University of Alberta

Parameter and delay estimation of continuous-time models from uniformly and non-uniformly sampled data

by

Salim Ahmed

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**

in

Process Control

Department of Chemical and Materials Engineering

Edmonton, Alberta
Fall 2006

Your file Votre référence
ISBN: 978-0-494-22981-1
Our file Notre référence
ISBN: 978-0-494-22981-1

Abstract

This thesis is concerned with simultaneous estimation of the time delay and system parameters of continuous-time transfer function models. Estimation of the delay is different from the estimation of the rest of the parameters due to the fact that the delay does not appear explicitly in the model equation. The initiative undertaken in this research is based on the idea of bringing the delay term within the parameter vector. The idea was facilitated by a specific formulation of a linear filter for continuous-time identification. A new filter structure has been proposed and the ensuing iterative method has been developed in a novel way to estimate the delay plus other system parameters.

Over the last few decades, a significant number of new methods and techniques for system identification have been developed. More theoretical aspects of the identification problem such as the convergence of the parameter estimates have been addressed in much details. However, many practical problems in real world applications of different identification techniques remain unsolved. In this work, aspects of system identification with respect to practical implementation are considered.

Lack of availability of uniformly sampled data is a common yet often overlooked problem in real industrial data. A simple algorithm for identification from non-uniformly sampled output data has been proposed based on the idea of model based prediction. Techniques based on step test are commonly used in process industries for identification. Novel identification methods based on open loop and closed loop step test data have been proposed in this thesis that use raw industrial data without preprocessing. The methods are applicable even if the step input is applied when the process is not at steady state.

A multiple input multiple output (MIMO) model identification method is introduced that involves transformation of MIMO data into its single input single output (SISO) equivalents and uses SISO model identification algorithms for the purpose of identification. Model validation is a complementary step to the identification exercise. A validation scheme for SISO and MIMO continuous-time models is also presented in this thesis. The proposed method is applicable for models with delays.

Acknowledgements

First of all, I express my gratitude to my Lord, the most High.

Graduate study in process control at the University of Alberta is a relishing experience. I have been fortunate to work as a part of one of the best process control groups on the globe. I would like to take this opportunity to thank all its past and present members whose contribution have raised the status of this group to its current level.

It has been a privilege to have both Prof. Sirish L. Shah and Prof. Biao Huang as supervisors. It is the encouragement, inspiration and ingenious direction of the former and the continuous guidance, constructive criticism and friendly support of the later that made this thesis a reality. The weekly meeting with Prof. Huang was instrumental for each and every step of this research work. I gratefully acknowledge the outstanding guidance of my supervisors.

I would like to thank Aris Espejo and Edgar Tamayo of Syncrude Canada Ltd. for sharing with us the industrial perspectives of system identification. Financial support from Natural Sciences and Engineering Research Council of Canada (NSERC) is also acknowledged.

Thanks are due to my colleagues in the process control group for the informative discussions and stimulating debates. I am thankful to Amar Halim, Dr. Arun Tangirala, Dr. Bhushan Gopaluni, Dr. Bo Li, David Chang, Enayet Halim, Fungwei Xu, Gbenga Adeleye, Hailei Jiang, Dr. Harigopal Raghavan, Ian Alleleye, Jiandong Wang, Dr. Liqian Zhang, Monjur Murshed, Nikhil Agarwal, Dr. Ramesh Kadali, Ryuou Cheng, Rumana Sharmin, Dr. Shoukat Choudhury, Sien Lu, Syed Imtiaz, Dr. Vinay Kariwala, Dr. Weihua Li, Xiaori Hwang, Dr. Xin Huang, Yutong Qi and Dr. Zhengang Han. I am thankful to the members of Bangladeshi Students Association and Muslim Students Association at U of A for making the social life during my PhD study a memorable experience.

Many many thanks are due to my beloved wife Rozy for all her support, loving care and enduring patience during the study period. I thank my son, Fuad, who was a great source of motivation for the completion of this thesis. Thanks are also due to my loving brothers.

I am grateful to my parents and my grandparents. They were always and they still are my inspiration, my sources of energy and my guides for positive thinking.

dedicated to

all my teachers

specially

my parents

Contents

List of Tables

List of Figures

List of symbols

β, λ	: *filter parameters*
δ	: *time delay*
θ	: *parameter vector*
Φ	: *regressor*
Ψ	: *instrument matrix*
a_i	: *coefficients of the denominator polynomial*
b_i	: *coefficients of the numerator polynomial*
n	: *order of the denominator polynomial*
m	: *order of the numerator polynomial*
p	: *the derivative operator*
s	: *the Laplace operator*
u	: *process input*
v	: *measurement noise*
w	: *dither input*
y	: *process output*
A	: *denominator polynomial*
B	: *numerator polynomial*
G	: *the process model*
$\mathfrak{L}$	: *the Laplace Transform*
$\mathfrak{L}^{-1}$	: *inverse Laplace Transform*
N	: *data length*
N_u	: *no. of inputs*
N_y	: *no. of outputs*

List of abbreviations

CT	: *Continuous-time*
CTID	: *Continuous-time identification*
DT	: *Discrete-time*
DTID	: *Discrete-time identification*
FOPTD	: *First order plus time delay*
IV	: *Instrumental variable*
LD	: *Linear dynamic*
LF	: *Linear filter*
LS	: *Least-squares*
MCS	: *Monte Carlo simulation*
MIMO	: *Multiple input multiple output*
MMF	: *Method of multiple filters*
MSE	: *Mean-squared error*
NSR	: *Noise to signal ratio*
PID	: *Proportional integral derivative*
PMF	: *Poisson moment functional*
RIVC	: *Refined instrumental variable method for continuous-time models*
SISO	: *Single input single output*
SVF	: *State variable filter*

Chapter 1

Introduction

1.1 Prologue

The way a particular field of science develops depends on a combination of two forces: the socio-technical developments created by the evolution of the neighboring fields of science and the demands of the application world (Gevers 2003). A careful review of the developments in the field of system identification would reveal that the first of the two forces has much greater effect on the development process while the demands of the application world have not been taken into consideration to the required extent. An answer to the simple question: "what fraction of identification methods developed so far have been used in industries?" would help to understand the status of the problem. A natural question follows; "How can the requirements of industries be taken into account?" In this work, an effort has been made to develop new methods for system identification that address some practical issues of the application world. The research initiative is based on the view that requirements of the practical applications can be met by tailoring methods to match the nature of industrial data, by enhancement of the methods extensively used in industrial applications, by developing generalized framework for different problems (e.g. single input single output and multiple input multiple output model identification) and so on. It is the opinion of the author that although numerous techniques and software are available for system identification, industries are still in need of appropriate tools for simultaneous estimation of the time delay and the system parameters of continuous-time models. There are problems associated with industrial data that have not been dealt within the identification algorithms. Also, widely used tools such as the step response based methods require data in a particular form that is not readily available from industries. Work on multiple input multiple output (MIMO) continuous-time model identification is in its infancy although there has been a considerable amount of research for identifying single input single output (SISO) model and the problem of continuous-time model validation is yet to be addressed. With these perspectives in mind, initiatives have been taken in this work to

- simultaneously estimate the model parameter and the time delay
- estimate parameters from raw data
- estimate parameters from non-uniformly sampled data
- enhance the step response based identification techniques
- consider MIMO identification within the SISO framework
- develop a scheme for model validation.

As mentioned earlier, the problem was seen from an application perspective and the presentation is more concerned with development of appropriate theories to be useful for implementation purposes. Consequently the more theoretical problems such as the proof of convergence of iterative algorithms have not been addressed.

An important focus of this work is the problem of estimation of the process time delay. In system identification, the time delay estimation and the estimation of other model parameters are often considered as two disjoint problems and different approaches are taken for their solutions. In this work, the delay is treated in the same way as the other parameters and estimated simultaneously. The approach under consideration is that of the linear filter method for continuous-time identification. It is the treatment of the delay that makes the proposed work different from the existing linear filter methods. To emphasize the importance of time delay estimation and highlight the significance of the current work a brief discussion on time delay and a review of time delay estimation methods are presented. But first a note on the motivations for continuous-time identification (CTID) and some general discussion on CTID is provided.

1.2 Why continuous-time identification (CTID)?

One of the primary objectives of system identification is to know the system. It is essential to know and understand a system before it is handled, i.e., manipulated or controlled (Unbehauen and Rao 1998). Understanding requires proper analysis of the system and models are the most appropriate tools to analyze. For the purpose of analysis, it is easier to relate to and characterize a process from the parameters of its continuous-time model. The coefficients in discrete-time models do not offer the same ease and appeal of physical interpretation as do the parameters in continuous-time models (Unbehauen and Rao 1990). On the other hand, characterizing systems by discrete-time models makes sense as the mathematical characterization of the systems matches the serial processing nature of the digital computer (Young 1981). This has led to significant activities in the field of discrete-time identification (DTID). However, the uniqueness of continuous-time identification (CTID) and the advantages of the use of continuous-time models have led to increasing interest in CTID over the last few years. The inherent continuous-time nature of physical systems, strong correlations of the model parameters with the system properties, the use of continuous-time models in controller design etc. acted as major motivating forces for CTID. For details on the advantages of CTID the interested readers are referred to (Sinha and Rao 1991, Unbehauen and Rao 1998, Rao and Unbehauen 2006). Numerical illustrations of the relevance of direct continuous-time identification have been discussed in (Rao and Garnier 2002) and the advantages of CT models in terms of the physical interpretation have been provided in (Young *et al.* 2003, Zak *et al.* 2003).

1.3 CTID: Basic principle and a brief review

Continuous-time identification is a vast area rich in well established methods. A number of survey papers (Unbehauen and Rao 1990, Young 1981), books (Sinha and Rao 1991, Unbehauen and Rao 1987) and review papers (Unbehauen and Rao 1998, Rao and Unbehauen 2006) give a broad overview of the existing methods. CT models and the associated CTID methods can be categorized based on different criteria e.g. parametric vs. nonparametric, time domain vs. frequency domain, direct vs. indirect and so on. We consider time domain identification of continuous-time transfer function (or equivalently linear ordinary differential equation) models of linear time-invariant systems using the direct approach.

