^{-1}))^{-1} - 1\right)\frac{dz}{z}\right\} \tag{5.27}$$

$$G_i^{(j)} = G(q^{-1}, \hat{\Omega}_{eij}), \quad H_i^{(j)}(z) = H(z, \hat{\Omega}_{eij}) \\ i = 1, 2. \tag{5.28}$$

Then the index for fault detection is introduced as

$$I_{FD} = \frac{1}{M}\sum_{j=1}^{M} I_{N_1}[1,2](j) \tag{5.29}$$

and the fault detection can be executed by the thresholding approach

$$I_{FD} \overset{>}{<} \eta \implies \begin{cases} \text{fault} \\ \text{no fault} \end{cases} \tag{5.30}$$

where the threshold $\eta$ should be appropriately determined under a specification of decision accuracy.

## 5.5 A Robust Fault Detection Like Scheme

In this section, we will constrain our discussion to a comparatively simple case where the system properties of interest can be described by using a linear model. That is, as shown in Fig. 5.3, the system fault can be reflected by the parameter changes of the linear model (a rotation of the $y(t) \sim \varphi(t)$ curve). For such a case, the KDI-based robust fault detection scheme can be applied on the basis of a linear model with model uncertainty, in which the error due to nonlinear undermodeling is treated as unmodeled dynamics.

### 5.5.1 Descriptions of System Dynamics and Unmodeled Dynamics

In order to apply the KDI-based fault detection scheme, let us rewrite the model (5.13) the ARMAX form

$$\mathcal{M}\{\theta_e,\xi\}: \quad \bar{A}(q^{-1},t,\theta_e,\xi)y(t) = \bar{B}(q^{-1},t,\theta_e,\xi)u(t) + \bar{C}(q^{-1},\theta_e,\xi)e(t) \tag{5.31}$$

$$e(t) \in N(0,\sigma^2) \tag{5.32}$$

where $\bar{A}(q^{-1},t,\theta_e,\xi)$, $\bar{B}(q^{-1},t,\theta_e,\xi)$ and $\bar{C}(q^{-1},\theta_e,\xi)$ are scalar polynomial function in $q^{-1}$ with the orders $n$, $m$ and $l$, respectively.

$$\begin{aligned}\bar{A}(q^{-1},t,\theta_e,\xi) &= 1 - \bar{a}_{1,t}q^{-1} - ... - \bar{a}_{n,t}q^{-n}\\ \bar{B}(q^{-1},t,\theta_e,\xi) &= \bar{b}_{1,t}q^{-1} + ... + \bar{b}_{m,t}q^{-m}\\ \bar{C}(q^{-1},\theta_e,\xi) &= 1 + \bar{c}_1q^{-1} + ... + \bar{c}_lq^{-l}.\end{aligned} \tag{5.33}$$

where $\bar{a}_{i,t}$ and $\bar{b}_{i,t}$ are coefficients containing two parts: constant parameters and nonlinear terms, while the constant parameters $\bar{c}_i$ are assumed to be divided into two parts

$$\begin{aligned}\bar{a}_{i,t} &= a_i + f_{i,t}\\ \bar{b}_{i,t} &= b_i + f_{i+n,t}\\ \bar{c}_i &= c_i + \delta_{c_i}\end{aligned} \tag{5.34}$$

Here $a_i$, $b_i$, $c_i$ are constant parameters (without confusion, the notation $c_i$ has been assumed to have slightly different meanings) and $f_{i,t}$ $(i = 1,...n+m)$ are the nonlinear terms which are represented by a group of NNMs (neural networks, adaptive fuzzy systems, etc.) described by (5.8). Then the parameters to be estimated for the *hybrid quasi-ARMAX model* are defined as follows:

$$\begin{aligned}\theta_e &= [a_1\ ...\ a_n\ b_1\ ...\ b_m\ c_1\ ...\ c_l]^T\\ \xi &= [\omega_{11}\ ...\ \omega_{n1}\ ...\ \omega_{r1}\ ...\ \omega_{rM}\ \delta_{c_1}\ ...\ \delta_{c_l}]^T.\end{aligned} \tag{5.35}$$

Note that it is easily to show that the parameter $\theta_e$ defined in (5.35) is a *redundant parameter vector* of $\mathcal{M}\{\theta,\xi\}$, which enable us to realize the identification in a hierarchical way. Now if we introduce the following notations

$$G = B(q^{-1},\theta_e)/A(q^{-1},\theta_e) \qquad H = \bar{C}(q^{-1},\theta_e,\xi)/A(q^{-1},\theta_e) \tag{5.36}$$

$$\triangle G_t = \frac{-A_\triangle(q^{-1},t,\xi)y(t) + B_\triangle(q^{-1},t,\xi)u(t)}{A(q^{-1},\theta_e)u(t)} \tag{5.37}$$

where

$$\begin{aligned}A(q^{-1},\theta_e) &= 1 + a_1q^{-1} + ... + a_nq^{-n}\\ B(q^{-1},\theta_e) &= b_1q^{-1} + ... + b_mq^{-m}\\ A_\triangle(q^{-1},t,\xi) &= f_{1,t}q^{-1} + ... + f_{n,t}q^{-n}\\ B_\triangle(q^{-1},t,\xi) &= f_{(1+n),t}q^{-1} + ... + f_{(m+n),t}q^{-m},\end{aligned} \tag{5.38}$$

then (5.31) can be rewritten as

$$\mathcal{M}\{\theta_e,\xi\}: \quad y(t) = Gu(t) + \triangle G_tu(t) + He(t) \tag{5.39}$$

where $G$ describes system dynamics, while $\triangle G_t$ denotes unmodeled dynamics. The identification of the model (5.39) can be realized in the following two steps:

(1) Fix $\xi = 0$ and match (5.39) to system. Then we can obtain the estimate $\hat{\theta}_e$.

(2) Fix $\theta_e = \hat{\theta}_e$ and match (5.39) to system. Then we can obtain the estimate $\hat{\xi}$.

### 5.5.2 Fault Detection Indexes (The KDI)

Assume that the data from system are available from two distinct time intervals $I_1$ and $I_2$, see Fig.5.2. We estimate the unknown parameter vectors specifying the model and the error description, $\{\theta_e, \xi\}$, using the data obtained from the two distinct intervals $I_1$ and $I_2$, and denote the estimates by $\{\hat{\theta}_{e1}, \hat{\xi}_1\}$ and $\{\hat{\theta}_{e2}, \hat{\xi}_2\}$ respectively. The KDI is then used to detect the distortion of identified models $\mathcal{M}\{\hat{\theta}_{e1}, \hat{\xi}_1\}$ and $\mathcal{M}\{\hat{\theta}_{e2}, \hat{\xi}_2\}$.

Applying the KDI to the likelihood functions of identified models, $p(Y_{N_1}|\hat{\theta}_{ei}, \hat{\xi}_i, U_{N_1-1})$ $(i = 1,\ 2)$, we get

$$I_{N_1}[1,2] = \int p(Y_{N_1}|\hat{\theta}_{e1}, \hat{\xi}_1, U_{N_1-1}) \log \frac{p(Y_{N_1}|\hat{\theta}_{e1}, \hat{\xi}_1, U_{N_1-1})}{p(Y_{N_1}|\hat{\theta}_{e2}, \hat{\xi}_2, U_{N_1-1})} dY_{N_1} \tag{5.40}$$

where $U_{N_1-1} = [u(1)\ u(2)\ \dots\ u(N_1-1)]^T$ and $Y_{N_1} = [y(1)\ y(2)\ \dots\ y(N_1)]^T$ are the input-output data sets from the interval $I_1$. Strictly speaking, the likelihood functions $p(Y_{N_1}|\hat{\theta}_{ei}, \hat{\xi}_i, U_{N_1-1})$ do not obey Gaussian distribution, since the (5.39) is a nonlinear model. Let us consider that the effect of $\Delta G_t$ is not so large, and assume that the likelihood functions $p(Y_{N_1}|\hat{\theta}_{ei}, \hat{\xi}_i, U_{N_1-1})$ obey Gaussian distribution. Then, by applying Bayes' rule, (5.40) can be approximately analyzed into an explicit form, by which the KDI can be evaluated in a feasible way for finite but fairly large data set, see [61, 56].

The result of such an analysis is (see Chapter 4 for the details of the derivation):

$$I_{N_1}[1,2] = I_0[1,2] + \sum_{i=1}^{4} I_{N_1}^{(i)}[1,2] \tag{5.41}$$

$$j = 1, \dots, M$$

where $I_0[1,2](j)$ is an initial value which might be neglected and $I_{N_1}^{(i)}[1,2]$ $(i = 1, \dots, 3)$ have the same forms as $I_{N_1}^{(i)}[1,2](j)$ $(i = 1, \dots, 3)$ described by (5.25)-(5.27), while $I_{N_1}^{(4)}[1,2]$ is given by

$$\begin{aligned} I_{N_1}^{(4)}[1,2] = \frac{1}{2} \sum_{k=0}^{N_1-1} & H_2^{-1}(2G_1 - 2G_2 + \Delta G_{1,k+1} - \Delta G_{2,k+1})u(k+1) \\ & \cdot H_2^{-1}(\Delta G_{1,k+1} - \Delta G_{2,k+1})u(k+1)/\hat{\sigma}_2^2 \end{aligned} \tag{5.42}$$

which is the term induced from unmodeled dynamics. And the notations $G_i$ and $\Delta G_{i,k}$ are defined as

$$G_i \triangleq G(q^{-1}, \hat{\theta}_{ei}) \tag{5.43}$$

$$\Delta G_{i,k} \triangleq \Delta G_k / \ \theta_e = \hat{\theta}_{ei}, \xi = \hat{\xi}_i, \quad i = 1, 2. \tag{5.44}$$

### 5.5.3 Robust Fault Detection Scheme

In the ideal case where no unmodeled dynamics exists, the fault detection can be executed by applying the thresholding approach to the KDI. In the presence of unmodeled dynamics, the identified model parameters may change depending on the noise realization, and this causes a fluctuation of the calculated KDI. When the modeling error and the noise are large, the fluctuation might be so large that the thresholding approach to the KDI becomes infeasible. Therefore, robust decision making scheme should be developed so as to use the indexes of the KDI and the indexes for the modeling error effectively.

Many approaches may be considered to build the robust decision making scheme. As an example, we will realize a robust decision making based on statistical test on the KDI. Let us consider the case where the fault detection is implemented on-line. The KDI $I_{N_1}[1, 2:t]$ will be calculated for the estimates $\hat{\theta}_{e2}(t)$ and $\hat{\xi}_2(t)$. The fault detection index can be constructed by averaging the sum of the first three terms of the KDI

$$\bar{I}_{N_1}[1, 2:t] = \frac{1}{\mathcal{L}} \sum_{i=t-\mathcal{L}}^{t} \left( I_{N_1}^{(1)}[1, 2:i] + I_{N_1}^{(2)}[1, 2:i] + I_{N_1}^{(3)}[1, 2:i] \right) \tag{5.45}$$

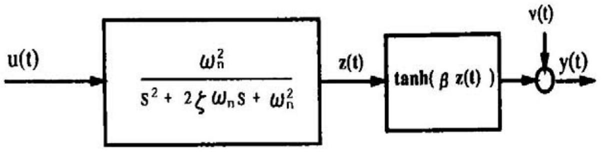


Figure 5.4: Nonlinear system for Example 1

Table 5.1: The $\mathcal{L}$, $\eta$ and the indexes for the modeling error for Example 1

| $\beta$ | Mean Value of $I_{\Delta G}^{(1)}(t)$ | Mean Value of $I_{\Delta G}^{(2)}(t)$ | Window Width $\mathcal{L}$ | Threshold $\eta$ |
|---|---|---|---|---|
| 0.5 | 0.4388 | 0.0207 | 2 | 8 |
| 1.0 | 2.7142 | 0.1312 | 5 | 4 |
| 2.5 | 4.3372 | 0.2316 | 15 | 3 |

and execute fault detection by the following thresholding approach

$$\bar{I}_{N_1}[1,2:t] \gtrless \eta \Longrightarrow \begin{cases} \text{fault} \\ \text{no fault} \end{cases} \tag{5.46}$$

In order to realize robustness of the fault detection, the window $\mathcal{L}$ and the threshold $\eta$ may be appropriately determined based on the evaluated indexes for the modeling error and by considering other information about the system such as variance of disturbance $\hat{\sigma}^2$. Two indexes of unmodeled dynamics may be introduced. One is based the fourth term of the KDI, which is introduced as

$$I_{\Delta G}^{(1)}(t) = \frac{1}{\mathcal{W}} \sum_{j=t-\mathcal{W}}^{t} I_{N_1}^{(4)}[1,2:j]. \tag{5.47}$$

The second one is the mean value of $|\Delta G_k(\hat{\theta}_{e2}(t), \hat{\xi}_2(t))|$; $k = 0, ..., N_1 - 1$ calculated from the input-output data sets in the internal $I_1$

$$I_{\Delta G}^{(2)}(t) = \frac{1}{N_1} \sum_{k=0}^{N_1-1} |\Delta G_k(\hat{\theta}_{e2}(t), \hat{\xi}_2(t))|. \tag{5.48}$$

## 5.6 Numerical Simulations

We will carry out some numerical simulations to examine the fault detection schemes discussed in the previous sections.

### 5.6.1 Example 1

Let us consider a nonlinear system shown in Fig. 5.4, which consists of a second order linear system and a nonlinear element $\tanh(x)$. Obviously, when $x$ is small, the nonlinearity resulted from $\tanh(x)$ is weak ($x \longrightarrow 0$, $\tanh(x) \longrightarrow x$) and when $x$ is large, the nonlinearity is strong. Therefore in Fig. 5.4, $\beta$ can be considered as a parameter for adjusting the system nonlinearity. In the normal mode, the parameters $\omega_n$ and $\zeta$ of the damped oscillator are chosen to be 2 and 0.32, respectively. After being sampled with a period of 0.25, the system is corrupted by a white noise sequence $v(t) \in N(0, 0.01)$, and the input of system $u(t)$ is PRBS (pseudo-random binary sequence).

The model described by (5.31) is used to identify the system, in which the model order are chosen to be $n = 2$, $m = 2$, $l = 0$. Moreover, $M = 16$ is obtained with choosing $X_{\min}$=[-1 -1 -1 -1], $X_{\max}$=[1

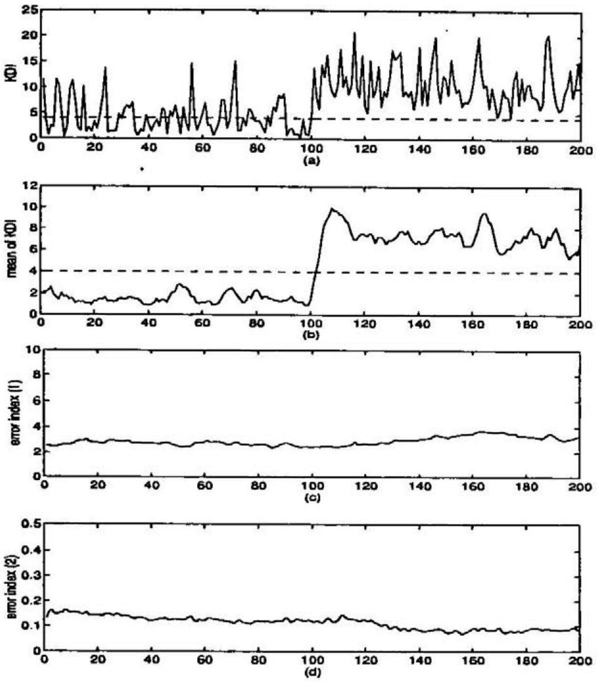


Figure 5.5: The results of Example 1: (a) the sum of the first three terms of KDI; (b) the fault detection index $\bar{I}_{N_1}[1,2t]$; (c) the index for the modeling error $I^{(1)}_{\triangle G}(t)$; (d) the index for the modeling error $I^{(2)}_{\triangle G}(t)$.

1 1 1] and $n_1 = n_2 = 4$, $n_3 = n_4 = 1$. In the simulation, the input-output data are available from two intervals $I_1$ and $I_2$. The interval $I_1$ contains 500 data sets, from which $\{\hat{\theta}_{e1}, \hat{\xi}_1\}$ is estimated, while the interval $I_2$ contains 700 data sets, from which $\{\hat{\theta}_{e2}(t), \hat{\xi}_2(t)\}$ $(t = 0, ..., 200)$ is estimated on-line. The initial values $\{\hat{\theta}_{e2}(0), \hat{\xi}_2(0)\}$ for the on-line estimation are obtained from the first 500 data sets. The fault is assumed to be a jump change, $\zeta = 0.32 \to 0.6$ at $t = 100$.

Three cases are considered, in which $\beta = 0.5,\ 1.0,\ 2.5$. When $\beta$ is large, the modeling error is expected to be large because the system nonlinearity increases. For each case, the output of KDI $I_{N_1}[1,2:t]$ is calculated from (5.41)-(5.42). The output of fault detection index $\bar{I}_{N_1}[1,2:t]$ is calculated from (5.45). And the indexes for the modeling error $I^{(1)}_{\triangle G}(t)$ and $I^{(2)}_{\triangle G}(t)$ are calculated from (5.47) with $\mathcal{W} = 60$ and (5.48), respectively.

Table 5.1 shows the values of indexes calculated from the estimate of the modeling error and the values of $\mathcal{L}$ and $\eta$ chosen in the simulations. We can see that $I^{(1)}_{\triangle G}(t)$ and $I^{(2)}_{\triangle G}(t)$ increase as $\beta$ increases. And when $I^{(1)}_{\triangle G}(t)$ and $I^{(2)}_{\triangle G}(t)$ increase, we confirmed that $\mathcal{L}$ should be chosen larger while $\eta$ be taken to be smaller.

Figure 5.5 shows the results of $\beta = 1.0$: (a) the sum of the first three terms of KDI; (b) the output of fault detection index $\bar{I}_{N_1}[1,2:t]$; (c) the index for the modeling error $I^{(1)}_{\triangle G}(t)$; (d) the index for the modeling error $I^{(2)}_{\triangle G}(t)$. From Fig.5.5(b), we can see that the fault was detected definitely.

On the other hand, in Fig. 5.5 we used the sum of the first three terms of KDI as index fault detection. However, in the three terms, each has its particular relationship with the model parameters and will provides certain information which may be useful for fault detection and fault analysis. Next, let us see the outputs of the four KDI terms shown in Fig. 5.6. We may find that

- the first term, which denotes the difference the prediction errors of two identified models, seems do not contain useful information for fault detection. However, in the case where an identification

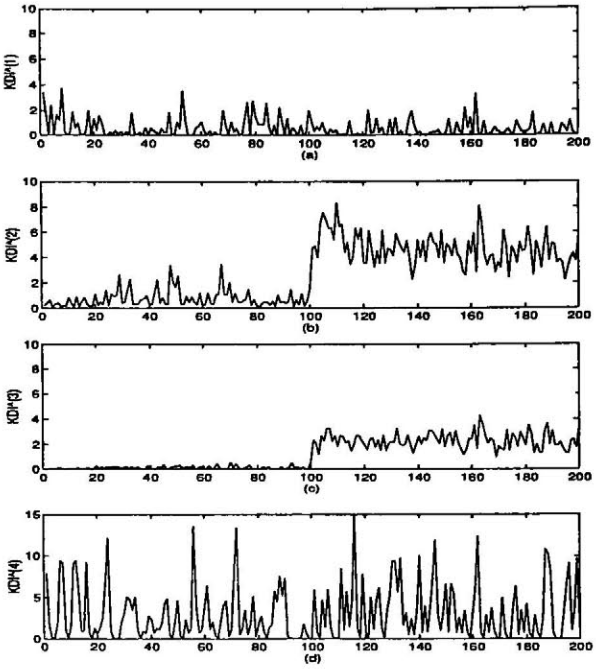


Figure 5.6: The outputs of the four KDI terms: (a) $\mathrm{KDI}^{(1)}[1,2:t]$, (b) $\mathrm{KDI}^{(2)}[1,2:t]$, (c) $\mathrm{KDI}^{(3)}[1,2:t]$, (d) $\mathrm{KDI}^{(4)}[1,2:t]$

algorithm can not tracing the system parameter change quickly enough, the first will give some useful information.

- the second and third terms contain useful information for fault detection. They obviously contain different information from each other, which may be useful for fault analysis or for realizing robust fault detection.
- the fourth term contains useful information for evaluating the effect of unmdoeled dynamics

However, further investigations are needed to find out how to use the information provided from the individual KDI terms.

### 5.6.2 Example 2

Let us consider a nonlinear function described by (5.49) as the true system

$$y(t) = f[y(t-1), y(t-2), y(k-t), u(t-1), u(t-2)] + e(t) \tag{5.49}$$

where

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + \alpha(x_2^2 + x_3^2)}$$

and $e(t) \in N(0, 0.01)$ is white Gaussian noise.