The main challenge in continuous-time identification arises due to the appearance of the derivative terms in the estimation equation. Generation of the derivatives from the input output measurements is not desirable due to the presence of noise in the data. This problem is overcome by performing some suitable linear dynamic (LD) operation on the estimation equation. The LD operation is such that it, while retaining the parameters of the continuous model in their original form, facilitates generation of the appropriate measurements for the parameter estimation equation and overcome the need for computing derivatives of the signals. As presented in (Unbehauen and Rao 1987), figure 1.1 describes the basic procedure for parameter estimation in CTID. The general scheme for the estimation of a continuous-time model involves

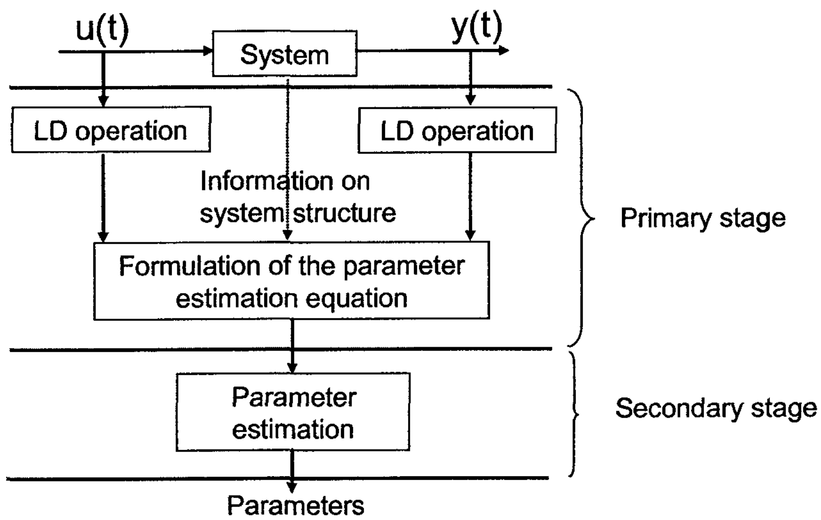


Figure 1.1: *General scheme for continuous-time parameter estimation.*

two stages, namely the primary stage and the secondary stage. In the primary stage, performing the linear dynamic operation, the continuous-time representation of the system is converted into a set of algebraic equations indexed in accordance with the time samples. In the secondary stage, the CT parameters are estimated by an adequate statistical estimation procedure from the system of algebraic equations. Most of the well-known linear regression methods can be used in the parameter estimation stage with some modifications (Garnier *et al.* 2003). A classical approach in the secondary stage has been to use the least-squares (LS) method. However, the LS method almost always gives biased estimates. The reason for this has been that the estimation of the derivatives almost always corrupted the noise sequence and made it colored (Gillberg 2004). Instrumental variable methods have been widely popular as a means for reducing this bias (Young 1981).

Different CTID methods differ on the basis of the linear dynamic operation involved in the methods. There are mainly three different approaches for linear dynamic operations: the method of modulating functions, the linear filter methods and spectral characterization of signals. Details on these three approaches can be found in the books (Sinha and Rao 1991, Unbehauen and Rao 1987). Brief discussion on the different approaches can be found in the theses (Gillberg 2004, Mehta 1996, Missailidis 2000). Detailed reviews on modulating functions are available in (Preisig and Rippin 1993*a*, Preisig and Rippin 1993*b*, Preisig and Rippin 1993*c*). A long list of references on CTID methods is also available in (O'Dwyer 2000). This work considers a linear filter method and a brief review of the existing linear filter methods is provided in chapter 2.

Linear filtering is one of the most commonly used linear dynamic operations for continuous-time identification. In fact, many of the linear dynamic operations can be interpreted as pre-filtering of input and output signals (Sagara and Zhao 1991). Process signal pre-filtering is indeed a useful way to improve statistical efficiency in system identification and yields lower variance of the parameter estimates (Garnier *et al.* 2003). The linear filter used for CTID plays the above role in addition to its role in avoiding direct differentiation of the noisy signals.

1.4 Time Delay

1.4.1 Definition, occurrence and importance

The time delay is an important physical property of a system. It is a measure of the time for which the response to an applied force is delayed in its effect (Shinskey 1967). It may be defined as the time interval between the start of an event in one point in a system and its resulting action at another point (O'Dwyer 1999). Whenever material or energy is physically moved in a process or plant there is a time delay associated with the movement (Seborg *et al.* 1989). Time delay is also referred to as dead time, transportation lag or distance-velocity lag. In addition to this pure time delay, apparent time delays may also result due to measurement processes or in the identification exercise when a higher order process is approximated by a lower order model.

Time delays arise in physical, chemical, biological and economic systems as well as in signal processing operations. A long list of examples of time delayed systems of different fields are presented in (O'Dwyer 2000) and the references therein. One distinctive characteristic of the process control field, as compared to control of most mechanical or electrical systems, is the common occurrence of time delays (Seborg *et al.* 1989). There are few processes where it is not present in some form (Shinskey 1967). O'Dwyer (2000) refers to a large number of sources and presents a list of processes such as pulp and paper manufacturing, diffusion systems, activated sludge processes, heating and ventilation systems, combustion processes, fertigation process, etc. as examples of processes having time delays. Latour *et al.* (1967) quoted a variety of sources to support the use of models with time delays for processes such as liquid-liquid extraction, mixing in agitated vessels, some heat exchangers, distillation columns and some chemical reactors.

Time delay has a significant bearing on the achievable performance for control systems (Wang and Zhang 2001*b*). Many popular control synthesis techniques, such as the minimum variance controller, exhibit poor or even unstable control performance if the delay is not correctly known. Moreover in general, a fundamental requirement for the proper tuning of control algorithms is the knowledge of phase contribution of the delay over the frequency of interest (Ferretti *et al.* 1991). When there is a long dead time in a process, the control performance obtained with a PID controller is limited. From a frequency response perspective, a time delay adds significant phase lag to the feedback loop, which adversely affects closed-loop stability. To improve the performance of systems with time delay, special control strategies such as the Smith predictor scheme have been developed which provide time-delay compensation. But these model based strategies depend largely on the model of the process. In case of inaccuracy in the model, controller performance deteriorates, perhaps to the point of instability. Schleck and Hanesian (1978) performed a detailed study analyzing the effect of model errors on the Smith predictor for a first order plus time delay process and found that if the assumed time delay is not within 30% of the actual process time delay, the predictor is inferior to a PI controller with no time delay compensation (Seborg *et al.* 1989).

For applications where the parameters of the estimated models are used as a measure of some physical properties e.g. in (Milinkovic 1997), the delay estimation, if any, should be correct. Otherwise, the inferred physical property can be grossly inaccurate. Also for cases where the models are used for prediction, the correct estimation of delay is equally important.

In summary, irrespective of the end use of the process model, if the process contains a delay, then estimation of the delay is important for the model to be useful.

1.4.2 Review of time delay estimation methods

Due to the common occurrence of the delay and the importance of its accurate estimation, time delay estimation techniques have drawn widespread interest in the research community. As a result, a large number of methods have evolved to "obtain an estimate of" or to "choose" time delay. In this section we will discuss methods which may be considered similar to our approach in terms of model structure and estimation procedure. More specifically we will consider techniques that deal with estimation of delay and other parameters. But as there are a large number of methods that estimate only the time delay we will review some of these very briefly. We will consider the time delay estimation problem from a control perspective and not from a signal processing perspective. This limits the discussion to the so called active time delay estimation problem.

Before presenting the review let us formulate the problem mathematically. A linear single input single output (SISO) system with time delay is described by a continuous time model as:

$$y(t) = G(p)u(t-\delta) + v(t) \tag{1.1}$$

where,

$$G(p) = \frac{B(p)}{A(p)}$$

$$A(p) = a_n p^n + a_{n-1}p^{n-1} + \cdots\cdots + a_1 p + a_0$$

$$B(p) = b_m p^m + b_{m-1}p^{m-1} + \cdots\cdots + b_1 p + b_0$$

$G(p)$ is the process model without the delay, $y(t)$ and $u(t)$ are measurable process output and input, respectively, $v(t)$ is the measurement error, $p = d/dt$ is the derivative operator and δ is the process delay in units of time.

The objective of system identification is to find an estimate of the parameters a_i, $i = 1 \cdots n$, b_j, $i = 0 \cdots m$ and δ. While the parameters a_i and b_j appear explicitly as coefficients of the signals and their derivatives, the delay δ appears implicitly in the estimation equation. This makes the estimation of the delay different from the estimation of other parameters.

Methods to estimate only the delay

In many of the methods available in the literature, the parameters other than the delay are considered to be nuisance. A detailed review of such methods can be found in (Björklund 2003) where the time delay estimation (TDE) problem is classified as the so called active TDE and passive TDE problem. The TDE problem encountered in the field of control is categorized as active TDE problem where both the input and output are known. The active time-delay estimation methods are further classified into several classes. We will discuss only one of the classes namely the time-delay approximation methods as it is the most commonly used. In the methods belonging to this type, the input and output signals are represented in a certain basis and the time-delay is estimated from an approximation of the relation (a model) between the signals in this basis. Depending on the basis there are the following subclasses.

Time domain approximation methods

The basis consists of impulse functions. The time delay is the delay for the impulse response to start (Björklund 2003, Carlemalm *et al.* 1999, Isaksson 1997, Kurz and Goedecke 1981). Finding the maximum of cross-correlation between input and output, which is a common method (Hero *et al.* 1998, Carter 1993), is in principle the same thing.

For dynamic systems if we use the time instant corresponding to the maximum of the impulse response as an estimate of time delay, we would get a bias in the estimate (Björklund 2003). One has to separate first the time delay and dynamics of the system to get the time delay. Also we need to estimate the impulse response from the input output data which itself can introduce errors. The uncertainty of the estimate of the impulse response may result in inaccuracies as shown by Björklund (2003). Moreover thresholding itself is a non trivial job for these methods.

Frequency domain approximation methods

The basis consists of complex sinusoids $e^{i\omega t}$. A time delay (δ) is equivalent to a phase shift and is estimated from the phase of $e^{-i\omega\delta}$ (Grennberg and Sandell 1994, Houghton and Reeve 1995, Isaksson 1997). In these methods an approximation of the system including the time delay is presented in the frequency domain. A Laguerre model was used by Isaksson *et al.* (2001) for the approximation. Model of other structures such as FIR, ARX or OE can also be used. Time delay can be estimated using the phase of the discrete-time allpass part (DAP methods). Björklund (2003) showed with the help of simple examples how the uncertainty in the zero location can lead to failure of the DAP estimates and concluded that DAP methods are inherently non-robust. Zero guarding is needed to make the methods more robust.

Laguerre domain approximation methods

The time delay is estimated from a relation between the input and output signals expressed in continuous time or discrete time Laguerre functions (Fischer and Medvedev 1999). A necessary condition for the Laguerre domain approximation estimation methods to be successful is that the input and output signals can be represented accurately in the Laguerre domain. But this is hard for random binary type signals. Also several parameters have to be selected by the user for this purpose.

Methods to estimate the delay and system parameter

In this section we will present a brief review of the methods that estimate delay plus other process parameters. A detailed review on these methods can be found in (O'Dwyer 2000). These methods can be broadly classified into time domain and frequency domain methods. They may be off-line or on-line and can be applicable either in open loop or in closed loop environment. However, the review is limited to the time domain methods.

Methods based on step response

In fact most, if not all, time domain methods to directly estimate the time delay along with other parameters are based on the step test. Many of these methods can be categorized as graphical methods where a few points of the step response are used to get a model of limited orders. A number of these graphical methods for open loop are described in (Oldenbourg and Sartorius 1948, Rake 1980, Seborg *et al.* 1989, Unbehauen and Rao 1987). For the closed-loop environment the delay is often approximated by a rational polynomial. There are some step test methods applicable for model with first or second order. A detailed review on the step response based techniques appears in chapter 3. The main limitation of the graphical methods is that they use only a few points of the step response and consequently may not be robust to noise.

Another group of identification methods uses the area under the step response curve to estimate the process parameters. The method of moments (Ingimundarson and Hagglund 2001) for closed loop identification of FOPTD model is one of the often-cited method belonging to this category. Wang and Zhang (2001*b*) proposed an integral equation approach for open loop systems that uses various order integrals of the step response data to identify models of any order.

Identification methods from step response data enjoy the advantages of the simplicity of the experiment. However, the parameters identified may vary with process operating conditions and with the size of the step change and its direction (Seborg *et al.* 1989). Smith and Corripio (1985) concluded that the precision of step response methods to estimate second or higher order process parameters is very low.

Approximation of time delay

In this class of methods the time delay term is approximated by methods such as Padé approximation, Laguerre approximation etc. or by a truncated Taylor series expansion. The result of this approximation is generally an augmented model. The parameters of the resulting model are then estimated using a conventional delay-free parameter estimation method from which the delay may be deduced. Alternatively, the method of over-parameterizations (Ferretti *et al.* 1991) is used which involves subsuming the delay term into an extended z *domain* numerator polynomial. The parameters are estimated recursively and the delay is calculated from the identified numerator polynomial.

The approximation methods become computationally intensive if a high order approximation is used. On the other hand, a lower order approximation introduces error in the approximation. As a result, there exist a trade-off between accuracy and computational load. Also the high order numerator polynomial increases the likelihood of the presence of common factors in the numerator and denominator polynomials, rendering identification more difficult (O'Dwyer 1999).

Model set estimation

Methods in this category estimate a number of different models for different delays within a predefined range and often for different set of model orders. A suitable cost function that is a measure of the model quality is estimated for every set of model parameters. Finally the set that gives the optimum value of the cost function is chosen. Some of the multiple model estimation methods e.g. in (Hsia 1977, Young 2002) obtain the model orders along with other parameters and delay while others e.g. (Rao and Sivakumar 1976, Unbehauen and Rao 1987) estimate only the delay and process parameters.

The attractiveness of model set estimation methods is that the grid search will facilitate the estimation of parameters corresponding to the global minimum of the cost function, even in the presence of local minima provided that enough models are estimated. However, this is computationally intensive (O'Dwyer 1999).

Optimization techniques

The so called gradient methods are the most commonly used optimization techniques used for simultaneous estimation of time delay and process parameters. These methods involve updating of the parameter vector and delay based on the information on a cost function. Typical gradient algorithms are the Newton-Raphson, the Gauss-Newton and the steepest descent algorithms, which differ in their updating vectors. Examples of gradient methods can be found in (O'Dwyer 1999). A major problem with the gradient methods is that the error surface is often multimodal. This is why locating global optimum needs multiple optimization runs each initiated at a different starting point which makes the method computationally intensive.

1.5 Organization of the thesis

As mentioned earlier, the purpose of this work is to develop methods for simultaneous estimation of the time delay and remaining system parameters of continuous-time transfer function models. The linear filter method is used to facilitate the conversion of the separate (delay and system parameters) identification problem into a simultaneous parameter estimation problem. On the constraint side of the problem are the demands of the real application world. Search for a solution to this problem led this work in different directions and the resulting developments are summarized. The contents of this thesis have been divided into seven different chapters. The main chapters (Chapter 2- 6) are independent of each other to some extent and each contains a brief review of the relevant literature. Presentation of results are also done chapter-wise.

A brief summary of each individual chapter appears next.

- The current chapter, chapter 1, provides an introduction to the thesis. It includes a discussion on continuous-time identification and a review of the estimation of time delay.

- Chapter 2 dwells on the linear filter method. It introduces the new filter structure and details the linear filter method based on the new filter. In fact, chapter 2 contains the core work of this thesis that is subsequently used in the remaining chapters. The problems of the least squares estimation and its solution and other implementation issues of the proposed linear filter method have also been addressed in this chapter.

- Chapter 3 focuses on identification of models from non-uniformly sampled data. A specific type of non-uniformity in data, common in data from process industries, is considered. An alternative form of the linear filter proposed in chapter 2 is presented in this chapter. An iterative prediction scheme is proposed that makes the linear filter method applicable for non-uniformly sampled output data. Details of the input only model used in the iterative prediction algorithm is also provided.

- Novel identification techniques from open loop and closed loop step response data are presented in chapter 4. The methods directly use raw industrial data without preprocessing. Also one is not required to first bring the process to a steady state before the step input is applied.

- Chapter 5 deals with MIMO system identification. A decomposition technique that enables the use of SISO identification methods for the identification of MIMO processes is introduced.

- Model validation is often seen as an integrated part of model identification. In chapter 6 a model validation scheme based on the asymptotic local approach is presented. The procedures for validation of both SISO and MIMO models are detailed therein.

- Finally in chapter 7, the main contributions of the thesis are summarized and future directions on work in this area are outlined.

Chapter 2

A New Linear Filter Method [1]

2.1 Introduction

The origin of filtering for the purpose of identification can be traced back to an article by Valstar (1963). In that article, rarely cited in the literature of linear filter methods, the author introduced a number of filters for the purpose of, as the author named it, 'tracking' transfer functions. Along with the idea of integration approach, the basic of low-pass filtering was presented by Valstar in terms of a r-c (resistor-capacitor) circuit. In a contemporary study, Young (1964) developed a technique of process parameter determination very similar to the low pass filtering approach introduced by Valstar (1963). However, the mathematical justification provided in (Young 1964) for the use of signals originating from the successive low pass filtering of the process forcing function and output variable is somewhat different and possibly more straightforward. Although the article by Young (1964) appeared as a discussion on the article by Valstar (1963), it is the simplicity of the presentation and closeness to the spirit of continuous-time identification that gave Young's article more popularity in the field of system identification. In fact this method, commonly known as method of multiple filters (MMF), formed the basis of linear filter methods developed subsequently. The MMF, also known as the state variable filter (SVF) method, has been used in continuous-time identification since then and further developments (Wang and Gawthrop 2001) as well as application (Wang *et al.* 2004*b*) have been reported in the literature.

[1] A modified form of this chapter has been published as:

Ahmed, S., B. Huang and S. L. Shah (2006). Parameter and delay estimation of continuous-time models using a linear filter, *Journal of Process Control*, **16**(4) 323-331

Similar to the SVF method is the Poisson moment functional (PMF) approach which, to the best of the author's knowledge, was introduced first by Fairman (1971). Later on different researchers e.g., Bastogne *et al.* (2001), Garnier *et al.* (2000), Saha and Rao (1983) made significant contribution and made the PMF method the most popular of the linear filter methods used in continuous-time identification.

The integral equation approach which we consider within the framework of linear filter method was first proposed by Diamessis (1965). The integral filter can be treated as just another filter with all of the poles in the origin. Subsequent developments of the integral equation approach can be attributed to Whitfield and Messali (1987), Sagara and Zhao (1990), Wang and Zhang (2001*b*), Hwang and Lai (2004) and so on.

Use of filters with different structures and user specified parameter(s) led to the search for an optimal filter. Young and Jakeman (1980) presented the refined instrumental variable method for continuous-time identification (RIVC). The method was later revisited by Young (2002). The RIVC method is considered as the optimal linear filter method for continuous-time identification. In this approach, the filter has the transfer function $\frac{1}{A(s)}$ where $A(s)$ is the denominator of the process transfer function. Since $A(s)$ is unknown, it is generated within the algorithm via an iterative procedure. However, the idea to iteratively use the process denominator for filtering input output data was presented first by Steiglitz and McBride (1965) for discrete-time identification.

The purpose of our work is to develop a linear filter method to simultaneously estimate the process parameters and the time delay of a process. Most of the currently available linear filter methods estimate the process parameters provided that the time delay is known in the parameter estimation step. This complicates the identification scheme and makes the identification algorithm a disjoint two-step procedure. The goal of this work is to develop a method that considers the delay as another parameter of the model and estimates the entire parameter set wholistically. We will provide a brief review on the time delay estimation techniques used in different linear filter methods. But first we discuss the basic mathematical formulation of the linear filter (LF) methods.

2.2 Mathematical formulation of the LF method

To describe the general approach of the LF method, let us consider a linear time-invariant single input single output (SISO) model with time delay described by

$$y(t) = G(p)u(t - \delta) + v(t) \tag{2.1}$$

where,

$$G(p) = \frac{B(p)}{A(p)}$$
$$A(p) = a_n p^n + a_{n-1} p^{n-1} + \cdots\cdots + a_1 p + a_0$$
$$B(p) = b_m p^m + b_{m-1} p^{m-1} + \cdots\cdots + b_1 p + b_0$$

G is the process model without the delay, $y(t)$ and $u(t)$ are measurable process output and input, respectively, $v(t)$ is the measurement error, $p = d/dt$ is the derivative operator and δ is the process delay in units of time.

In the parameter estimation stage it is assumed that the orders of the process, n and m, are known and $n \geq m$. Also, without loss of generality it can be assumed that $a_0 = 1$. So the objective is to derive an estimate of the parameter vector, $[a_n\ a_{n-1} \cdots a_1\ b_m\ b_{m-1} \cdots b_0\ \delta]^T$, from a given set of measurements of $y(t)$ and $u(t)$.