When identifying the system, we use the multi-ARMAX-model described by (5.15) as a model, where $n = 3$, $m = 2$, $l = 0$ are chosen for each local ARMAX models. And choosing $X_{\min}$=[-1 -1 -1 -1 -1], $X_{\max}$=[1 1 1 1 1], $n_1 = n_2 = n_3 = 3$, $n_4 = n_5 = 1$ and removing some of the rules, we obtained $M = 18$. In the simulation, the interval $I_1$ contains $N_1$ (= 5000) data sets, from which $\hat{\Omega}_{e1j}$; $j = 1, \ldots, M$ are obtained. The interval $I_2$ is divided into $N_s$ (= 100) parts, each of which contains $N_2$ (= 5000) data sets. From each part, we can obtain $\hat{\Omega}_{e2j}$, and calculate the distortion of

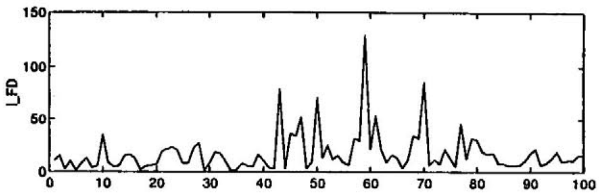


Figure 5.7: The result of setting $M = 1$, which implies that the robust fault detection like approach can not be applied to this example

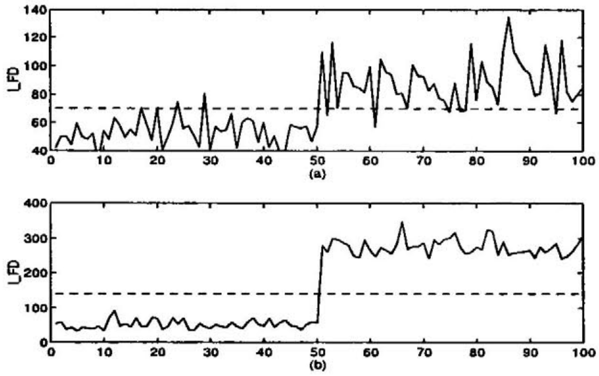


Figure 5.8: The results using the multi-model approach: (a) $\alpha$ changed from 1.0 to 1.2; (b) $\alpha$ changed from 1.0 to 2.0

the model $\mathcal{M}\{\hat{\Omega}_{e1j}\}$ and $\mathcal{M}\{\hat{\Omega}_{e2j}\}$ by using (5.24)-(5.29). The failure mode is assumed to be that $\alpha$ changes from its normal value 1.0 to 1.2, which occurs in the 50th part of the interval $I_2$.

First, we carried out a simulation by approximating the system using a linear ARMAX model, i.e, setting $M = 1$. Figure 5.7 shows the output of fault detection index $I_{FD}$, which was calculated by using (5.24)-(5.29). We can see that for such a nonlinear system, a global linear ARMAX model can not describe the properties of interest. This implies that a robust fault detection like approach may become infeasible, since the robust fault detection is mainly based on a best linear approximation of system and the estimate of the error due to nonlinear undermodeling is used in the KDI analysis and thresholding decision. Next, we use the multi-model based approach with $M = 18$. Figure 5.8 shows the output of $I_{FD}$. We can see that the detection of the failure becomes possible. Figure 5.8(b) shows the result when $\alpha$ is changed from 1.0 to 2.0.

## 5.7 Discussions and Conclusions

We have discussed fault detection of nonlinear systems based on the use of KDI for model discrimination. A hybrid quasi-ARMAX model has been proposed to describe general nonlinear systems in the input-output type. The hybrid quasi-ARMAX model has been transformed into a form combining a linear ARMAX model with a multi-ARX-model. This enable us to apply the hybrid quasi-ARMAX model to fault detection of nonlinear systems in two ways. For the case where the system properties of interest can be described by a linear ARMAX model, a robust fault detection like approach has been proposed, in which the hybrid quasi-ARMAX model is identified in such a way that the result gives a best linear approximation of the system and the estimate of error due to nonlinear undermodeling. For other cases, a multi-model based approach has been proposed, in which the hybrid quasi-ARMAX model is used as a multi-ARMAX-model consisting of several local linear ARMAX models, and the fault detection is performed by applying the KDI to discriminate the identified local ARMAX models. Based on the discussions and the simulations in this chapter, we can conclude:

(1) The hybrid quasi-ARMAX model is very suitable for fault detection and fault analysis of nonlinear systems.

(2) With the hybrid quasi-ARMAX model, we can extend the KDI-based fault detection scheme to nonlinear systems.

# Chapter 6

# Adaptive Control Using Hybrid Quasi-ARMAX Models

## 6.1 Introduction

When the systems to be controlled contain unknown parameters, adaptive controllers offer certain advantages over conventional controller. In the past three decades, adaptive control theory has been developed into a considerable mature stage [26, 63]. However, these major advances have only been made in adaptive control of linear time-invariant systems with unknown parameters [76]. Since most processes in real world are nonlinear to some extent, a crucial point of adaptive controller is its ability to deal with nonlinear systems.

Recently, neural networks have attracted much interest in system control community because of its ability to learn any nonlinear mapping. Many approaches have been proposed to construct a nonlinear control system by using neural networks [40]. Generally, neural networks have been incorporated into adaptive control systems in two ways. One is to use neural networks as nonlinear models . Most of such schemes are working as indirect control [76, 75]. The second is to use neural networks as nonlinear compensators to compensate the system nonlinearity or the error due to nonlinear undermodeling [102, 34]. In the latter schemes, since the control systems are mainly based on linear models, the convergence properties have been improved greatly. Loosely speaking, because of the nonlinearity in the parameters to be adjusted, the control systems incorporating neural networks usually have some drawbacks concerning the *convergence rate* and the *noise sensitivity* [39]. When stochastic systems are considered, the best known stochastic adaptive controller is the self-tuning regulator (STR) proposed by Åström and Wittenmark (1973) [1]. The self-tuning regulator combines the least-squares procedure for parameter estimation with an one-step-ahead (minimum variance) certainty equivalence controller. The STR adaptive controller is well known to have some advantages such as *quick convergence* and *insensitivity to noise.* But it also suffers some difficulties. For example, it is not so good in the robustness to unmodeled dynamics [39]. Recent researches show that by introducing some robust adaptive schemes such as *dead zone*, the STR adaptive controller is guaranteed to be robust to unmodeled dynamics [81, 18]. However, its tracing accuracy strongly depends on the performance of the prediction model. Therefore, when the system to be controlled contains nonlinearity, a nonlinear prediction model is required.

A difficulty with general nonlinear stochastic systems is that it may be difficult to determine an optimal predictor for the system output [26]. To get around this difficulty, one so far has to use physical insight, heuristics, informed guesswork, and so on, to come up with a predictor structure in practice. In this chapter, we will propose a hybrid quasi-ARMAX prediction model for adaptive control of general nonlinear stochastic systems. It is shown that a general nonlinear system can be expressed in a linear ARMAX structure whose coefficients consist of constant parameters and nonlinear terms. Then the nonlinear terms are represented by using a group of nonlinear nonparametric models (NNMs) (neural networks, adaptive fuzzy systems, etc.). Based on the fact that the NNMs have universal

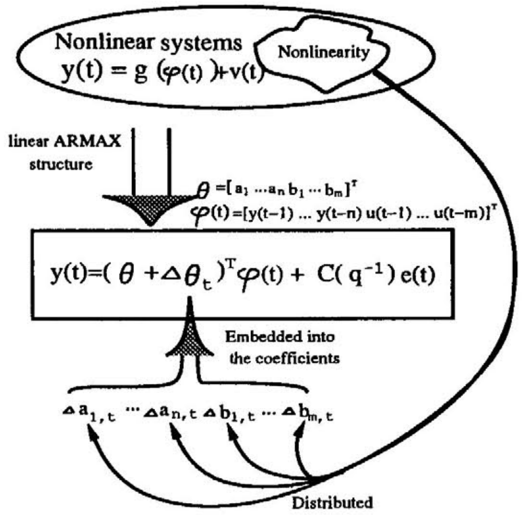

Nonlinear systems
$y(t) = g(\varphi(t)) + v(t)$
Nonlinearity
linear ARMAX structure
$\theta = [a_1 \dots a_n\, b_1 \dots b_m]^T$
$\varphi(t) = [y(t-1) \dots y(t-n)\, u(t-1) \dots u(t-m)]^T$
$y(t) = (\theta + \Delta\theta_t)^T \varphi(t) + C(q^{-1})e(t)$
Embedded into the coefficients
$\Delta a_{1,t} \cdots \Delta a_{n,t}\, \Delta b_{1,t} \cdots \Delta b_{m,t}$
Distributed

Figure 6.1: A schematic description of quasi-ARMAX modeling

approximation ability, a modeling scheme is developed by using knowledge information efficiently so that the obtained hybrid quasi-ARMAX prediction model is linear not only in the parameters to be adjusted but also in the one-step past input variable, so that a control law can be derived directly from the predictor. Based on the hybrid quasi-ARMAX predictor structure, a robust STR adaptive controller for general nonlinear stochastic systems can be designed in a similar way to the linear stochastic control theory.

The chapter is organized as follows: Section 6.2 proposes a hybrid quasi-ARMAX modeling scheme, with which a nonlinear predictor can be designed. In Section 6.3, we design a robust STR adaptive controller for general nonlinear stochastic systems in a similar way to the linear stochastic control theory. In Section 6.4, the proposed controller is applied to a variety of nonlinear systems to demonstrate its effectiveness. Finally, Section 6.5 is devoted to discussions and conclusions.

## 6.2 A Hybrid Quasi-ARMAX Modeling Scheme

In this section, we will propose a hybrid quasi-ARMAX modeling scheme for control design. The hybrid quasi-ARMAX model has been discussed extensively in Chapter 2. We here will discuss how to build such hybrid model easily applied to control design.

### 6.2.1 Quasi-ARMAX Modeling

Let us consider a SISO general nonlinear ARX system described by

$$\mathcal{S}: \quad y(t) = g(\varphi(t)) + v(t) \tag{6.1}$$

$$\varphi(t) = [y(t-1) \,\dots\, y(t-n)\; u(t-1) \,\dots\, u(t-m)]^T \tag{6.2}$$

where $y(t)$ is the output at time $t$ ($t = 1, 2, \dots$), $u(t)$ the input, $\varphi(t)$ the regression vector, $v(t)$ the system disturbance, and $g(\,\cdot\,)$ the unknown continuously differentiable nonlinear function.

Based on the idea of quasi-ARMAX modeling shown in Fig.6.1, we use a quasi-ARMAX model described by (6.3) to identify the system

$$\mathcal{M}: \quad y(t) = \varphi^T(t)(\theta + \Delta\theta_t) + C(q^{-1})e(t) \tag{6.3}$$

$$\left.\begin{array}{l}\theta = [a_1 \ldots a_n\; b_1 \ldots b_m]^T \\ \Delta\theta_t = [\Delta a_{1,t} \ldots \Delta a_{n,t}\; \Delta b_{1,t} \ldots \Delta b_{m,t}]^T \\ C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_l q^{-l}\end{array}\right\} \tag{6.4}$$

and where $a_i$, $b_i$ and $c_i$ are the constant parameters, while $\Delta a_{i,t}$ and $\Delta b_{i,t}$ are nonlinear functions of $\varphi(t)$, and $q^{-1}$ is backward shift operator, for instance, $q^{-1}u(t) = u(t-1)$, and $e(t)$ is white noise.

. If the nonlinear system is assumed to be the type of Kolmogrov-Gabor polynomial [17] of order $r$

$$\begin{aligned} y(t) &= \sum_{i=1}^{r} \alpha_i x_i(t) + \sum_{i=1}^{r}\sum_{j=1}^{r} \alpha_{ij} x_i(t) x_j(t) \\ &\quad + \sum_{i=1}^{r}\sum_{j=1}^{r}\sum_{l=1}^{r} \alpha_{ijl} x_i(t) x_j(t) x_l(t) + \ldots, \end{aligned} \tag{6.5}$$

in which the elements of $[x_i(t) \;\; i = 1, .., r]$ are assumed to be the past input-outputs of system

$$\left.\begin{array}{ll} x_i(t) = y(t-i); & i = 1, \ldots, n \\ x_{j+n}(t) = u(t-j); & j = 1, \ldots, m \\ (r = n+m) & \end{array}\right\}. \tag{6.6}$$

Then using (6.6) in (6.5), we can get

$$y(t) = \sum_{i=1}^{n} (a_i + \Delta a_{i,t}) y(t-i) + \sum_{i=1}^{m} (b_i + \Delta b_{i,t}) u(t-i) \tag{6.7}$$

where $\Delta a_{i,t}$ and $\Delta b_{i,t}$ can be explicitly expressed as

$$\Delta a_{i,t} = \sum_{j=1}^{n} \alpha_{ij} y(t-j) + \frac{1}{2} \sum_{j=n+1}^{r} \alpha_{ij} u(t-j+n) + \ldots \tag{6.8}$$

$$\Delta b_{i,t} = \sum_{j=n+1}^{r} \alpha_{ij} u(t-j+n) + \frac{1}{2} \sum_{j=1}^{n} \alpha_{ij} y(t-j) + \ldots \tag{6.9}$$

In (6.7), $a_i$ and $b_i$ stand for $\alpha_i$ in (6.5).

### 6.2.2 Hybrid Quasi-ARMAX Model for Control Design

For general nonlinear systems, $\Delta a_{i,t}$ and $\Delta b_{i,t}$ may become very complicated, and can not be expressed in explicit forms. We will therefore represent them using the NNMs. Generally speaking, the nonlinear terms $\Delta a_{i,t}$, $\Delta b_{i,t}$ are functions of the regression vector $\varphi(t)$ which includes the one-step past input $u(t-1)$ as its element. This implies that the input vector of the NNMs is $\varphi(t)$. However, when the model is applied to control design, the element $u(t-1)$ is preferred to be removed from the input vector of the NNMs so as to construct the model to be linear in the one-step past input $u(t-1)$ in favorable to deriving a control law directly from the hybrid quasi-ARMAX predictor. Based on the fact that the NNMs have universal approximation ability, we are able to remove $u(t-1)$ from the input vector of the NNMs. It follows that the hybrid quasi-ARMAX model for control design will be introduced by representing the nonlinear terms $\Delta a_{i,t}$, $\Delta b_{i,t}$ using NNMs whose input vector is $\bar{\varphi}(t) = [y(t-1) \ldots y(t-n)\; u(t-2) \ldots u(t-m-1)]^T$ instead of $\varphi(t)$.

Considering the fact that the inputs $\{u(t)\}$ $t = 1, 2, \ldots$ are not independent sequences in a controlled system since they are synthesized, let us assume that the variable $u(t-1)$ in the expressions of $\Delta a_{i,t}$ and $\Delta b_{j,t}$ can be approximated by the following equation[1]

$$u(t-1) = \rho\left(y(t-1) \ldots y(t-n)\; u(t-2) \ldots u(t-m-1)\right) \tag{6.10}$$

[1] Strictly speaking, the equation should be $u(t) = \rho(y(t) \ldots y(t-n-1)\; u(t-1) \ldots u(t-m)\; r(t))$, where $r(t)$ is the extra input used to drive the controlled system. For simplicity, we here omit the extra input $r(t)$.

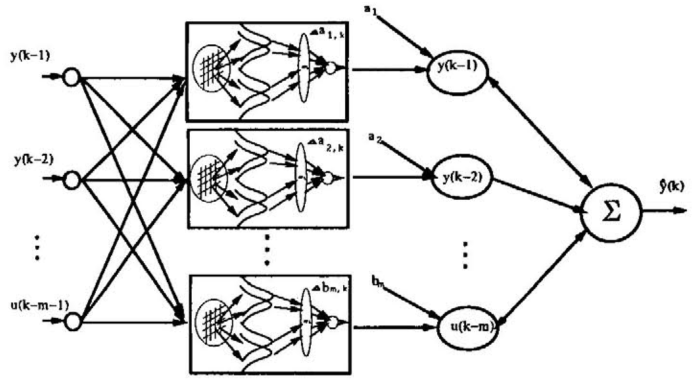


Figure 6.2: The hybrid quasi-ARMAX model shown as an associative memory networks. (MA noise model has be omitted for simplicity.)

provided that $\rho(\cdot)$ is a sufficiently smooth arbitrary function. Then we can get

$$\left.\begin{array}{ll}\Delta a_{i,t}=\Delta\tilde{a}_i(\tilde{\varphi}(t)) & (i=1,...,n)\\ \Delta b_{j,t}=\Delta\tilde{b}_j(\tilde{\varphi}(t)) & (j=1,...,m)\end{array}\right\}. \tag{6.11}$$

Now we represent the functions $\Delta\tilde{a}_i(\tilde{\varphi}(t))$ and $\Delta\tilde{b}_j(\tilde{\varphi}(t))$ using the NNMs described by (6.13)

$$\begin{array}{l}\Delta\tilde{a}_i(\tilde{\varphi}(t)=f_i(\tilde{\varphi}(t))\\ \Delta\tilde{b}_j(\tilde{\varphi}(t))=f_{j+n}(\tilde{\varphi}(t))\end{array} \tag{6.12}$$

$$f_i(\tilde{\varphi}(t))=\sum_{j=1}^{M}\omega_{ij}\mathcal{N}_f(\boldsymbol{p}_j,\tilde{\varphi}(t)) \quad (i=1,...,r) \tag{6.13}$$

where $\mathcal{N}_f(x)$ is the 'basis function', $\omega_{ij}$'s the coordinate parameters and $\boldsymbol{p}_j$'s the scale and position parameters of $\mathcal{N}_f(x)$. Without loss of generality, the NNMs used to realize $f_i(\cdot)$ are assumed to have the same structure, so that the 'basis functions' is independent of $i$. A preferable candidate for the NNMs is *adaptive fuzzy systems* (AFS) [110], which can be expressed explicitly as

$$f_i(\tilde{\varphi}(t))=\frac{\sum_{j=1}^{M}\omega_{ij}(\wedge_{k=1}^{r}\mu_{A_k^j}(x_k(t)))}{\sum_{j=1}^{M}(\wedge_{k=1}^{r}\mu_{A_k^j}(x_k(t)))} \tag{6.14}$$

where $\wedge$ is the minimum operator, $M$ is the number of rules, $x_k(t)$'s are the elements of $\tilde{\varphi}(t)$, and $\mu_{A_k^j}$ is the membership function of fuzzy set $A_k^j$. The model described by (6.3), (6.4) and (6.11)-(6.13) is named as *hybrid quasi-ARMAX model*, by which one can do control design easily. Furthermore, the proposed hybrid model can be considered as an *associative memory networks* which consists of two hidden layers: the first layer (next to the input layer) with weights determined by a set of simplified nonlinear nonparametric models; the second layer with weights simply taking the time delayed values of the system's input and output, see Fig. 6.2. Such a specially constructed network may be expected to be more suitable for control design.

On the other hand, from (6.3), (6.4) and (6.11)-(6.13) we can get another expression of the hybrid quasi-ARMAX model

$$y(t)=\underbrace{\varphi^T(t)\theta'+C(q^{-1})e(t)}_{\text{ARMAX}}+\sum_{j=1}^{M}\underbrace{\varphi^T(t)\Omega_j}_{\text{ARX}}\mathcal{N}_f(\boldsymbol{p}_j,\tilde{\varphi}(t)) \tag{6.15}$$

where $\Omega_j = [\omega_{1j} \; ... \; \omega_{rj}]^T$. (6.15) shows that the hybrid quasi-ARMAX model is equivalent to a hybrid model combining a linear ARMAX model and a multi-ARX-model. The multi-ARX-model consists of $M$ local ARX models and its overall performance is obtained via an interpolation using the 'basis functions' $\mathcal{N}_f(x)$. It also implies that the proposed model can be shown to be able to describe any sufficiently smooth nonlinear function in (6.1) on a compact interval arbitrarily well by merely increasing the value of $M$ (the number of ARX models in the multi-model). If we have $\theta = 0$ fixed, the (6.15) becomes

$$\begin{aligned} y(t) &= \sum_{j=1}^{M} \underbrace{\varphi^T(t)\Omega_j}_{\text{ARX}} \mathcal{N}_f(p_j, \tilde{\varphi}(t)) + C(q^{-1})e(t) \\ &= \sum_{j=1}^{M} \underbrace{\varphi_e^T(t)\Omega_{ej}}_{\text{ARMAX}} \mathcal{N}_f(p_j, \tilde{\varphi}(t)) \end{aligned} \tag{6.16}$$

where $\mathcal{N}_f(p_j, \tilde{\varphi}(t))$ is assumed to be normalized, i.e., $\sum_{j=1}^{M} \mathcal{N}_f(p_j, \tilde{\varphi}(t)) = 1$, and

$$\begin{aligned} &\Omega_{ej} = [\omega_{1j} \; ... \; \omega_{rj} \; c_1 \; ... \; c_l]^T \\ &\varphi_e(t) = [y(t-1) \; ... \; y(t-n) \; u(t-1) \; ... \; u(t-m) \; e(t-1) \; ... \; e(t-l)]^T. \end{aligned} \tag{6.17}$$

### 6.2.3 Estimation of the Hybrid Model

It is well known that the NNMs are flexible enough to represent most reasonable systems in practice. It should be noted that there are a group of such certain NNMs in the hybrid quasi-ARMAX model. The hybrid quasi-ARMAX model becomes so flexible (complex) that it is impossible to estimate all of the parameters ($a_i$, $b_i$, $c_i$, $\omega_{ij}$ and $p_j$) from observed data as usual. In order to make the problem feasible, we will appropriately determine the parameter vectors $p_j$ specifying the 'basis functions' in the NNM by using knowledge information, and only estimate $a_i$, $b_i$, $c_i$ and $\omega_{ij}$ from observed data.