To describe the general formulation of the linear filter methods, we express the differential equation model presented in eqn(2.1) in equation error formulation

$$\mathbf{a}_n \mathbf{y}^{(n)}(t) = \mathbf{b}_m \mathbf{u}^{(m)}(t - \delta) + e(t) \tag{2.2}$$

where,

$$\mathbf{a}_n = [a_n\ a_{n-1} \cdots a_0] \in \mathbb{R}^{1\times(n+1)}$$
$$\mathbf{b}_m = [b_m\ b_{m-1} \cdots b_0] \in \mathbb{R}^{1\times(m+1)}$$
$$\mathbf{y}^{(n)}(t) = \left[y^{(n)}(t)\ y^{(n-1)}(t) \cdots y^{(0)}(t)\right]^T \in \mathbb{R}^{(n+1)\times 1}$$
$$\mathbf{u}^{(m)}(t-\delta) = \left[u^{(m)}(t-\delta) \cdots u^{(0)}(t-\delta)\right]^T \in \mathbb{R}^{(m+1)\times 1}$$
$$e(t) = A(p)v(t)$$

$y^{(i)}$ and $u^{(i)}$ are i^{th} order time derivatives of y and u, respectively, i.e. $y^{(i)}(t) = p^i y(t)$ and $u^{(i)}(t) = p^i u(t)$.

Taking Laplace transformation on both sides of eqn(2.2), we can write

$$\mathbf{a}_n \mathbf{s}^n Y(s) = \mathbf{b}_m \mathbf{s}^m U(s) e^{-\delta s} + \mathbf{c}_{n-1} \mathbf{s}^{n-1} + E(s) \tag{2.3}$$

$Y(s)$, $U(s)$ and $E(s)$ are the Laplace transforms of $y(t)$, $u(t)$ and $e(t)$, respectively, and

$$\mathbf{s}^n = \left[s^n \; s^{n-1} \cdots s^0\right]^T \in \mathbb{R}^{(n+1)\times 1} \tag{2.4}$$

The elements of $\mathbf{c}_{n-1}$ capture the initial conditions of the output and are defined by

$$\mathbf{c}_{n-1} = \left[c_{n-1} \; c_{n-2} \cdots c_0\right] \in \mathbb{R}^{1\times n} \tag{2.5}$$

$$c_{n-i} = \mathbf{h}_i \mathbf{y}^{n-1}(0), \quad i = 1 \cdots n \tag{2.6}$$

$$\mathbf{h}_i = \left[\mathbf{0}^{1\times(n-i)} \; a_n \cdots a_{n-(i-1)}\right] \in \mathbb{R}^{1\times n} \tag{2.7}$$

$$\mathbf{y}^{(n-1)}(0) = \left[y^{(n-1)}(0) \; y^{(n-2)}(0) \cdots y(0)\right]^T \in \mathbb{R}^{n\times 1} \tag{2.8}$$

Here, we consider that the input is initially at rest. However, non-zero initial conditions of the input can be handled in the same way and this has been presented in chapter 4. Now, consider a causal filter described in Laplace domain as $F(s)$. If we apply the filtering operation on both sides of eqn(2.3) we end up with the formulation

$$\mathbf{a}_n \mathbf{s}^n F(s) Y(s) = \mathbf{b}_m \mathbf{s}^m F(s) U(s) e^{-\delta s} + \mathbf{c}_{n-1} \mathbf{s}^{n-1} F(s) + F(s) E(s) \tag{2.9}$$

Taking inverse Laplace transformation, $\mathfrak{L}^{-1}$, eqn(2.9) can be expressed in time domain

$$\mathbf{a}_n \mathbf{y}_F^{(n)}(t) = \mathbf{b}_m \mathbf{u}_F^{(m)}(t-\delta) + \mathbf{c}_{n-1} \mathbf{f}^{n-1}(t) + e_F(t) \tag{2.10}$$

with

$$\begin{aligned} \mathbf{y}_F^{(n)}(t) &= \left[y_F^{(n)}(t) \; y_F^{(n-1)}(t) \cdots y_F(t)\right]^T \\ y_F^{(i)}(t) &= \mathfrak{L}^{-1}[s^i F(s) Y(s)] \\ \mathbf{f}^{n-1}(t) &= \mathfrak{L}^{-1}[\mathbf{s}^{n-1} F(s)] \end{aligned}$$

Next, the filtered variables and their time derivatives are generated and eqn(2.10) is reformulated to give a standard form of least squares equations. In some methods, certain algebraic manipulations are carried out to avoid the derivative operation. In the linear integral method, time derivatives of signals are not involved; however, different order integrals of the variables are needed. The different linear filter methods differ in terms of the structure of the filter. Described below are the structures of different filters commonly used in literature.

2.3 Structures of existing filters

State variable filters (SVF)

The minimal order state variable filter has the form

$$F(s) = \left(\frac{\beta}{s+\lambda}\right)^n \tag{2.11}$$

PMF filter

The minimal order form of the Generalized Poisson Moment Functional (GPMF) approach uses the filter of the following form

$$F(s) = \left(\frac{\beta}{s+\lambda}\right)^{n+1} \tag{2.12}$$

For the basic PMF approach $\beta = 1$ while for Normalized PMF $\beta = \lambda$.

RIVC filter

The optimal RIVC filter is defined theoretically as $1/A(s)$ where $A(s)$ is the denominator of the system transfer function. Since $A(s)$ is unknown, it is generated within the algorithm via an iterative procedure.

2.4 Time delay estimation in LF methods

In the literature on linear filter methods, Only a few articles considers the problem of time delay estimation. In most cases it is assumed that there is no delay or the delay is known. Rao and Sivakumar (1976) presented the iterative shift algorithm that can be categorized as a model set estimation technique. In this approach, the available data is divided into a number of segments and for a guessed value of the delay the other model parameters are estimated using the poisson moment functional method. The basic idea is that the estimated parameters will be invariant with data on various subintervals if the delay is correctly guessed. With this in mind an error function is defined as the sum of the distances between two parameter vectors estimated using data from successive subintervals. Parameters are then solved by minimizing the error function with respect to the delay. Saha and Rao (1983) proposed another method also based on the PMF approach for processes having time delay smaller than the sampling interval. In this technique the filtered delayed input is expressed in terms of the filtered input at the previous and the following time instants by making a linear interpolation. Thus the input function appears with the unknown delay term. The estimation equation is then rearranged to place the delay term in the parameter vector and estimate it along with other parameters. Hsia (1977) proposed a model set estimation technique that consists of computing least-squares parameter estimates for a number of assumed time delay values which are integral multiples of the sampling interval and choosing that parameter set which yield the minimum mean squared error. Another approach is to choose the delay term in the same way as model orders are chosen. Young (2002) estimated model parameters for a set of model orders and delay; then based on the coefficient of determination and Young's information criterion chose the time delay and the model orders.

2.5 Development of a new linear filter method

2.5.1 Objective

Linear filter methods have been used in the field of continuous-time identification over a considerable time period. Due to the effectiveness and simplicity of the approach, they have found widespread applications and drawn much interest from the system identification community. However, the estimation of time delay along with continuous-time model parameters has remained an unsolved problem.

In this work, the primary objective is to develop a new linear filter method that would estimate the time delay along with other model parameters. Also the method should not be limited in application for any specific type of input signal.

2.5.2 Basic idea

In the parameter estimation equation of the linear filter method (eqn(2.10)), the time delay appears as an implicit parameter and cannot be estimated directly. To be solvable using the idea of regression analysis, the time delay should appear as an explicit parameter in the parameter vector. To get the delay in the parameter vector it is necessary that it appears explicitly in the estimation equation.

To have an explicit appearance of the delay term in the estimation equation and get it as an element in the parameter vector, we propose a filter having a first order integral dynamics along with a lag dynamics of order n. The filter transfer function $F(s)$ may have the following different forms

$$I : \frac{\beta^n}{s(s+\lambda)^n} \qquad II : \frac{\beta^n}{s(s+\lambda_1)(s+\lambda_2)\cdots(s+\lambda_n)} \qquad III : \frac{1}{sA(s)}$$

The reason to include an integrator in the filter is to generate an integration term of delayed input in the estimation equation. This integrated delayed input signal, which represents a certain area under the input curve, can be expressed by subtracting 3 sub-areas from the integrated input signal. By doing so, the delay, δ, becomes an explicit parameter in the estimation equation. Details on this idea is provided in the following section.

In this thesis, the discussion is limited to the filters I and III. Filter II has the disadvantage of having distinct poles to be specified by the user. However, use of a filter having distinct poles have certain advantages for special type of systems having their poles far apart or the multiple time-scale systems.

Filter I is similar in structure to that of the state variable filter (SVF) and PMF filter and have the advantages of these structures. One specific advantage of the PMF structure is that the different order derivatives of the filtered signals can be expressed in terms of filtered signal and the derivative operation is avoided. However, filter I has two user specified parameter. In most cases β is set equal to λ and only the λ is to be specified. Filter III can be considered as an extension of the RIVC filter and has the advantages of the RIVC filter. It uses the idea of adaptive filtering and needs only an initial guess of $A(s)$. The existing literature as well as the current study shows that the initial choice of $A(s)$ has little effect on the parameter estimated finally. In fact the RIVC filter is claimed to be the optimal filter as its frequency response exactly matches the frequency response of the process.