#### (1) Determining $\mathrm{p}_j$ and $M$ Using Knowledge Information

The efficient use of knowledge information for determining the parameter vector $p_j$ and the order $M$ plays a key role in the hybrid quasi-ARMAX modeling [37], see Chapter 2 for detail. The follows are some points:

- The least prior knowledge required for determining $p_j$ is the information concerning operating region of $\tilde{\varphi}(t) = [x_i; \; i = 1, ..., r]^T$. That is, $[X_{\min}, X_{\max}]$ should be known for the statement that the operating region is mostly located in $X_{\min} \leq \tilde{\varphi}(t) \leq X_{\max}$.
- For the case where AFSs described by (6.14) are used, if the number of fuzzy sets for variable $x_i$ is denotes as $n_i$, then the number of rules is $M = \prod_{i=1}^{r} n_i$. When $\dim(\tilde{\varphi}(t))$ is large, $M$ will be rather large. Therefore, knowledge information obtained from observed data or the prediction error during the estimation should be used to choose $n_i$ ($i = 1, ..., r$) as small as possible, and to remove some rules by employing interpolation.

#### (2) Estimation of $a_i$, $b_i$, $c_i$ and $\omega_{ij}$

Introduce a parameter vector $\Theta$ and a regression vector $\varphi_{NL}(t)$ defined as

$$\Theta = [\theta^T, \; \omega_{11} \; ... \; \omega_{rM}, \; c_1 \; ... \; c_l]^T \tag{6.18}$$

$$\varphi_{NL}(t) = [\varphi^T(t), \; \varphi^T(t) \otimes \varphi_{\mathcal{N}_f}^T(t), \; e(t-1) \; ... \; e(t-1)]^T \tag{6.19}$$

where $\varphi_{\mathcal{N}_f}^T(t) = [\mathcal{N}_f(p_j, \varphi(t)), \; j = 1, ..., M]$, and the symbol $\otimes$ denotes Kronecker production. Then we can express the hybrid quasi-ARMAX model in a linear regression structure

$$\mathcal{M}: \quad y(t) = \varphi_{NL}^T(t)\Theta + e(t). \tag{6.20}$$

It is well known that the estimation of $\Theta$ for (6.20) can be realized by using an Extended Least Square (ELS) recursive identification algorithm [70, 94].

### 6.2.4 Nonlinear Predictors

We have shown that a sufficiently smooth general nonlinear stochastic systems can be described by the hybrid quasi-ARMAX models (6.3) or (6.16). If we use these hybrid models as nonlinear predictors

$$\hat{y}(t) = \frac{1}{\hat{C}(q^{-1})}\left[\varphi^T(t)\hat{\theta} + \sum_{j=1}^{M}\underbrace{\varphi^T(t)\hat{\Omega}_j}_{ARX}\mathcal{N}_f(\boldsymbol{p}_j, \tilde{\varphi}(t)) + (\hat{C}(q^{-1}) - 1)y(t)\right] \tag{6.21}$$

or

$$\hat{y}(t) = \frac{1}{\hat{C}(q^{-1})}\left[\sum_{j=1}^{M}\underbrace{\varphi^T(t)\hat{\Omega}_j}_{ARX}\mathcal{N}_f(\boldsymbol{p}_j, \tilde{\varphi}(t)) + (\hat{C}(q^{-1}) - 1)y(t)\right], \tag{6.22}$$

a control law can be derived directly from these predictors because the predictors are linear not only in the parameters to be adjusted but also in the one-step past input variable. Based on these features, we will design a STR controller for general nonlinear systems under the framework of linear stochastic control theory.

## 6.3 Adaptive Control of Nonlinear Systems

A difficulty with general nonlinear stochastic systems is that it may be difficult to determine an optimal predictor for the system output [26]. The hybrid quasi-ARMAX models discussed in the previous section can however be used as such optimal predictor to get around the difficulty.

### 6.3.1 STR Controller for Nonlinear Systems

Suppose that the general nonlinear stochastic systems to be controlled is governed by (6.1). The system is further assumed to be able to be modeled by using the hybrid quasi-ARMAX model described by (6.3)-(6.13), which is rewritten as

$$\bar{A}(q^{-1},t)y(t) = q^{-1}\bar{B}(q^{-1},t)u(t) + C(q^{-1})e(t) \tag{6.23}$$

where $y(t)$ is the output at time $t$ ($t = 1, 2, ...$), $u(t)$ the input and $e(t)$ the white noise. $\bar{A}(q^{-1},t)$, $\bar{B}(q^{-1},t)$ and $C(q^{-1})$ are scalar polynomials in backward shift operator $q^{-1}$

$$\begin{array}{l} \bar{A}(q^{-1},t) = 1 + \bar{a}_{1,t}q^{-1} + ... + \bar{a}_{n,t}q^{-n} \\ \bar{B}(q^{-1},t) = \bar{b}_{0,t} + \bar{b}_{1,t}q^{-1} + ... + \bar{b}_{m-1,t}q^{-m+1} \\ C(q^{-1}) = 1 + c_1q^{-1} + ... + c_lq^{-l}. \end{array} \tag{6.24}$$

where $\bar{a}_{i,t}$, $\bar{b}_{i,t}$ are the coefficients consisting of constant parameters ($a_i$, $b_i$) and nonlinear terms ($\Delta a_{i,t}$, $\Delta b_{i,t}$). Moreover, $\bar{b}_{0,t} \neq 0$ is assumed. The index $t$ in the coefficients denote that the coefficients are functions of $\tilde{\varphi}(t)$.

In order to derive a control law for the system to be controlled described by (6.23), we consider the following one-step ahead cost function:

$$J(t+1) = E\left[\frac{1}{2}\left(y(t+1) - y^*(t+1)\right)^2 + \frac{\lambda}{2}u(t)^2\right] \tag{6.25}$$

where $y^*(t+1)$ is a given reference output and $\lambda$ is a weighting factor for the control input. We notice the following two points: (1) the hybrid quasi-ARMAX model described by (6.23) has a similar form as that of linear ARMAX model; (2) the coefficients in the polynomials $\bar{A}(q^{-1},t+1)$ and $\bar{B}(q^{-1},t+1)$ are functions of $\tilde{\varphi}(t+1)$ which does not include $u(t)$ as its element. The second point implies that in the differentiation with respect to $u(t)$, the coefficients can be considered as constants for any given time $t$. Therefore, in a similar way to the linear stochastic control theory [26], the controller minimizing (6.25) with respect to $u(t)$ can be given by

$$\begin{aligned} u(t) &= \frac{\bar{\beta}_{0,t+1}}{\bar{\beta}^2_{0,t+1}+\lambda}\left\{q\left[\bar{\beta}_{0,t+1}-\bar{\beta}(q^{-1},t+1)\right]u(t-1)\right. \\ &\quad +y^*(t+1)+\left[C(q^{-1})-1\right]y^\circ(t+1/t) \\ &\quad \left.-\bar{\alpha}(q^{-1},k+1)y(t)\right\} \end{aligned} \tag{6.26}$$

where $\bar{\alpha}(q^{-1},t)$ and $\bar{\beta}(q^{-1},t)$ are the polynomials in $q^{-1}$ given by

$$\begin{aligned} \bar{\alpha}(q^{-1},t) &= \bar{\alpha}_{0,t}+\bar{\alpha}_{1,t}q^{-1}+\ldots+\bar{\alpha}_{n-1,t}q^{-(n-1)} \\ &= \bar{G}(q^{-1},t) \end{aligned} \tag{6.27}$$

$$\begin{aligned} \bar{\beta}(q^{-1},t) &= \bar{\beta}_{0,t}+\bar{\beta}_{1,t}q^{-1}+\ldots+\bar{\beta}_{m-1,t}q^{-m+1} \\ &= \bar{F}(q^{-1},t)\bar{B}(q^{-1},t) \end{aligned} \tag{6.28}$$

The polynomials $\bar{G}(q^{-1},t)$ and $\bar{F}(q^{-1},t)$ are uniquely determined from

$$C(q^{-1})=\bar{A}(q^{-1},t)\bar{F}(q^{-1},t)+q^{-1}\bar{G}(q^{-1},t) \tag{6.29}$$

where

$$\begin{aligned} &\bar{F}(q^{-1},t)=1 \\ &\bar{G}(q^{-1},t)=\bar{g}_{0,t}+\bar{g}_{1,t}q^{-1}+\ldots+\bar{g}_{n-1,t}q^{-n+1}. \end{aligned} \tag{6.30}$$

On the other hand, $y^\circ(t+1/t)$ in (6.26) denotes the optimal one-step ahead prediction of $y(t+1)$, which is written by

$$C(q^{-1})y^\circ(t+1/t)=\bar{\alpha}(q^{-1},t)y(t)+\bar{\beta}(q^{-1},t)u(t) \tag{6.31}$$

Introduce a coefficient vector $\Theta_t$, and a regression vector $\varphi_c(t)$ defined by

$$\Theta_t=[\bar{\alpha}_{0,t}\ \bar{\alpha}_{1,t}\ \ldots\ \bar{\alpha}_{n-1,t}\ \bar{\beta}_{0,t}\ \bar{\beta}_{1,t}\ \ldots\ \bar{\beta}_{m-1,t}\ c_1\ \ldots\ c_l]^T \tag{6.32}$$

$$\begin{aligned} \varphi_c(t)=[&y(t)\ \ldots\ y(t-n+1)\ u(t)\ \ldots\ u(t-m+1) \\ &-y^\circ(t/t-1)\ \ldots\ -y^\circ(t-l+1/t-l)]^T \end{aligned} \tag{6.33}$$

Then (6.31) can be rewritten into the form

$$y^\circ(t+1/t)=\varphi_c^T(t)\Theta_t. \tag{6.34}$$

We will here call (6.34) a *nonlinear predictor*. In this way, the $\Theta_t$ is a coefficient vector, whose elements stand for functions of $\tilde{\varphi}(t)$ and the parameters of the nonlinear predictor as well as the regulator. It means that the components $\bar{\alpha}_{i,t}$ and $\bar{\beta}_{i,t}$ are not constant parameters but functions of $\tilde{\varphi}(t)$. We will therefore use the hybrid quasi-ARMAX modeling scheme to construct the nonlinear predictor.

### (1) Hybrid Quasi-ARMAX Predictor

Based on the idea of quasi-ARMAX modeling, we divide the components $\bar{\alpha}_{i,t}$ and $\bar{\beta}_{i,t}$ into two parts: constant parameters $\alpha_i$, $\beta_i$ and nonlinear terms $\Delta\alpha_{i,t}$, $\Delta\beta_{i,t}$, we have

$$\bar{\alpha}_{i,t}=\alpha_i+\Delta\alpha_{i,t} \qquad \bar{\beta}_{i,t}=\beta_i+\Delta\beta_{i,t}. \tag{6.35}$$

Now introduce a group of certain NNMs with the form described by (6.13) (without confusion, we here use the same notations for the NNMs) to represent the nonlinear terms

$$\begin{aligned} &\Delta\alpha_{i-1,t}=f_i(\tilde{\varphi}(t)) \quad (i=1,\ldots,n) \\ &\Delta\beta_{j-1,t}=f_{j+n}(\tilde{\varphi}(t)) \quad (j=1,\ldots,m). \end{aligned} \tag{6.36}$$

Then the controller parameter vector $\Theta_c$ to be adjusted are the ARMAX parameters $\{\alpha_i,\ (i = 0, ..., n-1),\ \beta_i,\ (i = 0, ..., m-1),\ c_i,\ (i = 1, .., l)\}$ and the parameters of the NNMs, $\{\omega_{ij},\ (i = 1 ... r,\ j = 1 ... M)\}$,

$$\Theta_c = [\alpha_0 ... \alpha_{n-1}\ \beta_0 ... \beta_{m-1}\ w_{11} ... w_{rM}\ c_1 ... c_l]^T. \tag{6.37}$$

The nonlinear predictor can thus be rewritten as

$$y^\circ(t+1/t) = \varphi^T_{NLC}(t)\Theta_c \tag{6.38}$$

where

$$\varphi_{NLC}(t) = [\varphi^T(t),\ \varphi^T(t) \otimes \varphi^T_{\mathcal{N}_f}(t),\ -y^\circ(t/t-1)\ ...\ -y^\circ(t-l+1/t-l)]^T \tag{6.39}$$

with $\varphi^T_{\mathcal{N}_f}(t) = [\mathcal{N}_f(p_j, \tilde{\varphi}(t)), j = 1, ..., M]$.

### (2) Multi-Model Based Predictor

Next, we use a multi-model based predictor structure. The hybrid quasi-ARMAX model described by (6.16) is actually a multi-model consisting of several local linear ARMAX models. Introduce $\Omega_{cj}$ as

$$\Omega_{cj} = [\alpha_{0j}\ ...\alpha_{(n-1)j}\ \beta_{0j}\ ...\ \beta_{(m-1)j}\ c_1\ ...\ c_l]^T. \tag{6.40}$$

Then the nonlinear predictor (6.34) can expressed as

$$y^\circ(t+1/t) = \sum_{j=1}^{M} \underbrace{\varphi_c^T(t)\Omega_{cj}}_{\text{ARMAX}} \mathcal{N}_f(p_j, \tilde{\varphi}(t)) \tag{6.41}$$

(6.41) can be further expressed as the linear regression structure (6.38), where $\Theta_c$ and $\varphi_{NLC}(t)$ will be given as

$$\Theta_c = [\{\alpha_{0j}\ ...\alpha_{(n-1)j}\ \beta_{0j}\ ...\ \beta_{(m-1)j}\}\ (j = 1, ..., M),\ c_1\ ...\ c_l]^T \tag{6.42}$$

$$\varphi_{NLC}(t) = [\varphi_c^T(t) \otimes \varphi^T_{\mathcal{N}_f}(t),\ -y^\circ(t/t-1)\ ...\ -y^\circ(t-l+1/t-l)]^T \tag{6.43}$$

From the expression (6.41), we can see that the multi-model based predictor is easy to be implemented in a multiresolution way, i.e., the number of ARMAX models can be increased on-line when a higher tracking accuracy is required.

## 6.3.2 Synthesis of the Control Law

The adaptive control law (6.26) based on direct approach can be synthesized by the following way.

Firstly, the parameter estimation of $\Theta_c$ is performed under the criterion

$$\hat{\Theta}_c = \arg\min_{\theta} E[y(t+1) - y^\circ(t+1/t)]^2. \tag{6.44}$$

It is well known that the following recursive least squares algorithm can be used for on-line estimation of $\Theta_c$ ($\lambda_0$ is forgetting factor)

$$\hat{\Theta}_c(t) = \hat{\Theta}_c(t-1) + L(t)\varepsilon'(t) \tag{6.45}$$

$$L(t) = \frac{P(t-1)\psi(t-1)}{\lambda_0 + \psi^T(t-1)P(t-1)\psi(t-1)} \tag{6.46}$$

$$P(t) = \frac{1}{\lambda_0}\left[P(t-1) - L(t)\psi^T(t-1)P(t-1)\right] \tag{6.47}$$

where

$$\psi(t) = \left(\frac{\partial y^\circ(t+1/t)}{\partial \Theta_c}\right)^T = \varphi_{NLC}(t) \tag{6.48}$$

$$\varepsilon'(t) = \begin{cases} \varepsilon(t) - \delta & \text{if } \varepsilon(t) > \delta \\ 0 & \text{if } |\varepsilon(t)| \le \delta \\ \varepsilon(t) + \delta & \text{if } \varepsilon(t) < -\delta \end{cases} \tag{6.49}$$

$$\varepsilon(t) = y(t) - \hat{y}(t) \tag{6.50}$$

$$\hat{y}(t) = \hat{\varphi}_{nlc}^{T}(t-1)\hat{\Theta}_c(t-1) \tag{6.51}$$

and where $\hat{\varphi}_{nlc}(t-1)$ is $\varphi_{NLC}(t-1)$ whose elements $y^{\circ}(t-i/t-i-1)$'s are replaced by $\hat{y}(t-i)$. In the algorithm, the *dead zone* $\delta$ in (6.49) will be assigned with appropriate positive value. Note that the *dead zone* is introduced to make the algorithm robust to unmodeled dynamics, which has been known to be very effective [50, 102].

Secondly, the adaptive control law (6.26) is synthesized based on *Certainty Equivalence Principle* by replacing $\Theta_t$ and $y^{\circ}(t+1/t)$ with the current calculation $\hat{\Theta}_t$ and the adaptive prediction $\hat{y}(t+1)$, respectively.

### 6.3.3 Implementation of the Adaptive Algorithm

It should be noticed that the parameters $\alpha_i$ and $\beta_i$ denote the linear part of the predictor, with which the predictor can gives a reasonable performance in many cases. Furthermore, the $\alpha_i$ and $\beta_i$ are redundant parameters, by which we mean that the $\alpha_i$ and $\beta_i$ are not independent of $\omega_{ij}$, i.e., the role they play can be replaced by that of $\omega_{ij}$. Therefore, the convergence property can be improved if the adaptive algorithm is implemented in the following two stages:

(1) Fix $\omega_{ij} = 0$ and adjust $\alpha_i$, $\beta_i$, $c_i$ for appropriate steps (For the case where a multi-model based predictor is used, set $M = 1$). This stage accounts for that the nonlinear systems is controlled using a linear robust adaptive controller.

(2) Continue the adaptive algorithm with adjusting all parameters ($a_i$, $b_i$, $c_i$, $\omega_{ij}$).

## 6.4 Numerical Simulations

In this section, we will apply the proposed nonlinear controller to a variety of nonlinear systems to test the effectiveness. In the literature of nonlinear adaptive control incorporating neural networks, system disturbances are rarely considered in the numerical simulations [76, 102, 34]. However, system disturbances are inevitable in practice. In our simulations, we will assume that the systems to be controlled have disturbances.

### 6.4.1 A System with Various Nonlinear Elements

*Example* 1: The unknown system to be controlled is shown in Fig. 6.3, in which the linear part of system is described by

$$G(q^{-1}) = \frac{0.7q^{-1} - 0.68q^{-2}}{1 - 1.72q^{-1} + 0.74q^{-2}} \tag{6.52}$$

while the nonlinear element is a backlash with width=4 and height=4. Namely

$$\bar{z}(t) = \begin{cases} u(t) - 0.5 \times \text{width} & \text{if } u(t) \ge u(t-1) \\ u(t) + 0.5 \times \text{width} & \text{if } u(t) < u(t-1) \end{cases} \tag{6.53}$$

$$z(t) = \begin{cases} \bar{z}(t) & \text{if } |\bar{z}(t)| \le 0.5 \times \text{height} \\ \text{sign}(\bar{z}(t)) \times \text{height}/2 & \text{if } |\bar{z}(t)| > 0.5 \times \text{height}. \end{cases} \tag{6.54}$$

And the system disturbance is assumed to be white noise, $v(t) \in N(0, 0.001)$

The desired output of system is assumed to be

$$y^{*}(t) = -0.2y^{*}(t-1) + 0.63y^{*}(t-2) + r(t-1) + 0.8r(t-2) \tag{6.55}$$

where $r(k) = \sin(2\pi t/25)$.

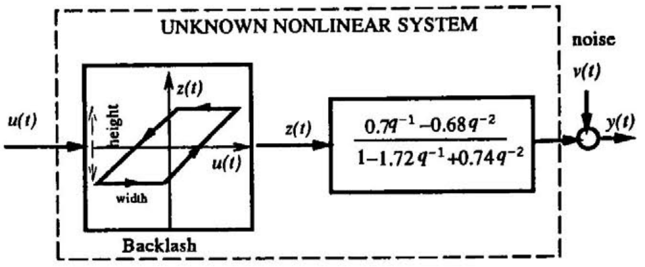


Figure 6.3: Unknown nonlinear system for Example 1

The hybrid quasi-ARMAX predictor with $n = m = 2$, $l = 1$ is used for the controller. From the results obtained by a linear robust STR controller (fixing $\omega_{ij} = 0$), we choose $X_{min}$=[-4 -4 -4 -4] and $X_{max}$=[4 4 4 4]. Assuming that no other useful information available, using Hint B [2] we choose $n_1 = n_3 = 4$, $n_2 = n_4 = 1$ for $\tilde{\varphi}(t)$, which gives $M = 16$. The prediction model then has 69 parameters to be adjusted. We further choose *weighting factor* $\lambda = 0.001$ for the control law, and *forgetting factor* $\lambda_0 = 0.995$, *dead zone* $\delta = 0.1$ for the robust adaptive algorithm.

First, fix $\omega_{ij} = 0$ and adjust $\alpha_i$, $\beta_i$, $c_i$ for 600 steps. This accounts for controlling the system using a linear robust controller. Figure 6.4(a) shows the result. The control performance is not impressive. Then adjust all parameters from the 601*st* step. Figure 6.4(b) shows the results of first 100 steps when all parameters are adjusted. At beginning the tracing error is rather large, but as the controller is adapted, the tracing error is getting smaller. Figure 6.4(c) shows the result of 3901 to 4000 steps. We can see that after being adapted for appropriate steps the proposed nonlinear robust adaptive controller has better performance than the linear one.

Next, we will use the multi-model based predictor to control the system. In the simulation, the number of local ARMAX model ($M$) is first chosen rather small, and then increases on-line to achieve higher tracking accuracy. For simplicity, we also use the knowledge information that the system is linear with respect to $y(t-i)$ and choose $n_1 = n_2 = n_4 = 1$, $n_3 = M$. In the first 600 steps, $M$ was set to 1, then $M$ was changed to 7 at 601st step, and then $M$ was changed to 13 at the 4001st step, finally $M$ was changed to 25 at the 6001st step. The results of $M = 1, 7, 13, 25$ are shown in Fig. 6.5(a)(b)(c)(d), respectively. We can see that when $M = 7$ the controlled system could not trace the desired output well, while $M = 13$ or larger, the controlled system could follow the desired output well.