In this chapter we will present detailed mathematical derivation for the filter I. The mathematical formulation for the filter III is presented in chapter 3. Before presenting the mathematical derivation first we summarize the necessary assumptions:

- The system is time-invariant and causal
- The polynomials $A(p)$ and $B(p)$ are co-prime

- The disturbance $v(t)$ is independent of $u(t)$

2.5.3 Mathematical formulation

In this section we present the new linear filter method for the filter having the following structure.

$$F(s) = \frac{\beta^n}{s(s+\lambda)^n} \tag{2.13}$$

For this filter, the different order derivatives of a filtered signal can be expressed as linear combination of the signal filtered with a set of filters having only lag dynamics of different orders. Mathematically this can be expressed as

$$\mathbf{s}^q F(s) = \mathbf{\Lambda}_q \mathbf{F}_n(s) \tag{2.14}$$

with

$$\mathbf{\Lambda}_q(i,j) = \begin{cases} \Omega_{n-j}^{n-i}\beta^n(-\lambda)^{j-i} & i = 1\cdots q,\, j = (n-q+i)\cdots n \\ -\frac{\beta^n}{\lambda^{n+1-j}} & i = q+1,\, j = 1\cdots n \\ \frac{\beta^n}{\lambda^n} & i = q+1,\, j = n+1 \\ 0 & i = 1\cdots q,\ j = n+1 \text{ and } \ j < (n-q+i) \end{cases} \tag{2.15}$$

$$\mathbf{\Lambda}_q \in \mathbb{R}^{(q+1)\times(n+1)}$$

$$\Omega_j^i = \frac{i!}{j!(i-j)!} \tag{2.16}$$

$$\mathbf{F}_n(s) = \left[\frac{1}{(s+\lambda)}\ \frac{1}{(s+\lambda)^2}\cdots\frac{1}{(s+\lambda)^n}\ \frac{1}{s}\right]^T \in \mathbb{R}^{(n+1)\times 1} \tag{2.17}$$

Adopting these notations, eqn(2.9) takes the form

$$\mathbf{a}_n\mathbf{\Lambda}_n\mathbf{F}_n(s)Y(s) = \mathbf{b}_m\mathbf{\Lambda}_m\mathbf{F}_n(s)U(s)e^{-\delta s} + \mathbf{c}_{n-1}\mathbf{\Lambda}_{n-1}\mathbf{F}_n(s) + E_F(s) \tag{2.18}$$

where, $\mathbf{\Lambda}_n$ and $\mathbf{\Lambda}_m$ are defined in the same way as $\mathbf{\Lambda}_q$. Taking inverse Laplace transform, $\mathfrak{L}^{-1}$, on eqn(2.18), at the $k-th$ sampling instant i.e. $t = t_k$, we can write

$$\mathbf{a}_n\mathbf{\Lambda}_n\mathbf{y}_{F_n}(t_k) = \mathbf{b}_m\mathbf{\Lambda}_m\mathbf{u}_{F_n}(t_k-\delta) + \mathbf{c}_{n-1}\mathbf{\Lambda}_{n-1}\mathbf{F}_n(t_k) + e_F(t_k) \tag{2.19}$$

where,

$$\mathbf{y}_{F_n}(t_k) = [y_{f_1}(t_k)\ y_{f_2}(t_k)\cdots y_{f_n}(t_k)\ y_I(t_k)]^T \in \mathbb{R}^{(n+1)\times 1}$$

$$y_{f_i}(t) = \mathfrak{L}^{-1}\left[\frac{Y(s)}{(s+\lambda)^i}\right]$$

$$y_I(t) = \mathfrak{L}^{-1}\left[\frac{Y(s)}{s}\right]$$

Similarly $\mathbf{u}_{F^n}(t_k - \delta)$ contains the filtered input and the integral of input

$$\mathbf{u}_{F_n}(t_k - \delta) = [u_{f_1}(t_k - \delta) \cdots u_{f_n}(t_k - \delta)\ u_I(t_k - \delta)]^T\, \mathbb{R}^{(n+1)\times 1} \tag{2.20}$$

$\mathbf{F}_n(t_k)$ contains the impulse response of the filters

$$\mathbf{F}_n(t_k) = [f_1(t_k) \cdots f_n(t_k)\ f_I(t_k)]^T \in \mathbb{R}^{(n+1)\times 1} \tag{2.21}$$

$$f_i(t) = \mathfrak{L}^{-1}\left[\frac{1}{(s+\lambda)^i}\right] = \frac{t^{i-1}e^{-\lambda t}}{(i-1)!} \tag{2.22}$$

$$f_I(t) = \mathfrak{L}^{-1}\left[\frac{1}{s}\right] = 1 \tag{2.23}$$

Assuming $a_0 = 1$, eqn(2.19) can be rearranged to give a least-squares equation (LSE) form

$$\begin{aligned} \tilde{\mathbf{\Lambda}}_n \mathbf{y}_{F_n}(t_k) &= -\bar{\mathbf{a}}_n \bar{\mathbf{\Lambda}}_n \mathbf{y}_{F_n}(t_k) + \mathbf{b}_m \mathbf{\Lambda}_m \mathbf{u}_{F_n}(t_k - \delta) \\ &\quad + \mathbf{c}_{n-1}\mathbf{\Lambda}_{n-1}\mathbf{F}_n(t_k) + e_F(t_k) \end{aligned} \tag{2.24}$$

where,
$\bar{\mathbf{\Lambda}}_n$: $\mathbf{\Lambda}_n$ with its last row removed, $\bar{\mathbf{\Lambda}}_n \in \mathbb{R}^{n\times(n+1)}$
$\tilde{\mathbf{\Lambda}}_n$: last row of $\mathbf{\Lambda}_n$, $\tilde{\mathbf{\Lambda}}_n \in \mathbb{R}^{1\times(n+1)}$
$\bar{\mathbf{a}}_n$: $\mathbf{a}_n$ with its last column removed, $\bar{\mathbf{a}}_n \in \mathbb{R}^{1\times n}$

As shown in eqn(2.20), the last element of $\mathbf{u}_{F_n}(t_k - \delta)$ is the integrated delayed input signal. Now δ can be expressed as $\delta = (d + \alpha)\Delta t$, where Δt is the sampling interval, d is an integer number and α is a pure fraction ($0 \le \alpha < 1$). Now if it is assumed that the input signal is piecewise constant, we can write for any sampling instant $t = t_k$,

$$u_I(t_k) = \int_0^{t_k} u(t)dt = \sum_{j=1}^{k-1} u_j \Delta t \tag{2.25}$$

$$\begin{aligned} u_I(t_k - \delta) &= \sum_{j=1}^{k-1} u_j\Delta t - \sum_{j=k-d}^{k-1}(u_j - u_{k-1})\Delta t - (u_{k-d-1} - u_{k-1})\alpha\Delta t - u_{k-1}\delta \\ &= u_I(t_k) - \sum_{j=k-d}^{k-1}(u_j - u_{k-1})\Delta t - (u_{k-d-1} - u_{k-1})\alpha\Delta t - u_{k-1}\delta \end{aligned} \tag{2.26}$$

However, it is not necessary to consider that the input is piecewise constant. A similar derivation for input signals that are not piecewise constant is provided in

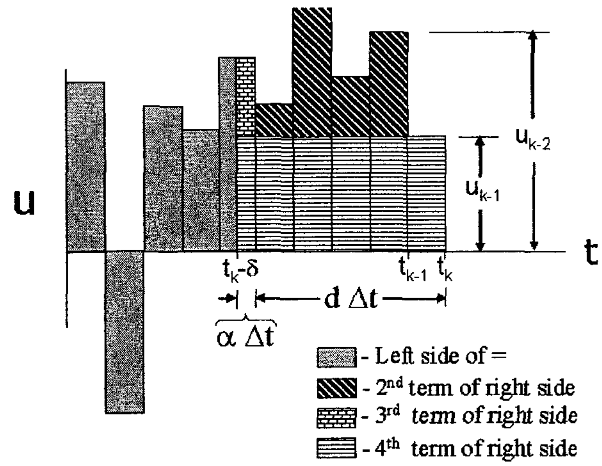


Figure 2.1: *Graphical representation of eqn(2.26).*

chapter 3 (eqn(3.25) and eqn(3.26)). Here, $u(t_i)$ is abbreviated as u_i. Eqn(2.26) can be presented graphically as in figure 2.1. In this figure, the integrated delayed input, $u_I(t_k - \delta)$, is the area under the input signal up to time $t_k - \delta$, while the integrated input signal, $u_I(t_k)$, is the area under the input curve up to time t_k. Also the 2nd, 3rd and 4th term of eqn(2.26) represents the areas as shown by the legends. From the figure, it is seen that by subtracting these 3 areas from $u_I(t_k)$, we get $u_I(t_k - \delta)$. Applying eqn(2.26), $\mathbf{u}_{F_n}(t_k - \delta)$ can be restructured in an extended form

$$\mathbf{u}^+_{F_n}(t_k - \delta) = \begin{bmatrix} u_{f_1}(t_k - \delta) \\ \vdots \\ u_{f_n}(t_k - \delta) \\ u_I(t_k) - \sum_{j=k-d}^{k-1}(u_j - u_{k-1})\Delta t - (u_{k-d-1} - u_{k-1})\alpha\Delta t \\ -u_{k-1}\delta \end{bmatrix} \in \mathbb{R}^{(n+2)\times 1} \tag{2.27}$$

Applying these, we can rewrite eqn(2.24) as

$$\begin{aligned} \tilde{\mathbf{\Lambda}}_n \mathbf{y}_{F_n}(t_k) &= -\bar{\mathbf{a}}_n \bar{\mathbf{\Lambda}}_n \mathbf{y}_{F_n}(t_k) + \mathbf{b}_m \mathbf{\Lambda}_m \bar{\mathbf{u}}^+_{F_n}(t_k - \delta) \\ &\quad + \mathbf{b}_m \underline{\mathbf{\Lambda}}_m \tilde{\mathbf{u}}^+_{F_n}(t_k - \delta) + \mathbf{c}_{n-1}\mathbf{\Lambda}_{n-1}\mathbf{F}_n(t_k) + e_F(t_k) \end{aligned} \tag{2.28}$$

where,

$\underline{\mathbf{\Lambda}}_m$: last column of $\mathbf{\Lambda}_m$

$\bar{\mathbf{u}}^+_{F_n}(t - \delta)$: $\mathbf{u}^+_{F_n}(t - \delta)$ with its last row removed

$\tilde{\mathbf{u}}^+_{F_n}(t-\delta)$: last row of $\mathbf{u}^+_{F_n}(t-\delta)$

Now $\mathbf{b}_m\underline{\mathbf{\Lambda}}_m = b_0\nu$ where $\nu = \mathbf{\Lambda}_m(m+1, n+1)$ as[2] $\mathbf{\Lambda}_m(1:m, n+1) = \mathbf{0}$. Using this, eqn(2.28) can be presented in standard least-squares form

$$\gamma(t_k) = \phi^T_+(t_k)\theta_+ + e_F(t_k) \tag{2.29}$$

where,

$$\gamma(t_k) = \tilde{\mathbf{\Lambda}}_n\mathbf{y}_{F_n}(t_k) \tag{2.30}$$

$$\phi_+(t_k) = \begin{bmatrix} -\bar{\mathbf{\Lambda}}_n\mathbf{y}_{F_n}(t_k) \\ \mathbf{\Lambda}_m\tilde{\mathbf{u}}^+_{F_n}(t_k-\delta) \\ -\nu u_{k-1} \\ \mathbf{\Lambda}_{n-1}\mathbf{F}_n(t_k) \end{bmatrix} \in \mathbb{R}^{(2n+m+2)\times 1} \tag{2.31}$$

$$\theta_+ = [\bar{\mathbf{a}}_n\ \mathbf{b}_m\ b_0\delta\ \mathbf{c}_{n-1}]^T \in \mathbb{R}^{1\times(2n+m+2)} \tag{2.32}$$

Eqn(2.29) can be written for $t_k = t_{d+1}, t_{d+2}\cdots t_N$ and combined to give the equation

$$\mathbf{\Gamma} = \mathbf{\Phi}_+\theta_+ + \mathbf{E}_F \tag{2.33}$$

with

$$\mathbf{\Gamma} = [\gamma(t_{d+1})\ \gamma(t_{d+2})\ \cdots\gamma(t_N)]^T \tag{2.34}$$

$$\mathbf{\Phi}_+ = [\phi_+(t_{d+1})\ \phi_+(t_{d+2})\ \cdots\phi_+(t_N)]^T \tag{2.35}$$

Solution of eqn(2.33) gives the parameter vector θ_+. From θ_+ we can directly get $\bar{\mathbf{a}}_n$ and $\mathbf{b}_m$. δ is obtained as $\delta = \theta_+(n+m+2)/\theta_+(n+m+1)$. To retrieve $\mathbf{y}^{(n-1)}(0)$ from $\mathbf{c}_{n-1}$, eqn(2.6) can be written for $i = 1\cdots n$ to give

$$(\mathbf{c}_{n-1})^T = \mathbf{H}\mathbf{y}^{(n-1)}(0) \tag{2.36}$$

where, $\mathbf{H} = [(\mathbf{h}_1)^T\ (\mathbf{h}_2)^T\cdots(\mathbf{h}_n)^T]^T \in \mathbb{R}^{n\times n}$. Finally

$$\mathbf{y}^{(n-1)}(0) = (\mathbf{H})^{-1}(\mathbf{c}_{n-1})^T \tag{2.37}$$

We see that using this method we can estimate the initial conditions along with other parameters. But the number of parameters to be estimated becomes large which is $2n+m+2$. The elements of $\mathbf{F}_n(t_k)$ as in eqn(2.21) shows that for a stable

[2] For any matrix M, $M(r_1 : r_2, c_1 : c_2)$ refers to the elements from row r_1 to row r_2 and from column c_1 to column c_2

filter parameter, i.e. positive λ, the first n terms decay exponentially and after a certain time become negligible. Only the last term $f_I(t)$ which is a constant remains significant. So after a certain time t_{ss}, we can write

$$\mathbf{c}_{n-1}\mathbf{\Lambda}_{n-1}\mathbf{F}_n(t_k) \simeq \mathbf{c}_{n-1}\underline{\mathbf{\Lambda}}_{n-1} f_I(t_k) \tag{2.38}$$

where, $\underline{\mathbf{\Lambda}}_{n-1}$ is the last column of $\mathbf{\Lambda}_{n-1}$. But $\mathbf{\Lambda}_{n-1}(1:n-1, n+1) = \mathbf{0}$. So we can write

$$\mathbf{c}_{n-1}\underline{\mathbf{\Lambda}}_{n-1} f_I(t_k) = c_0 \mu f_I(t_k) \tag{2.39}$$

with $\mu = \mathbf{\Lambda}_{n-1}(n, n+1)$.

Now eqn(2.28) can be written as

$$\gamma(t_k) = \phi_-^T(t_k)\theta_- + e_F(t_k) \tag{2.40}$$

where,

$$\phi_-(t_k) = \begin{bmatrix} -\bar{\mathbf{\Lambda}}_n \mathbf{y}_{F_n}(t_k) \\ \mathbf{\Lambda}_m \bar{\mathbf{u}}_{F_n}^+(t_k - \delta) \\ -\nu u_{k-1} \\ \mu f_I(t_k) \end{bmatrix} \in \mathbb{R}^{(n+m+3)\times 1} \tag{2.41}$$

$$\theta_- = [\bar{\mathbf{a}}_n \; \mathbf{b}_m \; b_0\delta \; c_0]^T \in \mathbb{R}^{1\times(n+m+3)} \tag{2.42}$$

If we are not interested in estimating the initial conditions but want to estimate other parameters in the presence of initial conditions, eqn(2.40) can be written for $t_k = t_{ss} \cdots t_N$ and combined in the same way as in eqn(2.34) and (2.35) to give

$$\mathbf{\Gamma} = \mathbf{\Phi}_-\theta_- + \mathbf{E}_F \tag{2.43}$$

2.5.4 Parameter estimation algorithm

Note that δ appears both in $\mathbf{\Phi}_-$ and θ_- in eqn(2.43). So, to solve the equation to get θ_-, it is necessary to devise an iterative way. A straightforward approach of successive iteration can be implemented that uses the estimated value of an iteration step as the guess for the next step. The iteration procedure is terminated when δ converges. The iteration steps are described in *Algorithm 2.1.*

The least-squares estimate of θ_- that minimizes the sum of the squared errors is given by

$$\hat{\theta}_-^{LS} = \left[\mathbf{\Phi}_-^T\mathbf{\Phi}_-\right]^{-1}\mathbf{\Phi}_-^T\mathbf{\Gamma} \tag{2.44}$$

But the least-squares solution does not give an unbiased estimate in the presence of general forms of measurement noise such as the colored noise. Even if the measurement noise is assumed to be white with zero-mean, the filtering operation causes coloring of the noise. So, the LS solution is not unbiased even for a white measurement noise and we need a bias elimination scheme. A popular bias elimination procedure is to use the instrumental variable (IV) method. A bootstrap estimation of IV type where the instrumental variable is built from an auxiliary model (Young 1970) is considered here. The instrument matrix $\mathbf{\Psi}_-$ is generated by replacing the y_{f_i} in the $\mathbf{\Phi}_-$ matrix by $\hat{x}_{f_i}$ i.e,

$$\psi_-(t_k) = \begin{bmatrix} -\bar{\mathbf{\Lambda}}_n \hat{\mathbf{x}}_{F_n}(t_k) \\ \mathbf{\Lambda}_m \bar{\mathbf{u}}^+_{F_n}(t_k - \delta) \\ -\nu u_{k-1} \\ \mu f_I(t_k) \end{bmatrix} \in \mathbb{R}^{(n+m+3)\times 1} \tag{2.45}$$

where

$$\hat{x}(t) = \frac{\hat{B}^{LS}(p)}{\hat{A}^{LS}(p)} u(t - \hat{\delta}^{LS}) \tag{2.46}$$

where, $\hat{B}^{LS}(p)$, $\hat{A}^{LS}(p)$ and $\hat{\delta}^{LS}$ are the least squares estimate of the numerator polynomial, the denominator polynomial and the delay. The IV-based bias-eliminated parameters are given by

$$\hat{\theta}_-^{IV} = \left[\mathbf{\Psi}_-^T \mathbf{\Phi}_-\right]^{-1} \mathbf{\Psi}_-^T \mathbf{\Gamma} \tag{2.47}$$

The IV estimate can also be calculated in a recursive or recursive/iterative manner.

Extensive simulation studies show that the iterative procedure converges monotonically except for non-minimum phase processes. However, for non-minimum phase processes it always exhibits monotonic divergence. Based on this, for such processes, we suggest the following *ad hoc* procedure. Details on the results based on which the following procedure is developed is given in *Appendix A*.

Special procedure for NMP processes

The iteration procedure described in the previous section is a fixed point iteration scheme expressed as $\delta = g(\delta) = \theta_-(n+m+2)/\theta_-(n+m+1)$ where, θ_- is given by eqn.(2.44) and eqn(2.47) with $\Phi = \Phi(\delta)$ and $\Psi = \Psi(\delta)$. For minimum phase processes, $g(\delta)$ maps δ in the region of monotonic convergence while for non-minimum phase(NMP) processes, it maps δ in the region of monotonic divergence. Here we use the following result to make the divergent scheme for NMP process converge.

Algorithm 2.1: Iterative procedure for simultaneous estimation of system parameters and the delay.

Step 1: Initialization Choose the filter parameter, λ, and the initial guess of the delay,δ_0.

Step 2: LS step $i = 1$. Construct $\mathbf{\Gamma}$ and $\mathbf{\Phi}_-$ by replacing δ by δ_0 and get the LS solution of θ_-

$$\hat{\theta}_-^{LS} = (\Phi_-^T \Phi_-)^{-1} \Phi_-^T \Gamma \tag{2.48}$$

$\hat{\theta}_-^1 = \hat{\theta}_-^{LS}$. Get $\hat{A}_1(p)$, $\hat{B}_1(p)$ and $\hat{\delta}_1$ from $\hat{\theta}_-^1$.

Step 3: IV step $i = i + 1$. Reconstruct Φ_- with $\delta = \hat{\delta}_{i-1}$. Estimate

$$\hat{x}(t) = \frac{\hat{B}_{i-1}(p)}{\hat{A}_{i-1}(p)} u(t - \hat{\delta}_{i-1}) \tag{2.49}$$

and construct $\mathbf{\Psi}_-$. Get the IV solution of $\hat{\theta}_-^i$

$$\hat{\theta}_-^i = (\mathbf{\Psi}_-^T \mathbf{\Phi}_-)^{-1} \mathbf{\Psi}_-^T \mathbf{\Gamma} \tag{2.50}$$

Obtain $\hat{A}_i(p)$, $\hat{B}_i(p)$ and $\hat{\delta}_i$ from $\hat{\theta}_-^i$ and repeat **step 3** until $\hat{\delta}_i$ converges.

Step 4: Termination When $\hat{\delta}_i$ converges, the corresponding $\hat{\theta}_-^i$ is the final estimate of parameters.

If a fixed point iteration scheme $x = g_1(x)$ diverges monotonically, another scheme $x = x + \frac{1}{r}[x - g_1(x)]$ with $r > 0$, will converge monotonically if $g_1(x)$ is bounded by the region $y = x$ and $y = (r + 1)x + c$ where c is a constant satisfying that y passes through the fixed point.

Simulation results for a large number of process models show that for an appropriate λ and input, for NMP processes $g(\delta)$ maps δ within the region $\delta = \delta$ and $\delta = (r+1)\delta+c$ with $r = 1$. Hence expressing the estimation equation as $\delta = \delta + [\delta - g(\delta)]$ will lead to convergence. So, if the diverging scheme gives

$$\delta_{i+1}^d = g_1(\delta_i) \tag{2.51}$$

To make the scheme converging, we choose

$$\delta^c_{i+1} = \delta_i + [\delta_i - g_1(\delta_i)] = \delta_i + [\delta_i - \delta^d_{i+1}] \tag{2.52}$$

We define $\Delta\delta = \delta_i - \delta^d_{i+1}$ and for successive iteration for a value of δ_i, δ_{i+1} is computed as

$$\delta_{i+1} = \delta_i + \Delta\delta \tag{2.53}$$

The iteration steps otherwise remain the same.