*Example* 2: The unknown system considered is the same as that of *Example* 1 except that the nonlinear element is a dead zone with width=4, see Fig. 6.6.

The controller used for this example is the same as *Example* 1. Since the unmodeled dynamics may be smaller, the *dead zone* $\delta$ is chosen to be 0.01.

Figure 6.7 shows the simulation results. The proposed nonlinear robust STR controller has better performance.

### 6.4.2 A System with Indivisible Nonlinearity

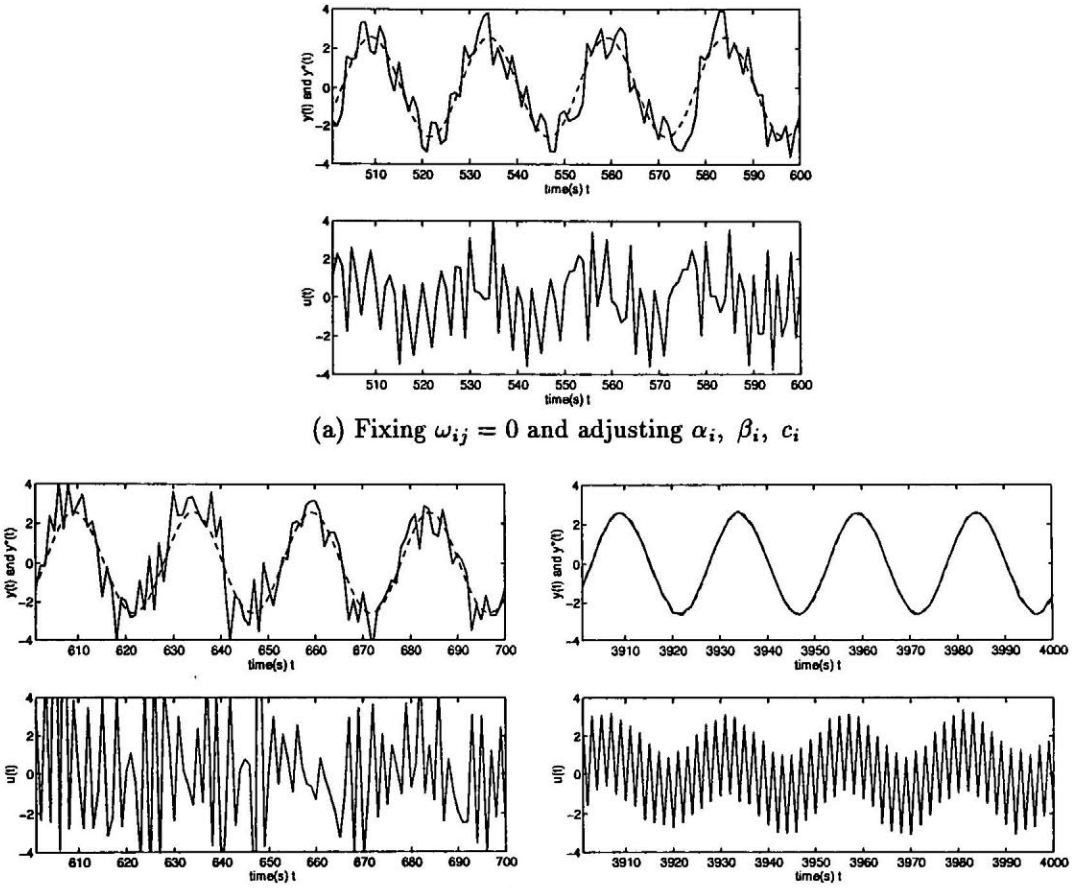
*Example 3*: In this example, the following nonlinear system treated in [76] is considered, which contains indivisible nonlinearity

$$y(t) = f[y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)] + v(t) \qquad (6.56)$$

where the disturbance $v(t) \in N(0, 0.001)$ is white noise, and

$$f[x_1, x_2, x_3, x_4, x_5] = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2}.$$

[2] Referred to Section 2.2.3 in Chapter 2 for the detail about the Hint A,B,C.



(a) Fixing $\omega_{ij} = 0$ and adjusting $\alpha_i$, $\beta_i$, $c_i$

(b) Adjusting all parameters ($\alpha_i$, $\beta_i$, $c_i$, $\omega_{ij}$)

(c) Controller adapted for 4000 steps

Figure 6.4: Results (a) (b) and (c) for *Example 1*: (*Upper diagram*) Controlled output $y(t)$ (solid lines) and desired output $y^*(t)$ (dashed-lines); (*Lower diagram*) Control input $u(t)$.

The desired output is assumed to be

$$y^*(t) = 0.4493y^*(t-1) + 0.57r(t-1) \tag{6.57}$$

where $r(t) = \sin(2\pi t/25)$.

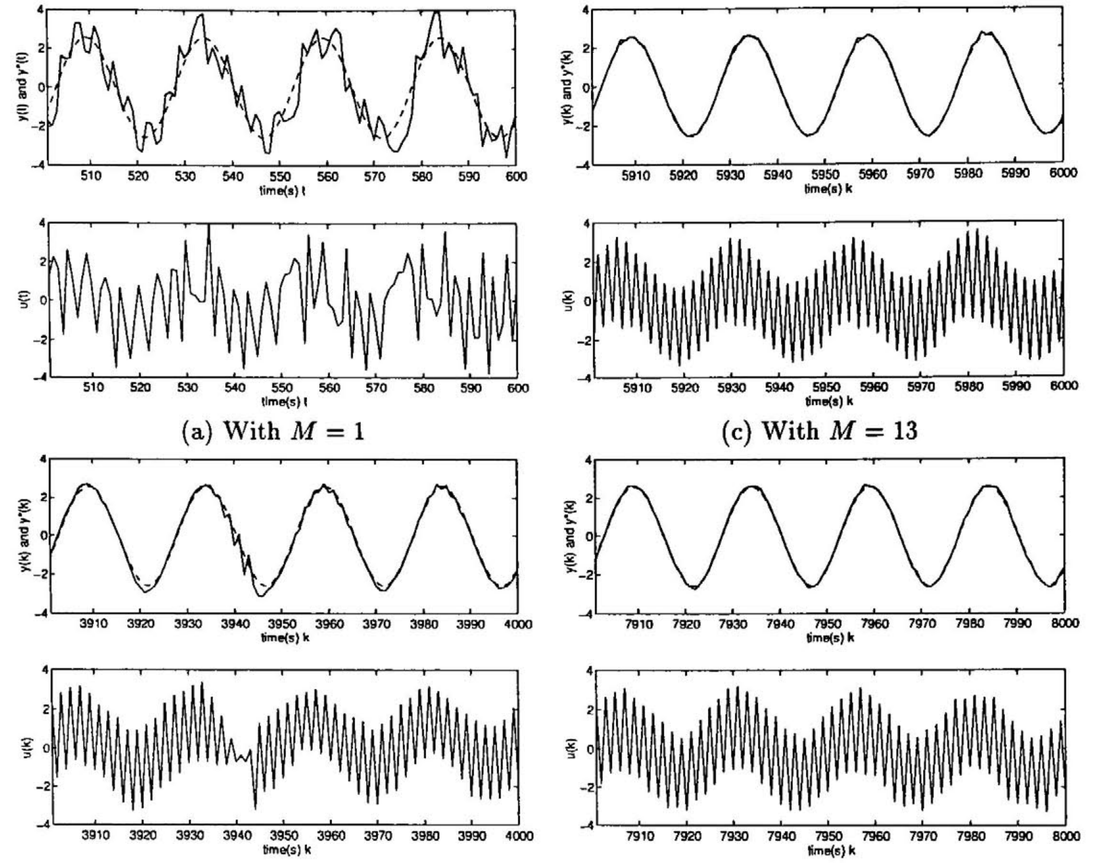
The hybrid quasi-ARMAX predictor with $n = 3$, $m = 2$, $l = 1$ is used for the controller, which gives the best regression vector for (6.56). $X_{min}$=[-1 -1 -1 -2 -2] and $X_{max}$=[1 1 1 3 3] are chosen using the information obtained from the linear robust controller. Since the system is linear with respect to $u(t-i)$, we choose $n_1 = n_2 = n_3 = 3$, $n_4 = n_5 = 1$ for $\bar{\varphi}(t)$ using Hint A. Then we obtain $M = 18$ by removing some nodes based on Hint C. The prediction model then has 96 parameters to be adjusted. We further choose *weighting factor* $\lambda = 0.001$ for the control law, and *forgetting factor* $\lambda_0 = 0.995$, *dead zone* $\delta = 0.002$ for the robust adaptive algorithm.

Figure 6.8 shows the simulation results of linear robust STR control and the proposed nonlinear robust STR control. We can see that again the proposed nonlinear controller has better performance.

### 6.4.3 A System uncontrollable via Local Linearized Model

*Example* 4: The nonlinear system is described by

$$y(k) = \frac{y(t-1)}{1+y^2(t-1)} + u^3(t-1) + v(t), \tag{6.58}$$



(a) With $M = 1$

(b) With $M = 7$

(c) With $M = 13$

(d) With $M = 25$

Figure 6.5: Results for *Example 1* using multi-model based predictor: (*Upper diagram*) Controlled output $y(t)$ (solid lines) and desired output $y^*(t)$ (dashed-lines); (*Lower diagram*) Control input $u(t)$.

which can not be controlled via local linearized model because if (6.58) is linearized at $u = y = 0$, $y(t) = y(t-1)$ will hold. This will make it difficult to control the system near $u = y = 0$. The system disturbance $v(t) \in N(0, 0.001)$ is white noise.

The desired output in this example is assumed to be

$$y^*(t) = 0.6y^*(t-1) + r(t-1) \tag{6.59}$$

where $r(t) = \sin(2\pi t/25) + \sin(2\pi t/10)$.

The hybrid quasi-ARMAX predictor with $n = m = 2$, $l = 2$ is used for the controller. Using the information obtained from the result of linear control, we choose $X_{min}$=[-4 -4 -2 -2] and $X_{max}$=[4 4 2 2]. Since no further useful information available, we choose $n_1 = n_3 = 5$, $n_2 = n_4 = 1$, which gives $M = 25$. The predictor obtained thus contains 106 parameters to be adjusted. We further choose *weighting factor* $\lambda = 0.001$ for the control law, and *forgetting factor* $\lambda_0 = 0.995$, *dead zone* $\delta = 0.3$ for the robust adaptive algorithm.

Figure 6.9 shows the simulation results of linear robust STR control and the proposed nonlinear robust STR control. The system is rather difficult to control. The proposed nonlinear controller performs quite well.

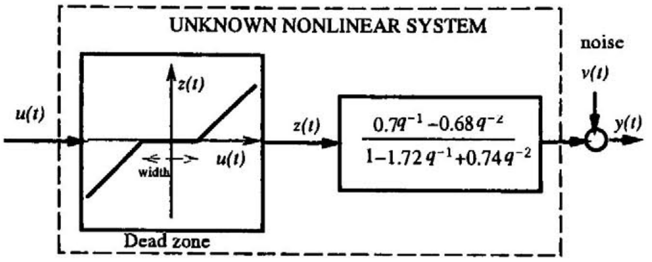
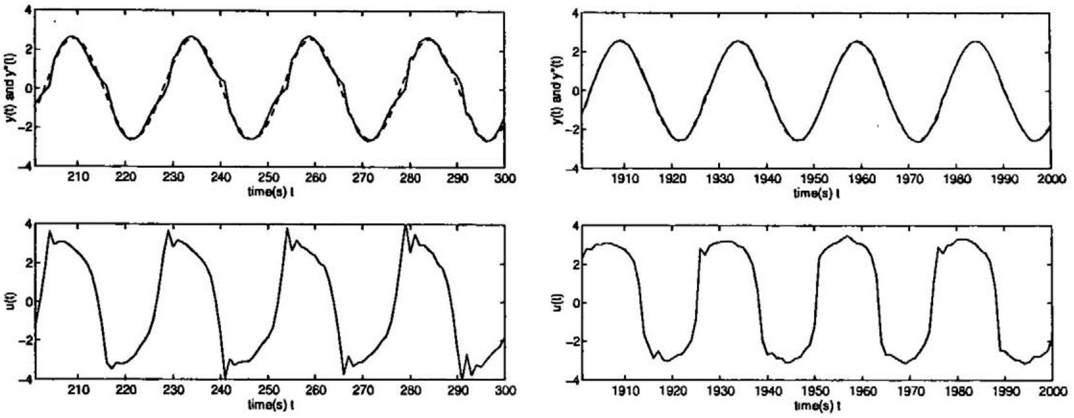


Figure 6.6: Unknown nonlinear system for Example 2



(a) Fixing $\omega_{ij} = 0$ and adjusting $\alpha_i$, $\beta_i$, $c_i$ (b) Adjusting all parameters ($\alpha_i$, $\beta_i$, $c_i$, $\omega_{ij}$)

Figure 6.7: Results (a) and (b) for *Example 2*: (*Upper diagram*) Controlled output $y(t)$ (solid lines) and desired output $y^*(t)$ (dashed-lines); (*Lower diagram*) Control input $u(t)$.

## 6.5 Discussions and Conclusions

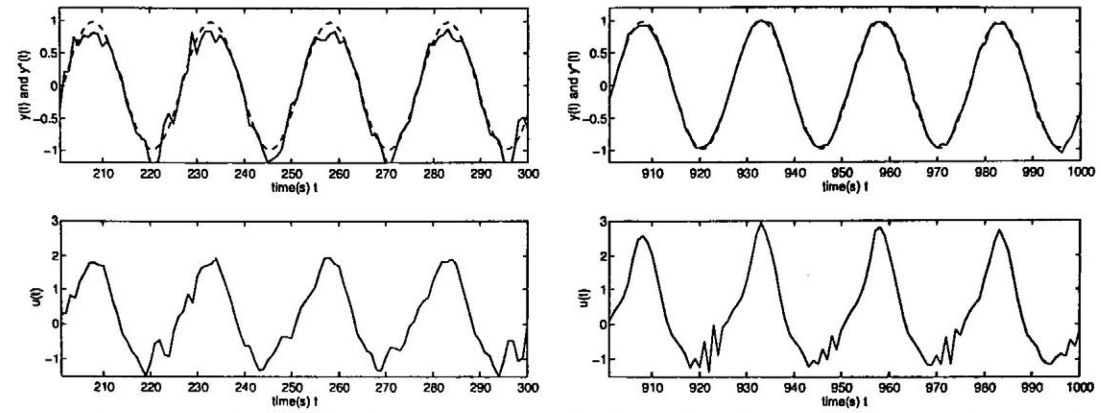
In this chapter, we have proposed a hybrid quasi-ARMAX modeling scheme for control design. The scheme has the following distinctive features:

- The obtained hybrid quasi-ARMAX prediction model has a similar structure as a linear ARMAX model.
- The model is linear not only in the parameters to be adjusted but also in the one-step past input ($u(t-1)$) so that a control law can be derived directly from the hybrid quasi-ARMAX predictor.

These features makes it possible to do control design of nonlinear systems under the framework of linear control theory. As an example, we have developed a STR adaptive control for general nonlinear stochastic systems based on the linear stochastic control theory [26]. The effectiveness of the proposed controller has been examined by applying it to a variety of nonlinear stochastic systems.

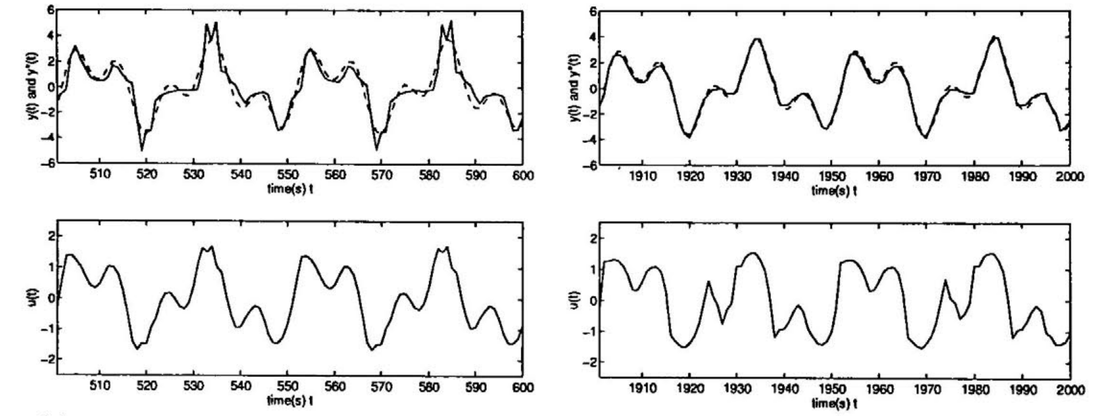
In the literature, there are many publications concerning with the use of neural networks for adaptive control of nonlinear systems. The control systems based on neural networks are usually rather complicated, sensitive to noise, and the convergence properties are not satisfying because the neural networks are usually large and nonlinear in the parameters to be adjusted. Some authors suggests to use neural networks only as nonlinear compensators, which makes the convergence properties improved greatly [102, 34]. Compared with those existing controllers for general nonlinear systems, the STR controller proposed in this chapter has the following advantages:

- It has a simple structure, which is similar in form to the linear STR controller. In fact, it can be considered as an extension of linear STR controller.



(a) Fixing $\omega_{ij} = 0$ and adjusting $\alpha_i$, $\beta_i$, $c_i$ (b) Adjusting all parameters ($\alpha_i$, $\beta_i$, $c_i$, $\omega_{ij}$)

Figure 6.8: Results (a) and (b) for *Example 3*: (*Upper diagram*) Controlled output $y(t)$ (solid lines) and desired output $y^*(t)$ (dashed-lines); (*Lower diagram*) Control input $u(t)$.



(a) Fixing $\omega_{ij} = 0$ and adjusting $\alpha_i$, $\beta_i$, $c_i$ (b) Adjusting all parameters ($\alpha_i$, $\beta_i$, $c_i$, $\omega_{ij}$)

Figure 6.9: Results (a) and (b) for *Example 4*: (*Upper diagram*) Controlled output $y(t)$ (solid lines) and desired output $y^*(t)$ (dashed-lines); (*Lower diagram*) Control input $u(t)$.

- It has better convergence properties, since the prediction model is linear in the parameter to be adjusted.
- It is not so sensitive to noise, because it is a stochastic STR adaptive controller.

Therefore, the controller will be useful in real applications.

On the other hand, a robust adaptive scheme using *dead zone* has been employed to improve the robustness and the stability of the controller. G. Feng (1995) [18] has shown that an adaptive controller based on linear model using such a robust adaptive scheme can be guaranteed to have global stability and convergence and to be robust to unmodeled dynamics. However, a proof of global stability and convergence for an adaptive controller based on nonlinear black-box models is not so far available. We are on investigating this problem concerned with our approach.

# Chapter 7

# Conclusions

In this final chapter, we will give a summary for whole thesis.

Linear system theory (the theory of system identification, system analysis and control design based on linear models) are very well developed. There exist many results which can be applied to an obtained linear models. The main theme of this thesis has been to develop a hybrid black-box modeling scheme so that the obtained nonlinear black-box models contain not only the linearity properties which are useful for applications, but also have good flexibility which is needed to deal with various nonlinear systems. Investigations have made to do identification, system analysis and control design of nonlinear systems under the framework of linear system theory, on the basis of the new model structure. The main work of the thesis has been described in Chapter 2, 3, 4, 5 and 6.

In Chapter 2, a hybrid quasi-linear black-box modeling scheme has been proposed, in which a group of certain NNMs are incorporated into a linear structure. Main efforts have been made to find better compromises to the trade-off between model flexibility and model simplicity. The aim is to obtain a nonlinear black-box model equipped with a linear structure, flexibility and simplicity. It has been shown that in the case of using the hybrid model for 'prediction' or 'simulation' of nonlinear systems, it has the following distinctive feature:

- It has a compatible flexibility to the existing nonlinear black-box models based on neural networks or wavelet networks etc.
- It has better properties for dealing with correlated system disturbance (noise).
- The estimator for its parameter estimation could find global minimum more reliably by using our proposed hybrid identification method.

The work in this chapter also shows that

(1) the gap between the existing linear and nonlinear black-box models can be filled by developing a hybrid modeling scheme, in which knowledge information is used efficiently;

(2) with the new hybrid model structure, the techniques based on well developed linear system theory could be extended to nonlinear systems.

In Chapter 3, a hybrid identification method using genetic algorithms has been proposed based on using a Non-Standard GA (NSGA) for searching a good initial value. The NSGA is proposed by introducing a new GA operator named as *development* inspired by the fact that living beings adapt themselves to their environment. The hybrid identification method is typically suitable for solving multimodal problems resulted from noise models. It combines the reliability properties of the GAs with the accuracy of optimization-based method, while requiring a computation time only slightly higher than the latter. The proposed hybrid identification method is significant, since

(1) it provides a better compromise to the trade-off among the accuracy, reliability and convergent rate in an identification algorithm;

(2) it gives a solution to the multimodality problem resulted from noise model.