2.5.5 Implementation issues

Choice of δ_0:

As seen in *Algorithm 2.1*, the initiation of the iteration procedure involves selection of δ_0. Through simulations it was found that a choice of even 0 leads to convergence of the iteration procedure. Of course, an initial guess close to the true value saves computation. Hence, we suggest choosing δ_0 based on available process knowledge. In case of unavailability of process knowledge, we suggest choosing $\delta_0 = 0$.

Choice of filter parameters:

In theory, there is no constraint on the choice of the filter parameters except that they should be non-zero real numbers and λ should be positive. But due to the presence of noise, the quality of models depends significantly on the filter parameter λ. On the other hand, β is found to have little or no effect on the parameter estimates as long as it is not very different from λ. But, there is no explicit rule to choose λ. The basic requirement is that the filter should cover the frequency band of the process. As the filtering operation is similar to that of PMF approach, guidelines available in literatures (Bastogne *et al.* 2001, Homssi and Tilti 1991, Roy *et al.* 1991) to select filter parameters for PMF method can be followed.

Based on these guidelines and simulation results, we suggest choosing λ slightly higher than the bandwidth of the system and choosing $\beta = \lambda$. But a very low value of λ is not recommended. In practice, the system bandwidth is unknown. Hence, an iterative procedure can be applied to choose λ. Process information can be used to get an initial idea of the minimal and maximal values of $\lambda \in \{\lambda_{min} \cdots \lambda_{max}\}$. A suitable value of iteration step, λ_{step}, can be chosen based on the range of λ. The mean squared error (MSE) can be used as the criterion to minimize. Finally

$$\lambda_{chosen} = \min_{\lambda} \; MSE(\lambda) \tag{2.54}$$

Choice of t_{ss} :

To satisfy eqn(2.38), it is sufficient to choose t_{ss} such that any linear combination of the first n elements of $\mathbf{F}_n(t_k)$ as in eqn(2.21) becomes negligible. This can be satisfied by choosing t_{ss} such that all the elements themselves are very small. Now, looking at the elements, we see that these are nothing but unit impulse responses of filters of different orders having the same real equal poles. From the characteristics of unit impulse response, we know that the response of the filter having the highest order dies down at the last. Hence it is sufficient that we choose t_{ss} that makes

$$f_n(t_k) = \frac{t^{n-1}e^{-\lambda t}}{(n-1)!} < \epsilon \text{ for } t_k \geq t_{ss} \tag{2.55}$$

where, ϵ is a user specified small number.

2.6 Simulation study

For the simulation study, the inputs are either random binary signals (RBS) or multi-sine signals generated using the *idinput* command in MATLAB with levels $[-1 \;\; 1]$. The number of available samples is 2000. The sampled noise free outputs are corrupted by discrete-time white noise sequences. The noise to signal ratio(NSR) is defined as the ratio of the variance of the noise to that of the noise free signal. Also we set $\beta = \lambda$. Simulink was used to generate the data and numerically simulate the filtered input and output. The process and filters were represented by continuous-time transfer function block. The input signals were passed through a zero order hold (ZOH) block. For plant data, the input may not remain constant over the sampling interval. This will have an impact on the quality of the estimated model. However, if the sampling interval is small, which is typical for continuous-time identification, the error will also be small.

2.6.1 Example 1: Importance of fractional delay estimation

A first order process is used to demonstrate the importance of the estimation of fractional delays (i.e. delays that are not integer multiples of the sampling interval).

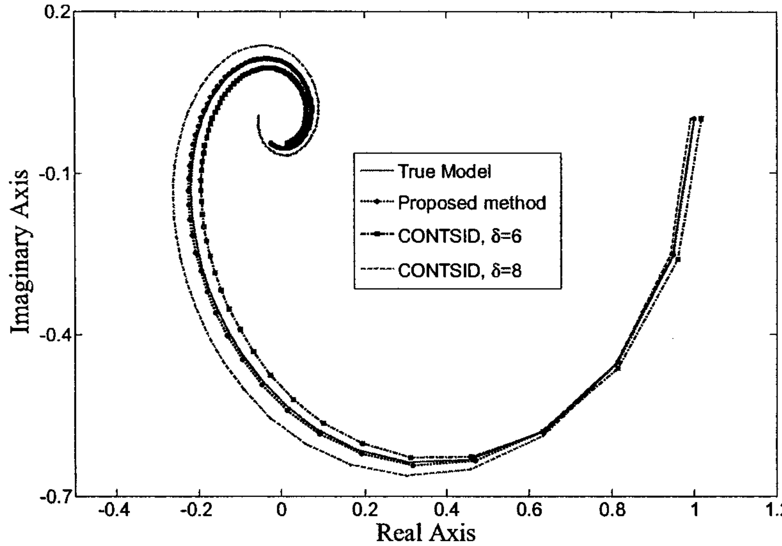


Figure 2.2: *Nyquist plots for example 1.*

The process has the following transfer function

$$G(s) = \frac{1}{20s+1}e^{-6.8s} \tag{2.56}$$

The sampling interval was 2 seconds. So the delay is equivalent to 3.4 sampling intervals. The input was a multi-sine signal with frequency band [0 0.05]. The band contains the lower and upper limits of the passband, expressed in fractions of the Nyquist frequency. The NSR was 10%. A zero initial condition was set for this example. We here compare the results with the models obtained using the PMF method of the CONTSID[3] toolbox. Figure 2.2 shows the Nyquist plots of the true model, the model estimated by the proposed method and the two models obtained from the CONTSID toolbox (for delay of $3\Delta t$ and $4\Delta t$, respectively). The figure clearly shows that neither of the frequency responses of the models estimated using the CONTSID toolbox matches the frequency response of the true model while the model estimated by the proposed method matches with the true process as expected.

[3]CONTSID (CONtinuous Time System IDentification) is a MATLAB based toolbox containing most of the identification methods found in the literature for CT model identification from DT data and is available at http://www.iris.cran.uhp-nancy.fr/contsid/

2.6.2 Example 2: Effect of filter and noise

The same process as in example 1 is used to demonstrate the effect of filter and noise on the quality of estimated models. Figure 2.3 shows the effect of the filter constant

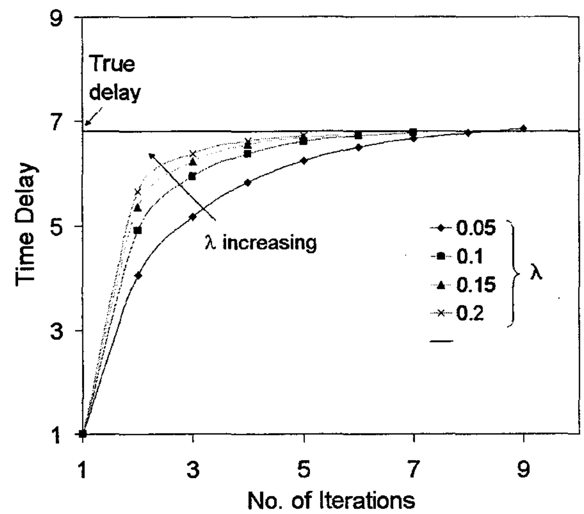
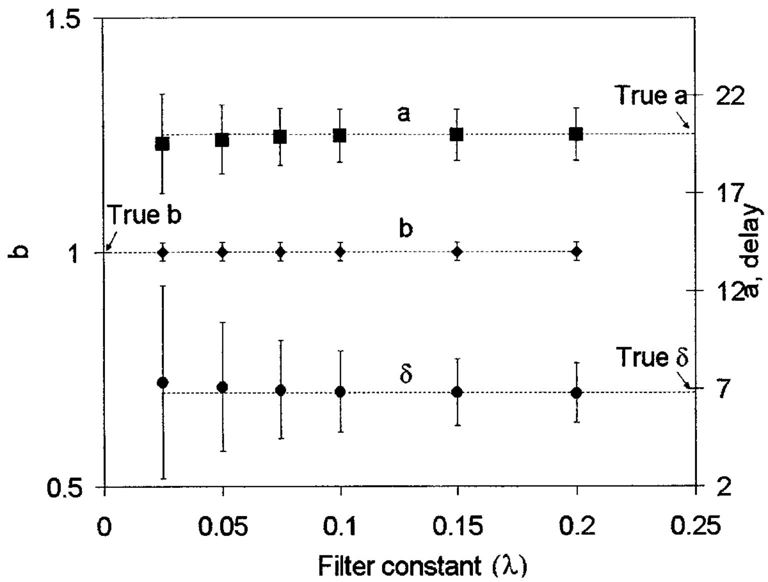
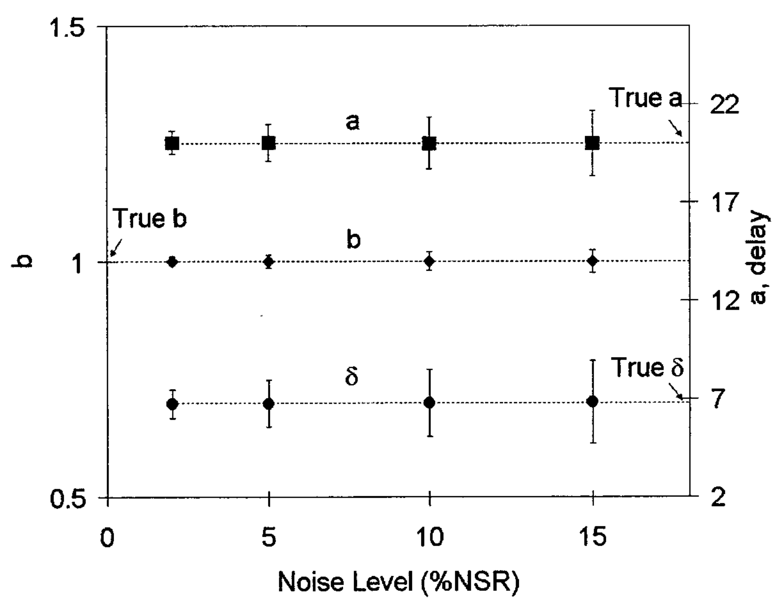


Figure 2.3: *Effect of filter parameter on the rate of convergence.*

(λ) on the convergence rate of the iteration step. This result is for a single noise sequence having $NSR = 10\%$. The legends show the value of λ. We see that as λ increases, the time constant of the filter decreases and faster convergence is achieved. But a higher value of λ means that the passband of the filter is wider with lower noise attenuation which affects parameter estimation. Figure 2.4(a) shows the effect of filter constant on parameter estimates. The results are that of 500 Monte Carlo simulations (MCS) runs. The mean values of 500 estimates are plotted bounded by the estimated $\pm$ values of the sample standard deviation. The figure shows that when the filter bandwidth is smaller than the process bandwidth, the results are poor. Still for a wide range of the filter constants good estimates are obtained. Figure 2.4(b) shows the effect of noise on parameter estimates. These results are also from 500 MCS. For this study λ is set at 0.15 as from figure 2.4(a) we see that the best results are obtained for that value of λ. It is seen that as the noise level increases the estimates get poorer. Nevertheless, for noise as high as $NSR = 15\%$, we get reasonably good estimates.