The fault detection scheme based on the use of Kullback discrimination information for model discrimination is known to be very effective for fault detection, but it was so far based on two important assumptions which are never in real applications. In Chapter 4 and 5, the assumptions has been relaxed so that the fault detection scheme is extended to the case of unmodeled dynamics and the case of nonlinear systems. In Chapter 4, a KDI-based robust fault detection scheme has proposed. Two robust identification methods have been suggested for robust fault detection. The KDI have been analyzed into a feasible form consisting of four meaningful terms. Fault decision schemes have been developed for robustness realization. In Chapter 5, two approaches, a robust fault detection like approach and a multi-model based approach, have been proposed for fault detection of nonlinear systems on the basis of the hybrid quasi-linear black-box model. The work in these two chapters shows that

(1) the two assumptions so far made in the KDI-based fault detection scheme can be relaxed, so that the fault detection scheme becomes more practicable;

(2) it is possible to extend the existing system analysis approach to nonlinear systems with the proposed hybrid quasi-linear black-box model structure.

In Chapter 6, the hybrid quasi-ARMAX modeling scheme has been further modified so that it is favorable to control design. Our efforts have been made to construct the hybrid model in such way that it is linear not only in the parameters to be adjusted but also in the one-step past input variable ($u(t-1)$) which should be synthesized in a control system. It is distinctive that with the hybrid quasi-ARMAX model structure, one could do control design of nonlinear systems under the framework of linear control theory. As an example, a STR adaptive controller has been designed for general nonlinear stochastic systems under the framework of linear stochastic control theory. The effectiveness of the controller has been confirmed through numerical simulations.

### Topics for further research

This thesis has reached its goal and it also motivates several topics for further research.

In hybrid quasi-linear black-box modeling scheme, the ability to incorporate knowledge information plays key role for obtaining the better properties. However in this thesis, we could only give some suggestions for how to use knowledge information. In order to made the modeling method less heuristically dependent, an algorithm should be developed to incorporate knowledge information effectively and efficiently. This gives an alternative perspective for *intelligent identification.*

In KDI-based robust fault detection scheme, the KDI has been analyzed into a feasible form consisting of four meaningful terms. What information these individual terms can provide and how to use the information in the fault detection and fault analysis are needed to be further investigated theoretically and experimentally.

In adaptive control of nonlinear systems, we have only shown that it is possible to do control design of general nonlinear systems under framework of linear control theory. The application of the hybrid quasi-ARMAX model to control design is just at its beginning stage. We believe that it is a perspective method. Further research may be directed toward the investigations of the tracing ability, convergence properties, global stability of control system, etc.

# Appendix A

# A Review of Black-Box Models

In this appendix, we will give a review for black-box models and highlight their distinctive features. In black-box models, no physical insight is available or used, but the chosen model structure belongs to families that are known to have good flexibility and have been "successful in the past". In the literature, there are linear black-box models and nonlinear black-box models.

## A.1 Linear Black-Box Models

Linear system theory is very well developed and there exist many results which can be applied to the obtained linear model. Let us first review the linear black-box models and highlight their distinctive features. When extending the linear black-box models to the nonlinear ones, we will try to retain those features.

### A.1.1 General Linear SISO Black-Box Model

The general form of a linear SISO black-box model can be described by (A.1) [94]

$$\begin{array}{c} y(t) = G(q^{-1},\theta)u(t) + H(q^{-1},\theta)e(t) \\ Ee(t)e(s) = \sigma^2\delta_{t,s} \end{array} \tag{A.1}$$

where $y(t)$ is the output at time $t$, $u(t)$ is the input, $e(t)$ is a sequence of independent and identically distributed (iid) random variable referred to as *white noise*, $G(q^{-1},\theta)$ and $H(q^{-1},\theta)$ are rational functions in the backward shift operator $q^{-1}$, for example, $q^{-1}u(t) = u(t-1)$.

A general parameterization for (A.1) can be given as

$$y(t) = \frac{B(q^{-1},\theta)}{A(q^{-1},\theta)F(q^{-1},\theta)}u(t) + \frac{C(q^{-1},\theta)}{A(q^{-1},\theta)D(q^{-1},\theta)}e(t) \tag{A.2}$$

where

$$\begin{array}{l} A(q^{-1},\theta) = 1 + a_1q^{-1} + ... + a_nq^{-n} \\ B(q^{-1},\theta) = b_1q^{-1} + ... + b_mq^{-m} \\ C(q^{-1},\theta) = 1 + c_1q^{-1} + ... + c_lq^{-l} \\ D(q^{-1},\theta) = 1 + d_1q^{-1} + ... + d_{n_d}q^{-n_d} \\ F(q^{-1},\theta) = 1 + f_1q^{-1} + ... + f_{n_f}q^{-n_f} \end{array} \tag{A.3}$$

Using 'pseudo-linear' regression form [70], we can express (A.2) as

$$y(t) = \varphi_e^T(t)\theta_e + e(t) \tag{A.4}$$

where

$$\theta_e = [a_1\ ...\ a_n\ b_1\ ...\ b_m\ c_1\ ...\ c_l\ d_1\ ...\ d_{n_d}\ f_1\ ...\ f_{n_f}]^T \tag{A.5}$$

and the regression vector $\varphi_e(t)$ consists of several components given by

1) $y(t-i)$, $i = 1, ..., n$ (associated with the $A$-polynomial)

2) $u(t-i)$, $i = 1, ..., m$ (associated with the $B$-polynomial)

3) $\varepsilon(t-i) = y(t-i) - \hat{y}(t-i)$, $i = 1, ..., l$. Prediction errors (associated with the $C$-polynomial)

4) $\varepsilon_u(t-i) = y(t-i) - \hat{y}_u(t-i)$, $i = 1, ..., n_d$. Simulation errors (associated with the $D$-polynomial)

5) $\hat{y}_u(t-i) = A(q^{-1}, \hat{\theta})y(t-i), i = 1, ..., n_f$. Simulated outputs from past $u$ only (associated with the $F$-polynomial)

On the other hand, a linear state-space model

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{A.6}$$

can also be described as a pseudo-linear regression form [89]. The states are introduced as virtual outputs generated only to obtain the regressors. Define

$$Y(t) = \begin{bmatrix} y(t) \\ x(t+1) \end{bmatrix} \tag{A.7}$$

Then (A.6) can be written as

$$Y(t) = \begin{bmatrix} C & D \\ A & B \end{bmatrix} \begin{bmatrix} Y_R(t-1) \\ u(t) \end{bmatrix} \tag{A.8}$$

where $Y_R(t-1)$ is the last components of $Y(t-1)$ corresponding to $x(t)$. A state-space model can, hence, also be described as a special model structure with a certain choice of the listed regressors. The virtual outputs are constructed only to be used as regressors. This implies that it might be possible to obtain a more efficient model with a smaller number of regressors by using a state-space model.

### A.1.2 Several Special Cases

Several special cases of (A.2) have been so successful in past applications that they have been given own names.

(1) FIR model

$A = C = D = F = 1$

$$y(t) = B(q^{-1}, \theta)u(t) + e(t) \tag{A.9}$$

(2) ARX model

$C = D = F = 1$

$$A(q^{-1}, \theta)y(t) = B(q^{-1}, \theta)u(t) + e(t) \tag{A.10}$$

(3) Output-Error (OE) model

$A = C = D = 1$

$$y(t) = \frac{B(q^{-1}, \theta)}{F(q^{-1}, \theta)}u(t) + e(t) \tag{A.11}$$

(4) ARMAX model

$D = F = 1$

$$A(q^{-1}, \theta)y(t) = B(q^{-1}, \theta)u(t) + C(q^{-1}, \theta)e(t) \tag{A.12}$$

(5) Box-Jenkins (BJ) model

$A = 1$

$$y(t) = \frac{B(q^{-1}, \theta)}{F(q^{-1}, \theta)}u(t) + \frac{C(q^{-1}, \theta)}{D(q^{-1}, \theta)}e(t) \tag{A.13}$$

In addition to the model structures above, there are AR model and ARMA model for time series modeling which also can be considered as the special cases of (A.2).

### A.1.3 Linear Structure and Simplicity

The distinctive features of a linear black-box model are certainly associated with the term of 'linear', which includes two concepts:

**(1) Linear Structure**

The first concept concerns the way in which $y(t)$ depends on past data. A linear black-box model can be expressed in linear regression form $y(t) = \varphi_e^T(t)\theta_e$. Especially, the regression vector $\varphi_e(t)$ depends on the measured data in a linear fashion, which allows one to build a control law easily from the identified model. Moreover, the parameter $\theta$ in the linear structure usually has useful physical interpretations.

**(2) Linear in the Parameters (LIP)**

The second concept concerns models that are *linear in the parameters* $\theta_e$ to be estimated (sometimes abbreviated to LIP [94]), so that $y(t)$ depends linearly on $\theta_e$. The unknown parameters $\theta_e$ is usually estimated by optimizing a performance criterion such as loss function. When the model is LIP, the performance criterion function will be simple with respect to $\theta_e$, so that the estimation becomes simple.

## A.2 Nonlinear Black-Box Models

When a linear black-box model is used to identify an unknown system, the unknown system is usually assumed to be linear. Since this is never true in real applications, a nonlinear black-box model may sometimes be needed to achieve enough accuracy. This section will give a brief summary for the nonlinear black-box models.

### A.2.1 General Expression

A nonlinear black-box structure for a dynamics system is a model structure that is prepared to describe virtually any nonlinear dynamics. There has been considerable recent interest in this area with structures based on neural networks [90, 76], radial basis function networks [12, 98, 77, 27], wavelet networks [117], hinging hyperplanes [9], as well as wavelet transform based methods [101, 7, 16], and models based on fuzzy sets and fuzzy rules [112, 109]. It has been pointed out that these nonlinear structures can be seen as a concatenation of a mapping from observed data to a regression vector $\varphi(t)$ and a nonlinear mapping $g(\cdot)$ from the regressor space to the output space [92, 46], namely,

$$y(t) = g\left(\varphi(t), \theta\right) + v(t) \tag{A.14}$$

where $g(\cdot)$ is some nonlinear function parameterized by $\theta$, $\varphi(t)$ is the regression vector with the same components as those in the linear case, typically $\varphi(t) = [y(t-1) \ldots y(t-n)\ u(t-1) \ldots u(t-m)]^T$ for ARX case, and $v(t)$ is an additive disturbance term.

For the nonlinear mapping $g(\cdot)$, it is natural to think of the parameterized function family as function expansions:

$$g\left(\varphi(t), \theta\right) = \sum_j \omega_j g_j\left(\varphi(t)\right) \tag{A.15}$$

where $g_j$ is referred as *basis functions*, since the role they play in (A.15) is similar to that of a functional space basis. In some particular situations, they do constitute a functional basis. Typical examples are wavelet bases [46].

A lot of choices can be considered for the basis function $g_j$. However, most well known nonlinear black-box model structures are composed of $g_j$ obtained by parameterizing a single 'mother basis function' that we generically denote by $\mathcal{N}_f(x)$. In this way, a nonlinear black-box model can be expressed generally as

$$y(t) = \sum_j \omega_j \mathcal{N}_f\left(\boldsymbol{p}_j, \varphi(t)\right) + v(t) \tag{A.16}$$

where $\omega_j$'s are coordinate parameters and $\boldsymbol{p}_j$'s are scale and position parameter vectors specifying the 'basis functions'.

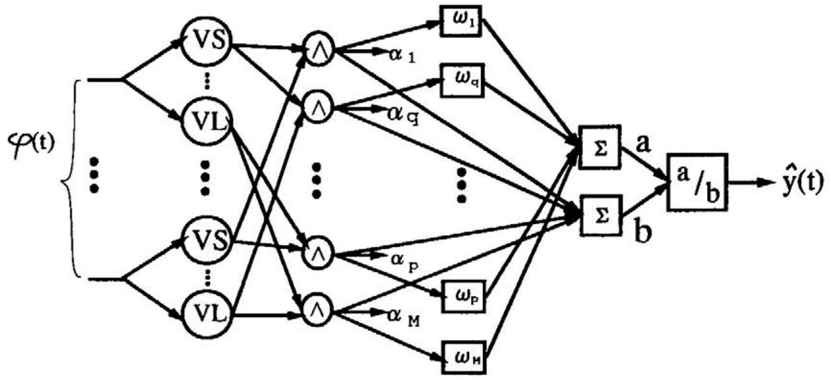


Figure A.1: One of the simplest adaptive fuzzy systems

### A.2.2 Several Nonlinear Black-Box Models

We will discuss several well known nonlinear black-box models.

#### (1) Adaptive Fuzzy System (AFS)

Fuzzy system usually consists of four principal elements: *fuzzification interface*, *knowledge base (fuzzy rule and data base)*, *fuzzy inference machine* and *defuzzification interface* [64, 65]. Recent studies show that fuzzy system can be developed into an adaptive way [99, 43, 108]. Figure A.1 shows one of the simplest adaptive fuzzy systems, which can be explicitly expressed as (A.17), see [110, 111] for details.

$$y(t) = \frac{\sum_{j=1}^{M} \omega_j \left( \wedge_{i=1}^{r} \mu_{A_i^j}(x_i(t)) \right)}{\sum_{j=1}^{M} \left( \wedge_{i=1}^{r} \mu_{A_i^j}(x_i(t)) \right)} + v(t) \tag{A.17}$$

where $r = \dim(\varphi(t))$, $\wedge$ is the minimum operator, $M$ is the number of rules, $x_i$'s are the elements of $\varphi(t)$, and $\mu_{A_i^j}$ is the membership function of fuzzy set $A_i^j$, which may, for example, be given as

$$\mu_{A_i^j}(x_i(t))) = \alpha_i^j exp \left[ -\frac{1}{2} \left( \frac{x_i - \bar{x}_i^j}{\sigma_i^j} \right)^2 \right] \tag{A.18}$$

where $\alpha_i^j$, $\bar{x}_i^j$ and $\sigma_i^j$ together with $\omega_j$ are adjustable parameters.

#### (2) Radial Basis Function Networks (RBFN)

Hartman et al. (1990)[30] has shown that the radial basis function network is capable of approximating arbitrarily well any continuous function defined on a convex, compact set. For MISO system (see Fig. A.2), it can be described as

$$y(t) = \sum_{j=1}^{M} \omega_j h \left( \mathbf{c}_j, \sigma^2, \varphi(t) \right) + v(t) \tag{A.19}$$

where $M$ is the number of radial basis function and $h$ is the radial basis function, for instance,

$$h \left( \mathbf{c}_j, \sigma^2, \varphi(t) \right) = exp \left\{ -\| \varphi(t) - \mathbf{c}_j \|_{\sigma^2}^2 \right\} \tag{A.20}$$

where $\omega_j$, $\mathbf{c}_j$ and $\sigma^2$ are adjustable parameters.

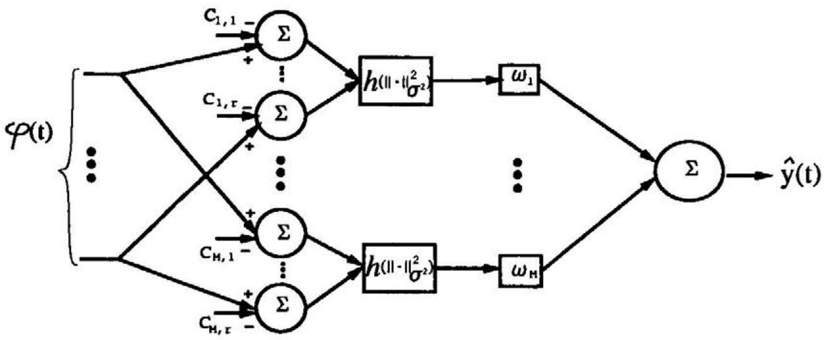


Figure A.2: A MISO radial basis function network

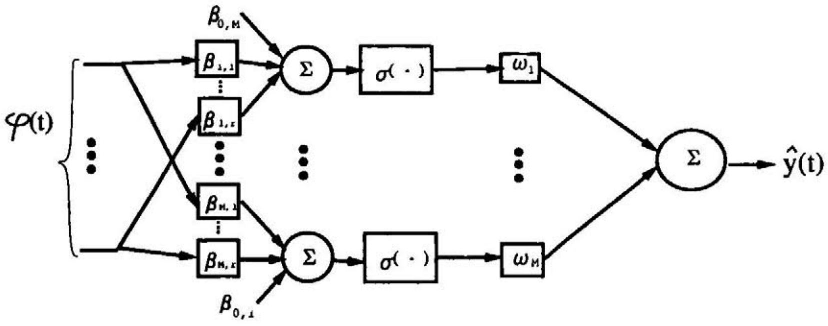


Figure A.3: A single hidden layer neural network

### (3) Neural Networks (NN)

Many researches have been done in using Neural Networks as a nonlinear black-box model for identifying nonlinear dynamic systems [11, 113, 23, 31, 66]. For a single hidden layer neural network shown in Fig. A.3, it can be expressed as

$$y(t) = \sum_{j=1}^{M} \omega_j \sigma(\beta_j^{\prime}\varphi(t) + \beta_{0j}) + v(t) \tag{A.21}$$

where $\sigma(\cdot)$ is the sigmoid function, for example

$$\sigma(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{A.22}$$

and $M$ is the number of neurons, $(\omega_j,\ \beta_j,\ \beta_{0j})$ are adjustable parameters.

There are still many other well-known nonlinear black-box models such as '**wavelet network**', '**B-splines based model**', '**hinging hyperplanes based model**', and so on, see [92, 46] for comprehensive discussions.

## A.2.3 Model Flexibility

We can see that the nonlinear black-box structures summarized above are basically well-known artificial intelligent (AI) techniques which have achieved great success in many areas such as neural network in Pattern Recognition, wavelet transform in Multiresolution Analysis and fuzzy system in Modeling

Heuristic Systems. These approaches have powerful function approximation abilities. Several authors have shown that neural networks, fuzzy systems and radial basis function networks can approximate any continuous functions with an arbitrary accuracy [35, 111, 30]. Because these techniques also have the ability to encompass truly nonlinear behaviors of dynamic systems, they have attracted much interest in system identification community. Recently, some researchers describe these approaches in a common framework, i.e, the nonlinear black-box structures [92, 46]. Obviously, these nonlinear black-box structures are all flexible enough to identify most reasonable systems in practice.

It should however be pointed out that a great deal of attention for the nonlinear black-box models is so far paid only to the flexibility of the model structures. Those nonlinear models abandon the properties of highly successful linear black-box models.

# Appendix B

# A Hierarchical Network

In the literature, classical models for representing nonlinear systems usually have structures corresponding to the orders of system nonlinearity. One typical example is the Volterra Series (SISO) [10]

$$y(t) = \sum_{n=0}^{\infty} \int_0^{\infty} \overset{n}{\cdots} \int_0^{\infty} \mathcal{K}_n(\tau_1, ..., \tau_n) x(t-\tau_1) \ ... \ x(t-\tau_n) d\tau_1 \ .... \ d\tau_n \tag{B.1}$$

where $x(t)$ is the input, $y(t)$ is the output, and $\mathcal{K}_n(\tau_1, ..., \tau_n)$ is the *nth-order Volterra kennel*. Obviously, the term associated with the 1st-order Volterra kennel $\mathcal{K}_1(\tau_1)$ corresponding to linear part, while the $n$th-order Volterra kennel $\mathcal{K}_n(\tau_1, ..., \tau_n)$ to the $n$th-order nonlinearity. Another example is the general Kolmogorov-Gabor polynomial (MISO) [17, 85]

$$y(t) = y_0 + \sum_{i=1}^{r} \alpha_i x_i(t) + \sum_{i=1}^{r}\sum_{j=1}^{r} \alpha_{ij} x_i(t) x_j(t) + \sum_{i=1}^{r}\sum_{j=1}^{r}\sum_{l=1}^{r} \alpha_{ijl} x_i(t) x_j(t) x_l(t) + ... \tag{B.2}$$

where $x_i(t)$, $(i = 1, ..., r)$ are inputs and $y(t)$ is output. We can see that the terms associated with $\alpha_i$ corresponding to linear part, associated with $\alpha_{ij}$ corresponding to the 2nd-order nonlinearity and so on.

We will show that such representations of nonlinear systems can be developed into a hierarchical structure, i.e, *hierarchical network*, in which the parameters are grouped into layers corresponding to the orders of system nonlinearity. This hierarchical structure will provide us a theoretical foundation for developing the hybrid quasi-linear black-box model.

## B.1 ARX Network

Let us consider SISO nonlinear discrete-time systems which can be described by the general Kolmogorov-Gabor polynomial (B.2), in which the elements $[x_i(t) \ \ i = 1, .., r]$ are assumed to be the past input-outputs of system,

$$\left.\begin{array}{ll} x_i(t) = y(t-i) & (i = 1, ..., n) \\ x_{j+n}(t) = u(k-j) & (j = 1, ..., m) \\ \quad (r = n+m) & \end{array}\right\}. \tag{B.3}$$

Using (B.3) in (B.2), we can get

$$y(t) = y_0 + \sum_{i=1}^{n} \bar{a}_{i,t} y(t-i) + \sum_{i=1}^{m} \bar{b}_{i,t} u(t-i) \tag{B.4}$$

where

$$\bar{a}_{i,t} = a_i + \Delta a_{i,t} \qquad \bar{b}_{i,t} = b_i + \Delta b_{i,t} \tag{B.5}$$

$$\Delta a_{i,t} = \sum_{j=1}^{n} \alpha_{ij} y(t-j) + \frac{1}{2} \sum_{j=n+1}^{r} \alpha_{ij} u(t-j+n) + ... \tag{B.6}$$

$$\Delta b_{i,t} = \sum_{j=n+1}^{r} \alpha_{ij} u(t-j+n) + \frac{1}{2} \sum_{j=1}^{n} \alpha_{ij} y(t-j) + ... \tag{B.7}$$

In (B.5), $a_i$ and $b_i$ stand for $\alpha_i$ in (B.2).

(B.4) shows that a general nonlinear system may be expressed in a linear ARX structure, in which the system nonlinearity is distributed and embedded into the coefficients. On the other hand, the nonlinear terms of the coefficients, $\Delta a_{i,t}$ and $\Delta b_{i,t}$, are nonlinear functions of input-output variables. Hence, each of them again can be expressed in a a linear ARX structure, in which each coefficient consists of a constant parameter and a nonlinear term. Obviously, this procedure can be continued for the nonlinear terms. As the result, a general nonlinear system can be converted into a form which consists of a network structure of ARXs. We call it *ARX Network*.

To simplify the discussion, we introduce a regression vector $\varphi(t)$ and parameter vector $\Theta_i$

$$\varphi(t) = [y(t-1) \; ... \; y(t-n) \; u(t-1) \; ... \; u(t-m)]^T \tag{B.8}$$

$$\left.\begin{array}{l} \Theta_0 = y_0 \\ \Theta_1 = [a_1 \; ... a_n \; b_1 \; ... \; b_m \; c_1 \; ... \; c_l]^T \\ \Delta\Theta_{1,t} = [\Delta a_{1,t} \; ... \; \Delta a_{n,t} \; \Delta b_{1,t} \; ... \; \Delta b_{m,t}]^T \end{array}\right\}. \tag{B.9}$$

Then (B.4) can be written as

$$y(t) = \Theta_1^T \varphi(t) + \Delta\Theta_{1,t}^T \varphi(t) + \Theta_0 + v(t). \tag{B.10}$$

Representing $\Delta\Theta_{1,t}$ in a linear ARX structure, we have

$$\Delta\Theta_{1,t}^T \varphi(t) = \Theta_2^T \varphi(t) \otimes \varphi(t) + \Delta\Theta_{2,t}^T \varphi(t) \otimes \varphi(t) \tag{B.11}$$

where the symbol $\otimes$ denotes Kronecker production and

$$\begin{array}{l} \Theta_2 = [a_{1,1} \; ... \; a_{n,1} \; b_{1,1} \; ... \; b_{m,1} \\ \qquad ... \; a_{1,n+m} \; ... \; a_{n,n+m} \; b_{1,n+m} \; ... \; b_{m,n+m}]^T. \end{array} \tag{B.12}$$

Continuing this procedure, (B.10) can be represented as

$$y(t) = \sum_i \Theta_i^T \left( \underbrace{\varphi(t) \otimes ... \otimes \varphi(t)}_{i-1} \right) + \Theta_0 + v(t) \tag{B.13}$$

where $i = 1, 2, ....$ Figure B.1 shows (B.13) in a network.

As shown in Fig B.1, (B.13) represents a nonlinear system into a hierarchical form, in which $\Theta_0$ and $\Theta_1$ represent linear part, $\Theta_2$ the second order nonlinear part, $\Theta_3$ the third order nonlinear part and so on. It should however be noticed that the number of parameters in $\Theta_i$ increases exponentially as $i$ increases, i.e, $\dim(\Theta_i) = (n+m)^i$. Therefore in practice, as a model the ARX Network can only be applied to the systems without too high order nonlinearity. But, the significance of ARX Network modeling is that it provides a framework to express a nonlinear system in a linear structure, in which the system nonlinearity is distributed and embedded the layered coefficients. By representing the nonlinear terms of coefficients corresponding to higher order nonlinearity using the NNMs such as adaptive fuzzy systems, a hybrid modeling can be realized.

As an example, let us suppose to represent the higher order nonlinearity of system, say the order higher than 2, by using a set of adaptive fuzzy systems. Introduce a vector $\mathcal{F}$

$$\mathcal{F}_t = [f_1(t) \; f_2(t) \; ... \; f_r(t)] \tag{B.14}$$

where $f_v(t)$, $(v = 1, .., r)$ are realized by using adaptive fuzzy systems and let

$$\Delta\Theta_{1,t}^T \varphi(t) = \Theta_2^T \varphi(t) \otimes \varphi(t) + \mathcal{F}_t \varphi(t) \tag{B.15}$$

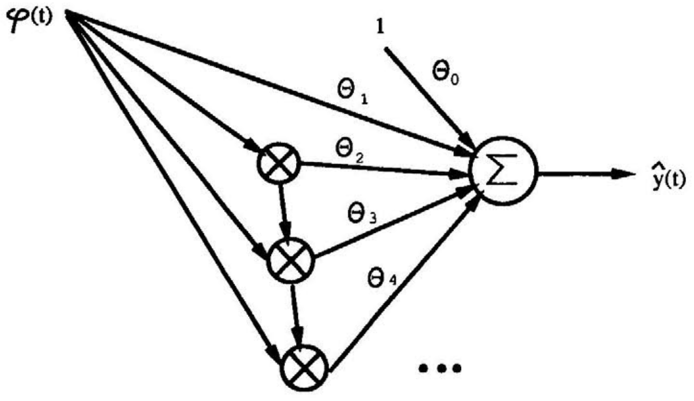


Figure B.1: ARX Network (All of transposition 'T' were omitted for simplicity)

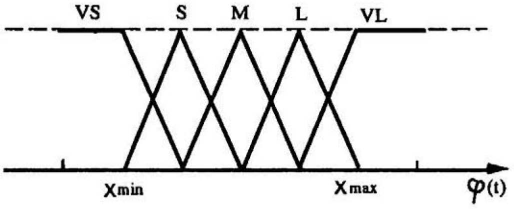


Figure B.2: The fuzzy sets in the input space

Then (B.10) can be written as

$$y(t) = \Theta_1^T \varphi(t) + \Theta_2^T \varphi(t) \otimes \varphi(t) + \mathcal{F}_t \varphi(t) + \Theta + v(t) \tag{B.16}$$

We call (B.16) as a *Hybrid ARX Network*, which can be used to identify systems with higher order nonlinearity.

The adaptive fuzzy systems used to realize $f_v(\varphi(t))$ are given by

$$f_v(\varphi(t)) = \frac{\sum_{j=1}^{M} \omega_{vj} \left(\wedge_{i=1}^{r} \mu_{A_i^j}(x_i(t))\right)}{\sum_{j=1}^{M} \left(\wedge_{i=1}^{r} \mu_{A_i^j}(x_i(t))\right)} \tag{B.17}$$

which is similar in form to (A.17). Note here that the fuzzy membership function is the input space, $\mu_{A_i^j}$, are determined *a priori*. For example, if the system operating region is mostly located in $X_{min} \leq \varphi(t) \leq X_{max}$, the possible fuzzy sets may be something like Fig. B.2.

## B.2 The Hierarchical Network and General Nonlinear Structure

Based on the framework stated in [92, 46], a general nonlinear ARX structure can be seen as a concatenation of a mapping from observed data to a regression vector $\varphi(t)$ and a nonlinear mapping $g(\cdot)$ from the regressor space to the output space

$$y(t) = g\left(\varphi(t)\right) + v(t). \tag{B.18}$$

Now let us perform Taylor expansion to $g\left(\varphi(t)\right)$ around the region $\varphi(t) = 0$

$$y(t) = g(0) + g'(0)\varphi(t) + \frac{1}{2}\varphi^T(t)g''(0)\varphi(t) + \ldots + v(t) \tag{B.19}$$

where the prime denotes differentiation with respect to $\varphi(t)$. If $g(\cdot)$ is assumed to be a continuously differentiable function, a model can be obtained from the expression (B.19). Introducing vectors $\Theta_i$ and appropriately arranging their elements

$$\Theta_0 \sim g(0), \quad \Theta_1 \sim g'(0), \quad \Theta_2 \sim g''(0), \ldots \tag{B.20}$$

(B.19) can be written as

$$\begin{aligned} y(t) &= \Theta_0 + \Theta_1^T\varphi(t) + \Theta_2^T\varphi(t) \otimes \varphi(t) + \ \ldots \ + v(t) \\ &= \sum_i \Theta_i^T \left( \varphi(t) \underbrace{\otimes \ldots \otimes}_{i-1} \varphi(t) \right) + \Theta_0 + v(t) \end{aligned} \tag{B.21}$$

which is the same as (B.13).

Note the elements of $\varphi(t)$ are discrete-time variables, but in the Taylor expansion we treated them as they were continuous variable. Therefor the treatment above is not strict. From which we however can know some insight about the hierarchical network.

# Appendix C

# Derivation of Recursive PEM Algorithm

Consider the problem of estimation $\Theta$ by minimizing a criterion based on prediction error

$$\hat{\Theta}(N) = \arg\min_{\Theta} \{V_N(\Theta)\} \tag{C.1}$$

$$V_N(\Theta) = \frac{1}{2}\sum_{t=1}^{N} \varepsilon^2(t,\Theta) \tag{C.2}$$

$$\varepsilon(t,\Theta) \triangleq y(t) - \hat{y}(t|\Theta) \tag{C.3}$$

We will here derive a recursive algorithm for this problem based on the derivation in [70].

Let $\hat{\Theta}(t-1)$ be our estimate at time $t-1$. We wish to obtain a $\hat{\Theta}(t)$ that (approximately) minimizes $V_t(\Theta)$. By mean of a Taylor expansion of $V_t(\Theta)$ around $\hat{\Theta}(t-1)$ we obtain

$$\begin{aligned} V_t(\Theta) &= V_t(\hat{\Theta}(t-1)) + V_t'(\hat{\Theta}(t-1))[\Theta - \hat{\Theta}(t-1)] \\ &\quad + \frac{1}{2}[\Theta - \hat{\Theta}(t-1)]^T V_t''(\hat{\Theta}(t-1))[\Theta - \hat{\Theta}(t-1)] \\ &\quad + o(|\Theta - \hat{\Theta}(t-1)|^2) \end{aligned} \tag{C.4}$$

where the prime denotes differentiation with respect to $\Theta$, and $o(x)$ denotes a function such that $o(x)/|x| \to 0$ as $|x| \to 0$. Minimization of this expression with respect to $\Theta$ gives

$$\hat{\Theta}(t) = \hat{\Theta}(t-1) - [V_t''(\hat{\Theta}(t-1))]^{-1} V_t'[\hat{\Theta}(t-1)]^T + o(|\Theta - \hat{\Theta}(t-1)|) \tag{C.5}$$

If we denote the *negative* derivative of $\varepsilon(t,\Theta)$ with respect to $\Theta$ by

$$\psi(t,\Theta) \triangleq \left[-\frac{\partial\varepsilon(t,\Theta)}{\partial\Theta}\right]^T \tag{C.6}$$

We have

$$[V_t'(\Theta)]^T = -\sum_{k=1}^{t} \psi(k,\Theta)\varepsilon(k,\Theta) = [V_{t-1}'(\Theta)]^T - \psi(t,\Theta)\varepsilon(t,\Theta), \tag{C.7}$$

and, by differentiating once more,

$$V_t''(\Theta) = V_{t-1}''(\Theta) + \psi(t,\Theta)\psi^T(t,\Theta) + \varepsilon''(t,\Theta)\varepsilon(t,\Theta), \tag{C.8}$$

where $\varepsilon''(t,\Theta)$ is the second-derivative matrix of $\varepsilon(t,\Theta)$ with respect to $\Theta$.

In order to evaluate (C.5) a number of approximations have to introduced.

- first we assume that the next estimate $\hat{\Theta}(t)$ is to be found in a small neighborhood of $\hat{\Theta}(t-1)$. This should be a reasonable approximation if $t$ is large. That assumption leads to the following two approximations:

Neglect $o(|\hat{\Theta}(t) - \hat{\Theta}(t-1)|)$ in (C.5) (C.9)

and take

$$V_t''(\hat{\Theta}(t)) = V_t''(\hat{\Theta}(t-1)) \tag{C.10}$$

- Then we assume that $\hat{\Theta}(t-1)$ is indeed the optimal estimate a time $t-1$, so that

$$V_{t-1}'(\hat{\Theta}(t-1)) = 0. \tag{C.11}$$

- Finally we set

$$\varepsilon_{t-1}''(\hat{\Theta}(t-1)) = 0. \tag{C.12}$$

The rationale for the approximation (C.12) is as follows. Close to the true value $\Theta_0$, $\{\varepsilon(t,\Theta)\}$ will be *almost* white noise, so that we may approximately consider $\varepsilon(t,\Theta)$ to be of zero mean and independent of what happened up to time $t-1$. In particular, it would then be independent of $\varepsilon''(t,\Theta)$. The expected value of the left-hand side of (C.12) then is indeed close to zero, so that the last term of (C.8) makes an order of magnitude less contribution to $V_t''$ than the second term.

With the assumption (C.12) and (C.10) inserted into (C.8) we can approximately evaluate the second-derivative matrix. Let this approximation be denoted by $\bar{R}(t)$. Then we have

$$\bar{R}(t) = \bar{R}(t-1) + \psi(t,\hat{\Theta}(t-1))\psi^T(t,\hat{\Theta}(t-1)) \tag{C.13}$$

Using this expression for $V_t''(\hat{\Theta}(t-1)$ and the approximation $\bar{R}(t)$ for $V_t''(\hat{\Theta}(t-1))$ in (C.5) together with the assumption (C.9), we have

$$\hat{\Theta}(t) = \hat{\Theta}(t-1) + \bar{R}^{-1}(t)\psi(t,\hat{\Theta}(t-1))\varepsilon(t,\hat{\Theta}(t-1)) \tag{C.14}$$

The algorithm (C.14) is not, however, well suited for computation as it stands, since a matrix has to be inverted in each time step. It is more nature to introduce

$$P(t) = \bar{R}^{-1}(t) \tag{C.15}$$

and update $P(t)$ directly, instead of using (C.13). This is accomplished by the so-called matrix inversion lemma as follows:

*LEMMA* Let $A$, $B$, $C$, and $D$ be matrices of compatible dimensions, so that the product $BCD$ and the sum $A+BCD$ exist. Then

$$[A+BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}DA^{-1} \tag{C.16}$$

{Proof} Multiply the right-hand side of (C.16) by $A+BCD$ for the right. This gives

$$\begin{aligned}
& I + A^{-1}BCD - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}D - A^{-1}B[DA^{-1}B + C^{-1}]DA^{-1}BCD \\
&= IA^{-1}B[DA^{-1}B + C^{-1}]^{-1}\{[DA^{-1}B + C^{-1}]CD - D - DA^{-1}BCD\} \\
&= I + A^{-1}B[DA^{-1}B + C^{-1}]^{-1}\{0\} = I
\end{aligned}$$

which proves (C.16).

Applying (C.16) and (C.15) to (C.13) with $A = \bar{R}(t-1)$, $B = \psi(t,\hat{\Theta}(t-1))$, $C = 1$, $D = \psi^T(t,\hat{\Theta}(t-1)$ gives

$$\begin{aligned}
P(t) &= \left[P^{-1}(t-1) + \psi(t,\hat{\Theta}(t-1))\psi^T(t,\hat{\Theta}(t-1))\right]^{-1} \\
&= P(t-1) - P(t-1)\psi(t,\hat{\Theta}(t-1))\left[\psi^T(t,\hat{\Theta}(t-1))P(t-1)\psi(t,\hat{\Theta}(t-1))\right]^{-1} \\
&\quad \psi^T(t,\hat{\Theta}(t-1))P(t-1) \\
&= P(t-1) - \frac{P(t-1)\psi(t,\hat{\Theta}(t-1))\psi^T(t,\hat{\Theta}(t-1))P(t-1)}{1+\psi^T(t,\hat{\Theta}(t-1))P(t-1)\psi(t,\hat{\Theta}(t-1))}.
\end{aligned} \tag{C.17}$$

From (C.17) we also find that

$$\begin{aligned} P(t)\psi(t,\hat{\Theta}(t)) &= P(t-1)\psi(t,\hat{\Theta}(t)) \\ &\quad -\frac{P(t-1)\psi(t,\hat{\Theta}(t-1))\psi^T(t,\hat{\Theta}(t-1))P(t-1)\psi(t,\hat{\Theta}(t-1))}{1+\psi^T(t,\hat{\Theta}(t-1))P(t-1)\psi(t,\hat{\Theta}(t-1))} \\ &= \frac{P(t-1)\psi(t,\hat{\Theta}(t))}{1+\psi^T(t,\hat{\Theta}(t-1))P(t-1)\psi(t,\hat{\Theta}(t-1))} \end{aligned} \tag{C.18}$$

The recursive PEM algorithm can thus be finally expressed as

$$\hat{\Theta}(t) = \hat{\Theta}(t-1) + L(t)\varepsilon(t,\hat{\Theta}(t-1)) \tag{C.19}$$

$$L(t) = \frac{P(t-1)\psi(t,\hat{\Theta}(t-1))}{1+\psi^T(t,\hat{\Theta}(t-1))P(t-1)\psi(t,\hat{\Theta}(t-1))} \tag{C.20}$$

$$P(t) = P(t-1) - \frac{P(t-1)\psi(t,\hat{\Theta}(t-1))\psi^T(t,\hat{\Theta}(t-1))P(t-1)}{1+\psi^T(t,\hat{\Theta}(t-1))P(t-1)\psi(t,\hat{\Theta}(t-1))} \tag{C.21}$$

# Appendix D

# Fault Detection Scheme Using the KDI

The Kullback discrimination information (KDI) is basically a measure to compare two probability density functions $p_1(x)$ and $p_2(x)$. It is given by

$$I[1,2] = \int p_1(x) \log \frac{p_1(x)}{p_2(x)} dx \geq 0 \tag{D.1}$$

where equality holds if and only if $p_1(x) = p_2(x)$ [52]. The value $I[1,2]$ hence can be used as a measure of how much $p_1(x)$ deviates from $p_2(x)$. Moreover, a symmetric form of such a measure can be introduced by considering $J[1,2] = I[1,2] + I[2,1]$.

Recently, Kumammaru and co-workers have developed a KDI-based fault detection scheme based on the use of KDI for model discrimination, see [93, 61, 60]. In this appendix, we first present a summary for the KDI-based fault detection scheme, then point out several restrictions of the scheme for practical applications.

## D.1 Preliminaries

Consider a discrete-time linear SISO system described by:

$$\begin{aligned} \mathcal{S}: \quad y(t) &= G_0(q^{-1})u(t) + H_0(q^{-1})e(t) \\ e(t) &\in N(0, \sigma_0^2) \end{aligned} \tag{D.2}$$

where $y(t)$ is the output at time $t$ ($t = 1, 2, ...$), $u(t)$ the input and $e(t)$ the white Gaussian noise. $G_0(q^{-1})$, $H_0(q^{-1})$ are scalar rational functions in the backward shift operator $q^{-1}$. Without loss of generality, we will assume that $G_0(q^{-1})$ and $H_0(q^{-1})$ meet the following conditions:

- $G_0(q^{-1})$, $H_0(q^{-1})$ are causal
- $G_0(0) = 0$, $H_0(0) = 1$
- $G_0(q^{-1})$, $H_0(q^{-1})$ are asymptotically stable.

When identifying the system, we assume that the parametric models of the form

$$\begin{aligned} \mathcal{M}(\theta): \quad y(t) &= G(q^{-1}, \theta)u(t) + H(q^{-1}, \theta)e(t) \\ e(t) &\in N(0, \sigma^2) \end{aligned} \tag{D.3}$$

are used, where $\theta$ and $\sigma^2$ denote unknown parameters. We further assume that for any value of $\theta$ the filters $G(q^{-1}, \theta)$ and $H(q^{-1}, \theta)$ fulfill the assumptions stated above for $G_0(q^{-1})$ and $H_0(q^{-1})$.

Assume then that data from the system are available from two distinct time intervals $I_1$ and $I_2$, see Fig. D.1, in which $N_i$ ($i = 1, 2$) denotes the number of data points in the interval $I_i$, while $X_i$ denotes the experimental condition (e.g. the input characteristics like the spectral density $\Phi_u(\omega)$).

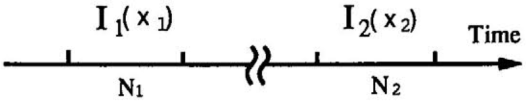


Figure D.1: Two distinct time intervals

Now perform the identification and denote the estimates of the unknown parameter vector $\theta$ using the data obtained from these two intervals by $\hat{\theta}_1$ and $\hat{\theta}_2$, respectively.

We now consider the problem to decide whether or not

$$\hat{\theta}_1 = \hat{\theta}_2 \tag{D.4}$$

reflecting the uncertainties in the data. This problem may be given two different interpretations:

(1) **Fault Detection.** In a system described by an input-output model, a system fault caused by the changes in system configuration parameters will results in changes of model parameters. Based on this assumption, if two identified models are significantly different, a decision can be made that a fault has occurred.

(2) **Model Validation.** If the model parameterization is chosen adequately for the system then we expect the obtained models (and hence also the two parameter vectors $\hat{\theta}_1$ and $\hat{\theta}_2$) to be reasonably equal. If $\mathcal{M}$ is not sufficiently rich to describe the system then the models might differ more.

We here only consider fault detection problem. Let us assume that the model parameterization does be chosen adequately for the system, that is, there is no unmodeled dynamics in the modeling. For such an ideal case, the distortion of $\mathcal{M}(\hat{\theta}_1)$ and $\mathcal{M}(\hat{\theta}_2)$ implies a change of system parameters which may be caused by a system failure. The discrimination of the identified models $\mathcal{M}(\hat{\theta}_1)$ and $\mathcal{M}(\hat{\theta}_2)$ can be executed via the KDI.

Applying the KDI to the likelihood functions of the identified models, $p(Y_{N_1}|\hat{\theta}_i, U_{N_1-1})$ $(i = 1,\, 2)$, we thus have:

$$I_{N_1}[1,2] = \int p(Y_{N_1}|\hat{\theta}_1, U_{N_1-1}) \log \frac{p(Y_{N_1}|\hat{\theta}_1, U_{N_1-1})}{p(Y_{N_1}|\hat{\theta}_2, U_{N_1-1}))} dY_{N_1} \tag{D.5}$$

where $Y_{N_1}$ and $U_{N_1}$ denote the data collections up to $N_1$ taken from the first interval $I_1$ defined by:

$$Y_{N_1} = [y(1)\ y(2)\ ...\ y(N_1)]^T, \quad U_{N_1} = [u(1)\ u(2)\ ...\ u(N_1)]^T. \tag{D.6}$$

The index in (D.5) hence indicates how well the model using $\hat{\theta}_2$ describes the data in the interval $I_1$. In other words the criteria $I_{N_1}[1,2]$ is an index for discriminating the models $\mathcal{M}(\hat{\theta}_1)$ and $\mathcal{M}(\hat{\theta}_2)$ via the difference in the corresponding likelihood functions, see Fig. D.2.

## D.2 Evaluation of the KDI

Obviously, evaluating (D.5) directly based on the Gaussian distribution of the likelihood functions requires a great computational labour due to large matrix operations for a finite but fairly large data sets. In order to solve this problem, an algorithm has been derived so that the KID can be evaluated in a feasible way. The derivation steps are as follows, see Söderström and Kumamaru (1985) [93] for details:

- *Apply Bayes' rule*

$$p(Y_{k+1}|\hat{\theta}_i, U_k) = p(y(k+1)|\hat{\theta}_i, Y_k, U_k)p(Y_k|\hat{\theta}_i, U_k) \tag{D.7}$$

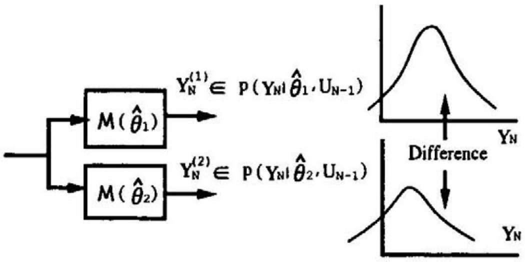

$M(\hat{\theta}_1)$
$M(\hat{\theta}_2)$
$Y_N^{(1)} \in p(Y_N|\hat{\theta}_1, U_{N-1})$
$Y_N^{(2)} \in p(Y_N|\hat{\theta}_2, U_{N-1})$
Difference
$Y_N$
$Y_N$

Figure D.2: Model discrimination using the KDI

- *Due to the assumption on Gaussian distributed disturbance we have*

$$p(y(k+1)|\hat{\theta}_i, Y_k, U_k) = N(m_{ki}, \sigma_i^2) \tag{D.8}$$

  Here the conditional mean $m_{ki}$ is equal to one-step ahead prediction of $y(k+1)$.
- *Use of straightforward manipulation with the model equations, which are assumed to describe the data.*

The result of such a derivation is:

$$I_{N_1}[1,2] = I_0[1,2] + \sum_{j=1}^{3} I_{N_1}^{(j)}[1,2] \tag{D.9}$$

where $I_0[1,2]$ is an initial value which might be neglected and other components are given as:

$$I_{N_1}^{(1)}[1,2] = \frac{N_1}{2}\left[(\hat{\sigma}_1^2/\hat{\sigma}_2^2 - 1) - \log(\hat{\sigma}_1^2/\hat{\sigma}_2^2)\right] \tag{D.10}$$

$$I_{N_1}^{(2)}[1,2] = \frac{1}{2}\sum_{k=0}^{N_1-1} \left\|H_2^{-1}(G_1 - G_2)u(k+1)\right\|_{\hat{\sigma}_2^{-2}}^2 \tag{D.11}$$

$$I_{N_1}^{(3)}[1,2] = \frac{N_1}{2}\left\{\frac{\hat{\sigma}_2^{-2}}{2\pi i}\oint \left(H_2^{-1}(z)H_1(z) - 1\right)\hat{\sigma}_1^2\left(H_1(z^{-1})H_2^{-1}(z^{-1}) - 1\right)\frac{dz}{z}\right\} \tag{D.12}$$

$$G_i = G(q^{-1}, \hat{\Theta}_i), \qquad H_i(z) = H(z, \hat{\Theta}_i) \qquad i = 1,2 \tag{D.13}$$

These different terms have the following meanings: The first term (D.10) expresses the deviation of $\hat{\sigma}_2^2$ from $\hat{\sigma}_1^2$. The second term (D.11) shows the difference between $G_1$ and $G_2$. Note that the difference in the deterministic output $Gu(t)$ is filtered with $H_2^{-1}$. In this way we relate the difference to the prediction error instead of the output. The last term (D.12) describes the difference between the noise filters $H_1$ and $H_2$. All of these terms become zero when $\hat{\theta}_1 = \hat{\theta}_2$ and otherwise positive from the property of the KDI construction. In particular the case $G_1 = G_2$ is the only case when the second term becomes zero if the input is persistently exciting the system. Therefore, the following criteria for fault detection can be introduced based on the KDI:

$$W^{\sigma^2} = N_1\left[(\hat{\sigma}_1^2/\hat{\sigma}_2^2 - 1) - \log(\hat{\sigma}_1^2/\hat{\sigma}_2^2)\right] \tag{D.14}$$

$$W^{o} = \sum_{k=0}^{N_1-1} \left(H_2^{-1}(G_1 - G_2)u(k+1)\right)^2/\hat{\sigma}_2^{-2} \tag{D.15}$$

$$W^{e} = \frac{\hat{\sigma}_2^{-2}N_1}{2\pi i}\oint \left(H_2^{-1}(z)H_1(z) - 1\right)\hat{\sigma}_1^2\left(H_1(z^{-1})H_2^{-1}(z^{-1}) - 1\right)\frac{dz}{z} \tag{D.16}$$

All criteria $W$ can be used as indexes fault detection by the thresholding approach

$$W \gtrless \eta \Longrightarrow \begin{cases} \text{fault} \\ \text{no fault} \end{cases} \tag{D.17}$$

where the threshold $\eta$ should be appropriately determined under a specification of decision accuracy, e.g. a significance level for a false alarm rate, in which the statistical properties of $I_{N_1}$ provide useful information.

## D.3 Asymptotic Properties of the Criteria

Assume that a prediction error method (PEM) [70, 94] is used for parameter estimation. Then we take estimated parameter vector as:

$$\hat{\theta}_i = \arg\min_{\theta_i} \left\{ \frac{1}{N_i} \sum_{t=1}^{N_i} \varepsilon(t, \theta_i)^2 \right\} \tag{D.18}$$

where

$$\begin{aligned} \varepsilon(t, \theta_i) &= y(t) - \hat{y}(t|t-1, \hat{\theta}_i) \\ &= H_i^{-1} y(t) - H_i^{-1} G_i u(t) \end{aligned} \tag{D.19}$$

The asymptotic parameter vector $\theta_i^*$ are given by

$$\theta_i^* = \arg\min_{\theta_i} \left\{ E\varepsilon(t, \theta_i)^2 \right\} \tag{D.20}$$

$\theta_i^*$ does not depend on $i$ when the system is included in the model structure. It is occasionally denoted by $\theta^*$ in such a case and called as the *true parameter vector*. The prediction error $\varepsilon(t, \theta^*)$ becomes identical to the white noise sequence $e(t)$ of the true system, see (D.2). The estimates have the following asymptotic Gaussian distributions [94]:

$$\sqrt{N_i}(\hat{\theta}_i - \theta^*) \xrightarrow{\text{dist}} N(0, P_i) \tag{D.21}$$

$$P_i = \left[ E\psi^T(t) \sigma_0^{-2} \psi(t) \right]^{-1} \tag{D.22}$$

where $\psi(t)$ denotes the gradient of the prediction error with respect to the parameter vector:

$$\psi(t) = \left. \frac{\partial \varepsilon(t, \theta)}{\partial \theta} \right|_{\theta = \theta^*} \tag{D.23}$$

Note that the expectation in (D.22) is to be carried out for the experimental condition $X_i$. Thus the covariance matrix $P_i$ will depend on $X_i$.

In the ideal case where the system is stationary and the model structure is sufficiently rich to describe the system, the estimated parameter vector $\hat{\theta}_1$ and $\hat{\theta}_2$ are close to each other, but not identical since the time intervals $I_1$ and $I_2$ are finite and there will always be some random fluctuations. It can be shown that all criteria of KDI are approximately expressed by:

$$W/N_i = (\hat{\theta}_1 - \hat{\theta}_2)^T Q((\hat{\theta}_1 - \hat{\theta}_2) \tag{D.24}$$

where the weighting matrix $Q$ depends on the criteria and the model structure. By using the Taylor expansion, the $Q$ matrices are given as follows [93]:

$$Q(W^{\sigma^2})^{i,j} = \left. \frac{1}{2} \sigma_0^2 \frac{\partial \sigma^2}{\partial \theta_i} \sigma_0^2 \frac{\partial \sigma^2}{\partial \theta_j} \right|_{\theta = \theta^*} \tag{D.25}$$

$$Q(W^o)^{i,j} = E \left[ H_0^{-1}(q_1) \frac{\partial G(q^{-1}, \theta)}{\partial \theta_i} u(t) \right] \sigma_0^{-2} \left[ H_0^{-1}(q_1) \frac{\partial G(q^{-1}, \theta)}{\partial \theta_j} u(t) \right]_{\theta = \theta^*} \tag{D.26}$$

$$Q(W^e)^{i,j} = \frac{\sigma_0^{-2}}{2\pi i}\oint \left\{ H_0^{-1}(z)\frac{\partial H(z,\theta)}{\partial \theta_i}\sigma_0^2\frac{\partial H(z,\theta)}{\partial \theta_j}H_0^{-1}(z^{-1})\right\}_{\theta=\theta^*}\frac{dz}{z} \tag{D.27}$$

Since $\hat{\theta}_1$ and $\hat{\theta}_2$ are independent, from (D.21) we have

$$\sqrt{N_1}(\hat{\theta}_1 - \hat{\theta}_2) = \sqrt{N_1}(\hat{\theta}_1 - \theta^*) - \sqrt{\frac{N_1}{N_2}}\cdot\sqrt{N_1}(\hat{\theta}_2 - \theta^*) \xrightarrow{\text{dist}} N(0,\bar{P}) \tag{D.28}$$

with

$$\bar{P} = P_1 + (N_1/N_2)P_2 \tag{D.29}$$

If in the special case the experimental conditions are equal ($X_1 = X_2$), we have $P_1 = P_2$ and hence

$$\bar{P} = P_1(1 + N_1/N_2) \tag{D.30}$$

By inspection of (D.24) and (D.28), it can be shown that the asymptotic distribution of the criteria will be a $\chi^2$-distributed if the matrices $Q$ and $\bar{P}$ satisfy $Q = \bar{P}^{-1}$. The results are summarized as follows, see [93, 60] for details:

Assume that the model structure includes the true system and that the system is stationary in both intervals $I_1$ and $I_2$ with the same experimental condition ($X_1 = X_2$). If the model parameterization is such that (D.3) can be written as:

$$\begin{aligned} \mathcal{M}: \quad & y(t) = G(q^{-1},\theta_u)u(t) + H(q^{-1},\theta_e)e(t) \\ & e(t) \in N(0,\sigma^2) \end{aligned} \tag{D.31}$$

where $\theta_u$ and $\theta_e$ with dimensions $d_u = dim(\theta_u)$, $d_e = dim(\theta_e)$ do not have any common parameters, then as $N_1$ and $N_2$ tend to infinity we have

$$W^{\sigma^2} \xrightarrow{\text{dist}} \chi^2(1) \tag{D.32}$$

$$W^o/(1 + N_1/N_2) \xrightarrow{\text{dist}} \chi^2(d_u) \tag{D.33}$$

$$W^e/(1 + N_1/N_2) \xrightarrow{\text{dist}} \chi^2(d_e). \tag{D.34}$$

These asymptotic results can conveniently be used for a reasonable selection of the threshold value in the fault detection application.

## D.4 Fault Detection via KDI: Restrictions

As shown earlier, the criteria $W$ derived from the KDI, are indexes corresponding to weighted norm of $(\hat{\theta}_1 - \hat{\theta}_2)$, see (D.24), in which the weighting matrix $Q$ reflects the model structure and statistical properties of the disturbance. They therefore are appropriate indexes for fault detection, see [62, 59, 53, 57, 58]. However, as far as the fault detection is carried out by a simple thresholding approach like (D.17), there are several restrictions for the practical applications.

The KDI-based fault detection is based on two important assumptions: (1) The system to be diagnosed is linear; (2) the model parameterization is chosen adequately for the system, which are never true in real applications. In the KDI-based fault detection scheme, the identified model is discriminated via the difference in the corresponding likelihood functions by using the KDI. The algorithm for evaluating the KDI in a feasible way was derived based on the assumption of Gaussian distribution for the likelihood functions. For a general nonlinear model, the likelihood function may no longer be assumed to be Gaussian distributed and results in the KDI-based fault detection scheme infeasible. Therefore, the assumption (1) is necessary. On the other hand, When the model structure does not include the system ($\mathcal{S} \notin \mathcal{M}$), that is, there is unmodeled dynamics in the modeling, the difference between the models identified from two intervals will become large so that the fault detection based on (D.17) might become infeasible. That follows the assumption (2).

# Appendix E

# A Review of Robust Identification Methods

The research dealing with the problem of quantifying the errors in estimated models for dynamic systems is a large field. Many approaches have been proposed, see e.g. [78, 79, 80]. Some of the typical approaches are related to the well-known titles of 'estimation in $H_\infty$' [32, 29, 28, 42, 120], 'worst-case estimation' [105, 48, 115], 'estimation in $l_1$' [71, 49] and 'stochastic embedding of undermodeling' [25]. Broadly classifying these approaches according to the forms for describing model uncertainty, we may however group them into two kinds: a *soft bound* approach which characterizes modeling error as a random quantity with *soft bound*, and a *hard bound* approach which describes modeling error as a deterministic quantity with *hard bound*.

## E.1 A Soft Bound Approach

The *soft bound* error description is so called 'stochastic embedding approach' proposed by Goodwin and his coworkers [25]. The basic idea can be summarized as follows.

Consider the problem of estimating a model for a dynamic system on the basis of the observation of an $N$ point input-output data sequence $Z^N = [\{u(t)\}, \{y(t)\}]$ where the observed data $Z^\infty$ is assumed to be generated by the system $\mathcal{S}$ according to

$$\mathcal{S}: \quad y(t) = G_0(q^{-1})u(t) + H(q^{-1})e(t) \tag{E.1}$$

Here $G_0(q^{-1})$ and $H(q^{-1})$ are rational transfer functions in the backward shift operator $q^{-1}$.

To obtain an estimate for the on-average characteristics of the total error, assume that the true transfer function $G_0(e^{-j\omega})$ is a stochastic process indexed by variable $\omega$. And further assume that, for the given choice of model set $\mathcal{M}_p^*$, and for some value $\theta_0$, it can be decomposed as

$$\begin{array}{rcl} G_0(e^{-j\omega}) & = & G(e^{-j\omega}, \theta_0) + G_\triangle(e^{-j\omega}) \\ \text{with} & & E\left\{G_0(e^{-j\omega})\right\} = G(e^{-j\omega}, \theta_0). \end{array} \tag{E.2}$$

It follows that $G_\triangle(e^{-j\omega})$ is a zero mean stochastic process

$$E\left\{G_\triangle(e^{-j\omega})\right\} = 0 \tag{E.3}$$

where $E\{\cdot\}$ means averaging over different realizations of the undermodeling.

In the stochastic embedding paradigm, $\left\{\nu_t = H(q^{-1}e(t)\right\}$ and $G_\triangle$ are assumed to be independent. And moreover, both $\nu_t$ and $G_\triangle$ are assumed to be associated with the probability density functions (pdf) $f_\nu(\nu_t, \gamma)$ and $f_\triangle(G_\triangle, \beta)$ respectively, where $\gamma$ and $\beta$ are real parameter vector to be estimated from the data. Examples for the pdfs $f_\nu(\nu_t, \gamma)$ and $f_\triangle(G_\triangle, \beta)$ are:

$$\nu_t \sim N(0, \sigma_\nu^2) \tag{E.4}$$

and

$$G_{\Delta}(q^{-1}) = \sum_{k=1}^{L} \eta_k q^{-k} \tag{E.5}$$

with

$$\begin{array}{l} \eta = [\eta_1 \; \eta_2 \; ... \; \eta_k] \sim N(0, C_\eta) \\ E\left\{\eta_k^2\right\} = \alpha\lambda^k \end{array} \tag{E.6}$$

## E.2 A Hard Bound Approach

In a *hard bound* approach, the modeling error is described as

$$\left|G_0(e^{-j\omega}) - G(e^{-j\omega}, \hat{\theta})\right| \leq r(e^{-j\omega}). \tag{E.7}$$

Several methods have been proposed to identify the error bound $r(e^{-j\omega})$, see e.g. [48, 105]. Here we only summarize the method discussed in [105].

The true system is assumed to be given by an IIR model

$$y(t) = \sum_{k=1}^{\infty} g_k^0 u(t-k) + v(t) \tag{E.8}$$

where the transfer function

$$G_0(q^{-1}) = \sum_{k=1}^{\infty} g_k^0 q^{-k} \tag{E.9}$$

is assumed to be stable ($q^{-1}$ is delay operator). Moreover, the noise $v(t)$ is assumed to be bounded.

Similar to the stochastic embedding approach, the following assumption concerning the true system is made:

The stable transfer function $G_0(q^{-1})$ can be divided into a simple and dominating low-order parametric parametric part $G(q^{-1}, \theta_0)$, and a more complex part $\Delta G(q^{-1})$. Thus

$$G_0(q^{-1}) = G(q^{-1}, \theta_0) + \Delta G(q^{-1}) \tag{E.10}$$

### E.2.1 The Nominal Model

The nominal model is assumed to be linear in the parameter vector $\theta$

$$G(q^{-1}, \theta) = \sum_{k=1}^{m} b_k B_k(q^{-1}), \qquad \theta = [b_1 \; ... \; b_m]^T \tag{E.11}$$

where $\left\{B_k(q^{-1})\right\}$ is given set of linear filters. Many models can be put into this framework. For example, the choice $B_k(q^{-1}) = q^{-k}$ gives a well-known FIR model. By instead taking

$$B_k(q^{-1}) = \frac{\sqrt{1-a^2}}{q-a}\left[\frac{1-aq}{q-a}\right]^{k-1}, \qquad -1 < a < 1 \tag{E.12}$$

we obtain the Laguerre model set [103, 104]. Here the Laguerre parameter $a$ should be taken close to the dominating pole of the system to be modeled.

### E.2.2 Modeling Error Part

The modeling error term $\Delta G(q^{-1})$ will be modeled as

$$\Delta G(q^{-1}) = D(q^{-1})\Delta(q^{-1}) \tag{E.13}$$

where $D(q^{-1})$ is a fixed and known "shaping" filter and

$$\Delta(q^{-1}) = \sum_{k=1}^{\infty} \delta_k q^{-k} \tag{E.14}$$

is a general stable transfer function.

⟨ *Summary* ⟩: A general stable linear system will be modeled as

$$\begin{aligned} G(q^{-1}) &= G(q^{-1}, \theta) + D(q^{-1})\Delta(q^{-1}) \\ &\Delta(q^{-1}) = \sum_{k=1}^{\infty} \delta_k q^{-k} \end{aligned} \tag{E.15}$$

where $G(q^{-1}, \theta)$ is a given parametric model and $D(q^{-1})$ is a fixed filter. The unknown quantities are the $m \times 1$ parameter vector $\theta$, together with the infinite sequence $\{\delta_k\}$.

### E.2.3 Estimation

The input-output relation corresponding to (E.15) can be written as

$$\begin{aligned} y(t) &= \varphi(t)^T \theta + \nu_t \\ &\nu_t = v(t) + \omega(t) + z(t) \end{aligned} \tag{E.16}$$

where

$$\begin{aligned} &\Theta = [\theta^T \ \delta_1 \ \ldots \ \delta_n]^T \\ &\varphi(t) = [B_1(q^{-1})u(t) \ \ldots \ B_m(q^{-1})u(t) \ D(q^{-1})u(t-1) \ \ldots \ D(q^{-1})u(t-n)]^T \end{aligned} \tag{E.17}$$

and the unknown input $\nu_t$ consists of three terms: $v(t)$ represents the noise contribution, $\omega(t)$ contains the transient effects from unknown initial conditions of the system, and $z(t)$ denotes the truncation error of $\Delta(q^{-1})$. Then this estimation can be realized using set membership identification method [6, 72] based on *a priori* information about the bound of the unknown input $\nu_t$, see [105] for details.

# Bibliography

[1] K.J. Åström. On self-tuning regulators. *Automatica*, 9:185–199, 1973.

[2] E.W. Bai. Adaptive identification of model uncertainties by rational. *IEEE Trans. on Automatic Control*, 36(4):441–453, 1991.

[3] M. Basseville. Detecting changes in signals and systems–a survey. *Automatica*, 24(3):309–326, 1988.

[4] M. Basseville, A. Benveniste, G. Mathis, and Q. Zhang. Detection and diagnosis of small changes in nonlinear systems – application to the monitoring of a gas turbine. In *Proc. of the TOOLDIAG'93 international Conference on Fault Diagnosis (Toulouse, France)*, volume 2, pages 487–495, 1993.

[5] D. Beasley, D.R. Bull, and R.R. Martin. An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, 15(2):58–69, 1993.

[6] G. Belforte, B. Bona, and V. Cerone. Parameter estimation algorithms for a set-membership description of uncertainty. *Automatica*, 26(5):887–898, 1990.

[7] A. Benveniste, A. Juditsky, B. Delyon, Q. Zhang, and P. Y. Glorennec. Wavelets in identification. In *Proc. of the 10th IFAC Symposium on Identification (Copenhagen)*, volume 2, pages 27–48, 1994.

[8] S. Billings. Identification of nonlinear systems - a survey. *IEE Proc.*, 127(6):272–, 1980.

[9] L. Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Trans. on Information Theory*, 39(3):999–1013, 1993.

[10] H.W. Chen. Modeling and identification of parallel nonlinear systems: Structural classification and parameter estimation methods. *Proceedings of the IEEE*, 83(1):39–66, 1995.

[11] S. Chen and S.A. Billings. Neural networks for nonlinear dynamic system modeling and identification. *INT. J. Control*, 56(2):319–346, 1992.

[12] S. Chen, C. Cowan, and Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. on Neural Networks*, 2(2):302–309, 1991.

[13] E.Y. Chow and A.S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Trans Automatic Control*, 29:603–614, 1984.

[14] L. Davis. *Genetic Algorithms and Simulated Annealing.* London, Pitman Publishing, 1987.

[15] G. Delmaire, J.P. Cassar, and M. Staroswiecki. Comparison of identification and parity space approach for failure detection in single-input single-output systems. In *Proc. of the IEEE Conference on Control Applications (Glasgow)*, volume 2, pages 865–870, 1994.

[16] B. Delyon, A. Juditsky, and A. Benveniste. Accuracy analysis for wavelet approximators. *IEEE Trans. on Neural Networks*, 6(2):332–348, 1995.

[17] P. Eykoff. *System Identification.* New York, Wiley, 1974.

[18] G. Feng. Robust direct adaptive control with least prior knowledge. *IEEE Trans on Circuits and Systems-I: Fundamental Theory and Applications*, 42(1):30–34, 1995.

[19] P.M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy–a survey and some new results. *Automatica*, 26(3):459–474, 1990.

[20] P.M. Frank. Analytical and qualitative model-based fault diagnosis — a survey and some new results. *European Journal of Control*, (2):6–28, 1996.

[21] P.M. Frank and X. Ding. Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis. *Automatica*, 30(5):789–804, 1994.

[22] P.M. Frank and J. Wunnenberg. Robust fault diagnosis using unknown input observer schemes. In R.J. Patton, P.M. Frank, and R. Clark, editors, *Fault Diagnosis in Dynamic Systems: Theory and Application*, pages 47–98. Prentice Hall, 1989.

[23] C.S. Fu and M. Poch. Application of a multi-layered neural network to system identification. *INT. J. SYSTEMS SCI.*, 24(8):1601–1609, 1993.

[24] D.E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning.* Addison-Wesley Publishing Company, Inc., 1989.

[25] G.C. Goodwin, M. Gevers, and B. Ninness. Quantifying the error in estimated transfer functions with application to model order selection. *IEEE Trans on Automatic Control*, 37(7):913–928, 1992.

[26] G.C. Goodwin and K.S. Sin. *Adaptive Filtering Prediction and Control.* Prentice-Hall, Inc., 1984.

[27] D. Gorinevsky. On the persistency of excitation in radial basis function network identification of nonlinear systems. *IEEE Trans. on Neural Networks*, 6(5):1237–1244, 1995.

[28] G. Gu and P.P. Dhargonekar. A class of algorithms for identification in $H_\infty$. *Automatica*, 28:299–312, 1992.

[29] G. Gu and P.P. Dhargonekar. Linear and nonlinear algorithms for identification in $H_\infty$ with error bounds. *IEEE Trans. on Automatic Control*, 37(7):953–963, 1992.

[30] E.J. Hartman, J.D. Keeler, and J.M. Kowalski. Layered neural networks with gaussian hidden units an universal approximations. *Neural Computation*, 2:210–215, 1990.

[31] M. Hasheminjad, J. Murata, k. Hirasawa, and S. Sagara. System identification using neural networks with parametric sigmoid functions. *Trans. of the Society of Instrument and Control Engineers*, 31(3):277–283, 1995.

[32] A. Helmicki, C.A. Jacobson, and C.N. Nett. Control oriented system identification: A worst case/deterministic approach in $H_\infty$. *IEEE Trans. on Automatic Control*, 36:1163–1176, 1991.

[33] F. Herrera, M. Lozano, and J.L. Verdegay. Genetic algorithms applications to fuzzy logic based systems. Technical Report DECSAI-93116, Facultad de Ciencias, Universdad de Granada, 1993.

[34] Y. Hirashima, Y. Iiguni, and N. Adachi. An adaptive control method using neural network (in Japanese). *Trans. of the Institute of Systems, Control and Information Engineers*, 9(1):281–286, 1994.

[35] K. Hornik. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[36] J. Hu and K. Kumamaru. Identification of nonlinear systems based on adaptive fuzzy systems embedding quasi-ARMAX model. In *Proc. of the 34th SICE Annual Conference (international session), Sapporo*, pages 1211–1216, 1995.

[37] J. Hu, K. Kumamaru, and K. Inoue. A hybrid quasi-ARMAX modeling scheme for identification and control of nonlinear systems. In *Proc. of 35th IEEE Conference on Decision and Control (Kobe)*, pages 1413–1418, 1996.

[38] J. Hu, K. Kumamaru, and K. Inoue. A hybrid robust identification using genetic algorithm and gradient method. *Trans. of the Society of Instrument and Control Engineers*, 32(5):714–721, 1996.

[39] L.G. Kraft III and D.P. Campagna. A summary comparison of CMAC neural network and traditional adaptive control systems. In W.R. Miller III, R.S. Sutton, and P.J. Werbos, editors, *Neural Networks for Control*, pages 143–169. The MIT Press, Massachusetts, 1990.

[40] W.T. Miller III, R.S. Sutton, and P.J. Werbos, editors. *Neural Networks for Control*. The MIT Press, Massachussetts, 1990.

[41] R. Isermann. Fault diagnosis of machines via parameter estimation and knowledge processing – tutorial paper. *Automatica*, 29(4):815–835, 1993.

[42] C. Jacobson and G. Tadmor. A note in $H_\infty$ system identification with probabilistic *a priori* information. In *Proc. American Control Conf., San Francisco*, pages 1539–1543, 1993.

[43] J.R. Jang. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst., Man & Cybern.*, 23(3):665–685, 1993.

[44] C.Z Jin, K. Wada, K. Hirsawa, J. Murata, and S. Sagara. Identification of nonlinear continuous systems by using neural network compensator (in Japanese). *Trans. of IEE of Japan*, 114-C(5):595–602, 1994.

[45] T.A. Johansen and B.A. Foss. Identification on non-linear system structure and parameters using regime decomposition. In *Proc. of the 10th IFAC Symposium on Identification (Copenhagen)*, volume 3, pages 131–136, 1994.

[46] A. Juditsky, H. Hjalmarssom, B. Delyon A. Benveniste, L. Ljung, J. Sjöberg, and Q. Zhang. Nonlinear black-box modeling in system identification: Mathematical foundations. *Automatica*, 31(12):1725–1750, 1995.

[47] H. Kitano. *Genetic Algorithm (in Japanese)*. Sanyou Tosyo, Japan, 1993.

[48] R.L. Kosut, M.K. Lau, and S.P. Boyd. Set-membership identification of systems with parametric and nonparametric uncertainty. *IEEE Trans. on Automatic Control*, 37(7):929–941, 1992.

[49] J.M. Krause and P.P. Khargonekar. Parameter identification in the presence of non-parametric dynamic uncertainty. *Automatica*, 26(1):113–123, 1990.

[50] G. Kreisselmeier and B.D.O. Anderson. Robust model reference adaptive control. *IEEE Trans. on Automatic Control*, 31(2):127–133, 1986.

[51] K. Kristinsson and G.A. Dumont. System identification and control using genetic algorithms. *IEEE Trans. Syst., Man & Cybern.*, 22(5):1033–1046, 1992.

[52] S. Kullback. *Information Theory and Statistics*. John Wiley & Sons Ltd., 1959.

[53] K. Kumamaru. a method of fault diagnosis in adaptive control systems by using fuzzy inference (in Japanese). *Trans. of IEE of Japan*, 113-C(3):113–122, 1991.

[54] K. Kumamaru, J. Hu, K. Inoue, and H. Ono. Fault detection via KDI in presence of unmodelled uncertainty. In *Proc. of the 26th ISCIE International Symposium on Stochastic Systems Theory and Its Applications (Osaka, Japan)*, pages 173–178, 1994.

[55] K. Kumamaru, J. Hu, K. Inoue, and T. Söderström. Robust fault detection using index of Kullback discrimination information. In *Proc. of the 13th IFAC World Congress (San Francisco)*, volume N, pages 205–210, 1996.

[56] K. Kumamaru, J. Hu, K. Inoue, and T. Söderström. A method of robust fault detection for dynamic systems by using quasi-ARMAX modeling. 1997. Submitted to the 11th IFAC Symp. on Identification (Kitakyushu).

[57] K. Kumamaru, K. Inoue, S. Nonaka, and T. Nakashima. Knowledge-aided model-based approach to failure diagnosis of adaptively controlled systems. In *Proc. of the IMACS/SICE Int. Symp. on Robotics, Mechatronics and Manufacturing Systems (Kobe, Japan)*, pages 841–846, 1992.

[58] K. Kumamaru, K. Inoue, S. Nonaka, H. Ono, and T. Söderström. A neural network approach to failure decision of adaptively controlled systems. In *Proc. of the 10th IFAC Symp. on Identification (Copenhagen)*, volume 2, pages 303–308, 1994.

[59] K. Kumamaru, S. Sagara, A. Nakai, and T. Söderström. A fuzzy inference approach to failure diagnosis of adaptive controlled systems. In *Proc. of the 9th IFAC/IFORS Symp. on Identification and System Parameter Estimation*, pages 657–662, 1991.

[60] K. Kumamaru, S. Sagara, and T. Söderström. Some statistical methods for fault diagnosis for dynamical systems. In R.J. Patton, P.M. Frank, and R. Clark, editors, *Fault Diagnosis in Dynamic Systems: Theory and Application*, pages 439–475. Prentice Hall, 1989.

[61] K. Kumamaru and T. Söderström. Fault detection and model validation using index of Kullback discrimination information (in Japanese). *Trans. of the Society of Instrument and control Engineers*, 22(10):1135–1140, 1986.

[62] K. Kumamaru, T. Söderström, S. Sagara, and K. Morit. On-line fault detection in adaptive control systems by using kullback discrimination index. In *Proc. of the 8th IFAC Symp. on Identification and System Parameter Estimation (Beijing, China)*, pages 1135–1140, 1988.

[63] Y.D. Landau. *Adaptive Control*. Marcel Dekker, Inc., New York, 1979.

[64] C.C. Lee. Fuzzy logic in control systems: Fuzzy logic controller, part I. *IEEE Trans. on Syst., Man, and Cybern.*, 20(2):404–418, 1990.

[65] C.C. Lee. Fuzzy logic in control systems: Fuzzy logic controller, part II. *IEEE Trans. on Syst., Man, and Cybern.*, 20(2):419–435, 1990.

[66] A.U. Levin and K.S. Narendra. Recursive identification using feedforward neural networks. *INT. J. Control*, 61(3):533–547, 1995.

[67] P. Lindskog. *Methods, Algorithms and Tools for System Identification Based on Prior Knowledge.* PhD thesis, Department of Electrical Engineering, Linköping University, 1996.

[68] P. Lindskog and L. Ljung. Tools for semiphysical modeling. *Int. J. of Adaptive Control and Signal Processing*, 9(6):509–523, 1995.

[69] L. Ljung. *System Identification: Theory for the User.* Prentice-Hall, Englewood Cliffs, N.J., 1987.

[70] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification.* MIT press, Cambridge, Mass, 1983.

[71] P.M. Mäkilä and J.R. Partingto. Robust identification of strongly stabilizable system. *IEEE Trans. on Automatic Control*, 37(7):1709–1716, 1992.

[72] M. Milanese and A. Vicino. Optimal estimation theory for dynamic systems with set membership uncertainty: An overview. *Automatica*, 27(6):997–1009, 1991.

[73] T. Murada, H. Ishibuchi, and H. Tanaka. Flowshop scheduling by genetic algorithm and its application to multi-objective problem (in Japanese). *Trans. of the Society of Instrument and control Engineers*, 31(5):583–590, 1995.

[74] S.R. Naidu, E. Zafiriou, and T.J. McAvoy. Use of neural networks for sensor failure detection in in a control system. *IEEE control Syst. Mag. (USA)*, 10(3):49–55, 1990.

[75] K.S. Narendra. Adaptive control using neural networks. In W.R. Miller III, R.S. Sutton, and P.J. Werbos, editors, *Neural Networks for Control*, pages 115–142. The MIT Press, Massachusetts, 1990.

[76] K.S. Narendra and K. Parthasathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks*, 1(1):4–27, 1990.

[77] O. Nelles and R. Isermann. A comparison between RBF networks and classical methods for identification of nonlinear dynamic systems. In *Proc. of 5th IFAC Symposium on Adaptive Systems in control and Signal Processing (Budapest)*, pages 219–224, 1995.

[78] B. Ninness. *Stochastic and Deterministic Modeling.* PhD thesis, The University of Newcastle, Australia, 1993.

[79] B. Ninness and G. Goodwin. Estimation of model quality. In *Proc. of the 10th IFAC Symposium on Identification (Copenhagen)*, volume 1, pages 25–44, 1994.

[80] B. Ninness and G. Goodwin. Estimation of model quality. *Automatica*, 31(12):1771–1797, 1995.

[81] R. Ortega and Y. Tang. Robustness of adaptive controllers-a survey. *Automatica*, 25(5):651–677, 1989.

[82] T. Ozaki, L.K. Hotta, H. Nakamura, R. Seki, H. Tamua, K. Tanabe, and H. Akaike. Nonlinear prediction of the water flow in an interconnected multi-reservoir power system. In *Proc. of the IFAC Conference on Water Resourse*, 1985.

[83] R.J. Patton, J. Chen, and T.M. Siew. Fault diagnosis in nonlinear dynamic systems via neural networks. In *International Conference on Control'94 (IEE London, UK)*, volume 2, pages 1346–1351, 1994.

[84] R.J. Patton and S.M. Kangethe. Robust fault diagnosis using eignstructure assignment of observers. In R.J. Patton, P.M. Frank, and R. Clark, editors, *Fault Diagnosis in Dynamic Systems: Theory and Application*, pages 99–154. Prentice Hall, 1989.

[85] M. Pottmann, H. Unbehauen, and D.E. Seborg. Application of a general multi-model approach for identification of highly nonlinear processes – a case study. *INT. J. Control*, 57(1):97–120, 1993.

[86] P. Pucar. *Modeling and Segmentation using Multiple Models.* PhD thesis, Linköping University, Sweden, 1995.

[87] A. Rault, D. Jaume, and M. Verge. Industrial process fault detection and localization. In *Proc. of the 9th IFAC Symp. on Identification (Budapest)*, volume 4, pages 1789–1794, 1984.

[88] K.C. Sharman and T.S. Durrani. Annealing algorithms for adaptive array processing. In *Proc. of the 8th IFAC Symposium on Identification and System Parameter Estimation (Beijing, China)*, 1988.

[89] J. Sjöberg. *Non-Linear System Identification with Neural Networks.* PhD thesis, Linköping University, Sweden, 1995.

[90] J. Sjöberg, H. Hjalmarsson, and L. Ljung. Neural networks in system identification. In *Proc. of the 10th IFAC Symposium on Identification (Copenhagen)*, volume 3, pages 49–72, 1994.

[91] J. Sjöberg and L. Ljung. Overtraing, regularization and searching for a minimum, with application to neural networks. *INT. J. Control*, 62(6):1391–1407, 1995.

[92] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deglon, P.Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

[93] T. Söderström and K. Kumamaru. On the use of Kullback discrimination index for model validation and fault detection. Technical Report UPTEC85 20 R, Institute of Technology, Uppsala University, 1985.

[94] T. Söderström and P. Stoica. *System Identification.* Prentice Hall International, Hemel Hempstead, 1989.

[95] T. Sorsa, H.N. Koivo, and H. Koivisto. Neural networks in process fault diagnosis. *IEEE Trans. on Syst., Man, and Cybern.*, 21:815–825, 1991.

[96] A. Srinivasan and C. Batur. Fault detection and isolation in an unsupervised learning environment. *Pattern Recognition Letter*, 15:235–242, 1994.

[97] M. Staroswiecki, J.P. Cassar, and V. Cocquempot. generation of optimal structured residuals in the paity sapce. In *Proc. of the 12th IFAC World Congress (Sydney)*, volume 5, pages 535–542, 1993.

[98] J. Suontausta, B.T. Tuoppila, and H.N. Koivo. Modeling of nonlinear systems using radial basis function networks. In *Proc. of the 10th IFAC Symposium on Identification (Copenhagen)*, volume 2, pages 325–330, 1994.

[99] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on Syst., Man, and Cybern*, 15(1):116–132, 1985.

[100] M. Tanaka, T. Hattonri, and T. Tanino. Jump detection and identification of linear systems by the genetic algorithm (in Japanese). *Trans. of the Society of Instrument and control Engineers*, 28(11):1383–1385, 1992.

[101] M.K. Tsatsanis and G.B. Giannakis. Time-varying system identification and model validation using wavelets. *IEEE Trans. on Signal Processing*, 41(12):3512–3523, 1993.

[102] J. Tsuji, H. Ohmori, and A. Sano. Adaptive control incorporating neural network (in Japanese). *Trans. of the Society of Instrument and Control Engineers*, 30(3):295–302, 1994.

[103] B. Wahlberg. System identification using laguerre models. *IEEE Trans. on Automatic Control*, 36(5):551–563, 1991.

[104] B. Wahlberg. Laguerre and kautz model. In *Proc. of the 10th IFAC Symposium on Identification (Copenhagen)*, volume 3, pages 1–12, 1994.

[105] B. Wahlberg and L. Ljung. Hard frequency-domain model error bounds from least-squares like identification. *IEEE Trans. on Automatic Control*, 37(7):900–912, 1992.

[106] H. Wang, M. Brown, and C.J. Harris. Fault detection for a class of unknown non-linear systems via associative memory networks. *Proc Instn Mech Engrs I, J. Syst. Contro Eng. (UK)*, 208(12):101–107, 1994.

[107] H. Wang and S. Daley. A fault detection method for unknown systems with unknown input and its application to hydraulic turbine monitoring. *INT J. of Control*, 57(2):247–260, 1993.

[108] L.X. Wang. *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Prentice-Hall, Englewood Cliff, N.L., 1994.

[109] L.X. Wang. Design and analysis of fuzzy identifiers of nonlinear dynamic systems. *IEEE Trans on Automatic Control*, 40(1):11–23, 1995.

[110] L.X. Wang and J.M. Mendel. Back-propagation fuzzy system as nonlinear dynamic system identifiers. In *Proc. of IEEE Int. Conf. on Fuzzy Systems (San Diego, CA)*, pages 1409–1418, 1992.

[111] L.X. Wang and M. Mendel. Fuzzy basis functions, universal approximation, and orthogonal least- squares learning. *IEEE Trans. on Neural Networks*, 3(5), 1992.

[112] C.W. Xu and Y.Z. Lu. Fuzzy model identification and self-learning for dynamic systems. *IEEE Trans. on Syst., Man, and Cybern.*, 17(4):683–689, 1987.

[113] T. Yamada and T. Yabuta. Dynamic system identification using neural networks. *IEEE Trans. on Syst., Man, and Cybern.*, 23(1):204–211, 1993.

[114] Z.J. Yang, T. Hachino, T. Tsugi, , and S. Sagara. Identification on parameters and time delay of continuous systems using the genetic algorithm. In *Proc. of the 10th IFAC Symposium on Identification (Copenhagen)*, volume 3, pages 657–662, 1994.

[115] R. Younce and C.E. Rohrs. Identification with parametric and non-parametric uncertainty. *IEEE Trans. on Automatic Control*, 37(7):715–728, 1992.

[116] Q. Zhang. Wavenet. *Public domain MATLAB toolbox Anonymous ftp: ftp.irisa.fr:/local/ wavenet*, 1993.

[117] Q. Zhang and A. Benveniste. Wavelet networks. *IEEE Trans. on Neural Networks*, 3(6):889–898, 1992.

[118] Qinghua Zhang. Using nonlinear black-box model in fault detection and isolation. In *Proc. of 35th IEEE Conference on Decision and Control (Kobe)*, 1996.

[119] C. Zhou, W. Jiang, and S. Fuju. The multiple mode armax model for nonlinear system its parameter identification. *Control Theory Appl. (China)*, 12(1):40–45, 1995.

[120] T. Zhou and H. Kimura. Simultaneous identification of nominal model, parametric uncertainty and unstructured uncertainty for robust control. *Automatica*, 30(3):391–402, 1994.