(a) Filter effect



(b) Noise effect

Figure 2.4: *Effect of filter parameter and noise on parameter estimates.*

2.6.3 Example 3: Second order system modeling

In this example a number of second order processes are considered. The NSR for all cases are 10%. Table 2.1 shows the true and estimated models of different second order processes ranging from slow to fast ones and from underdamped to overdamped. Processes with a zero in the numerator of the transfer function are also considered. For the non-minimum phase process the special procedure described in section 2.5.4 is applied. The parameters shown here are the mean of 500 Monte Carlo simulations. The numbers in the parentheses are the estimated standard deviation of the 500 estimates.

Table 2.1: *True and estimated models for different second order processes.*

True models	Estimated models
$\frac{1.25e^{-0.234s}}{0.25s^2+0.7s+1}$	$\frac{1.25(\pm 0.02)e^{-0.239(\pm 0.042)s}}{0.25(\pm 0.029)s^2+0.697(\pm 0.02)s+1}$
$\frac{2e^{-4.1s}}{100s^2+25s+1}$	$\frac{2(\pm 0.04)e^{-4.13(\pm 0.742)s}}{99.4(\pm 19.7)s^2+25(\pm 0.67)s+1}$
$\frac{(4s+1)e^{-0.615s}}{9s^2+2.4s+1}$	$\frac{(4(\pm 0.54)s+1(\pm 0.07))e^{-0.61(\pm 0.09)s}}{9(\pm 0.76)s^2+2.4(\pm 0.2)s+1}$
$\frac{(-4s+1)e^{-0.615s}}{9s^2+2.4s+1}$	$\frac{(-4(\pm 0.0913)s+1(\pm 0.06))e^{-0.6157(\pm 0.07)s}}{8.99(\pm 0.15)s^2+2.41(\pm 0.15)s+1}$

Also we consider here a fifth order process with no delay modeled by a second order process with delay. The process has the transfer function

$$G(s) = \frac{1}{(s+1)^5} \tag{2.57}$$

with $\mathbf{y}^{n-1}(0) = [0.1\ 0.1\ 0.1\ 0.1\ 0.1]^T$. The t_{ss} was set as $350\Delta t$. A RBS was used as the input for this example.

Figure 2.5 shows the identification results from 500 MCS. In the top figure validation data are shown. The dashed line is the output from the true model and the solid line is that from the estimated model. The parameters of the estimated model is the mean of 500 MCS. The figure at the bottom shows the step responses of the 500 estimated models (black lines) and that of the true fifth order model (white line).

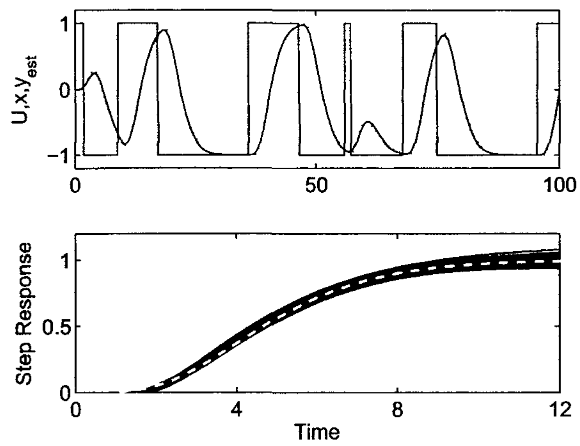


Figure 2.5: *Validation data (top) and step responses (bottom) for example 3.*

2.7 Experimental evaluation

In this section, identification results of a laboratory process are presented. This exercise is carried out using the data set from a dryer (*dryer.mat*) available in the CONTSID toolbox. Details on the process and experiment are obtained from (Garnier 2002) and described below.

The SISO laboratory set-up is a bench-scale hot air-flow device. Air is pulled by a fan into a 30 *cm* tube through a valve and heated by a mesh of resistor wires at the inlet. The output is the voltage delivered by a thermocouple proportional to the air temperature at the outlet of the tube. The input is the voltage over the heating device.

The input signal is a Pseudo Random Binary Signal (PRBS) with maximum length. The sampling period is 100 *ms*. There are two data sets, one for identification and the other for validation, each containing 1905 measurements collected under the same conditions.

A first order model with time delay was estimated for this process. Figure 2.6 shows the validation data set. The simulated output matches the measured one quite well. Here, no *a priori* knowledge of the time delay is used and an initial guess of 0 converged to the final estimate of 0.53 sec.

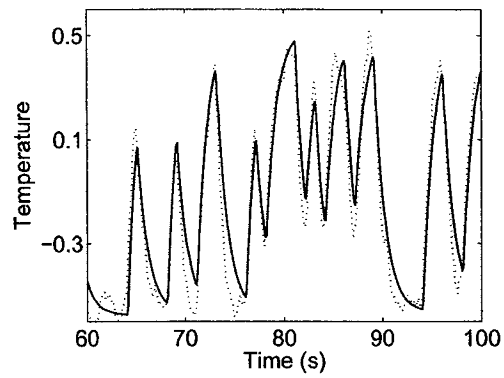


Figure 2.6: *Validation data for dryer. solid line: Simulated, dashed line: Measured.*

2.8 Concluding remarks

In both discrete-time and continuous-time identification, time delay estimation and parameter estimation are often considered two disjoint or separate problems and different approaches are applied for their solution. Among the few exceptions are some optimization based techniques and step response based methods. In this work a new linear filter method is proposed that simultaneously estimates model parameters and the delay. Detailed mathematical derivation has been provided to show how the delay term can be brought in the parameter vector using a filter of a novel structure. Finally an algorithm for the solution of the resulting iterative procedure is provided. The ability to estimate fractional time delay is a unique feature of the proposed method. Through simulations, it is shown that the proposed algorithm is robust in the presence of significant noise and gives satisfactory results over a wide range of values of filter parameter. Finally, the performance of the proposed procedure is demonstrated by experimental application to estimate the delay and model parameters of a laboratory scale process.

Chapter 3

Identification From Non-uniformly Sampled Data[1]

3.1 Overview

A common problem that prevents many of the identification methods to be applicable in real processes is the unavailability of uniformly sampled data. In process industries, the strategy of sampling may be different for different variables resulting in non-uniformly sampled data matrices. A simple form of non-uniformity is sampling at unequal intervals. Multi-rate data is another form of non-uniform data where different variables are sampled at different sampling intervals. For example, from a time and cost consideration, concentrations are less frequently measured than temperatures and pressures. The less frequently sampled variables have sampling intervals as integer multiples of the sampling interval of the most frequently sampled variable. Another form of non-uniformity is data with missing elements where measurements of all variables are available at some time instants, but at others, measurements of only some variables are available. In chemical processes, data can be missing for two basic reasons: failure in the measurement devices and errors in data management. The most common failures are, sensor breakdown, measurement outside the range of the sensor, data acquisition system malfunction, energy blackout, interruption of transmission lines etc. The common errors in data management are wrong format in logged data, crashes in data management software, data storage errors and so on

[1]This chapter is a modified form of the following article:

Ahmed, S., B. Huang and S. L. Shah (2006). Parameter and delay estimation of continuous-time models from irregular output data, *In Proc. ADCHEM 2006, Gramado, Brazil*

(Imtiaz *et al.* 2004). In robust analysis of data, observed values which lie far from the normal trend of the data are considered as outliers and often discarded or treated as missing. Also highly compressed data or unequal length batch data, which may not immediately appear as non-uniform data, can be analyzed within the framework of non-uniform data analysis. An extreme form of irregular data may be asynchronized data for which different variables are sampled at different time instants.

The problem of non-uniform data has been considered in discrete-time identification literature e.g. in (Isaksson 1993) for ARX models and in (Raghavan 2004, Raghavan *et al.* 2005) for state-space models using the expectation maximization (EM) algorithm. Use of the lifting technique for identification from multi-rate data has been reported in a number of articles e.g. in (Li *et al.* 2001, Wang *et al.* 2004*a*). In continuous-time identification literature, methods have been proposed for unevenly sampled data in (Huselstein and Garnier 2002, Larsson and Söderström 2002) where the problem of non-uniform sampling is handled by adopting numerical algorithms suitable for the data type. Methods for frequency domain identification from non-uniformly sampled data have been presented for continuous-time autoregressive (AR) models in (Gillberg and Gustafsson 2005), for autoregressive moving average (ARMA) models in (Gillberg and Ljung 2004) and for output error (OE) models in (Gillberg and Ljung 2006). In fact, continuous-time identification methods can handle the uneven sampling problem by nature provided that appropriate numerical techniques are used. However, the inherent assumption of the numerical methods on the inter-sample behavior of the variables may introduce errors in the parameter estimates. Pintelon and Schoukens (1999) presented a frequency domain identification technique for continuous-time models where missing elements in both of the input and the output signals are considered. The basic idea behind this method is to treat the missing data as unknown parameters of the identification problem.

In this work we consider non-uniformly sampled data with missing elements in the output signals. In general, input variables of an identification exercise are the manipulated variables of the process and are available regularly and at a faster rate. On the other hand some quality variables may be sampled at slower rates or may be missing at some time instants. More often the quality variables are the output for the model to be identified.

3.2 Proposed iterative prediction algorithm

3.2.1 The algorithm

For identification with non-uniformly sampled data, we propose an algorithm based on iterative prediction. However, it is not possible to develop a single algorithm that can deal with every type of data irregularity. We will consider a specific type of non-uniform data where the input is available at all sampling instants but not necessarily in regular intervals while the output is available at some sampling instants and missing at others. This is a more general form of synchronized data. Multi-rate data can be considered as a special form of this non-uniform data. As the initialization of the iterative procedure, a so called input only model is used. A distinguishing feature of these models is that the current output is expressed in terms of only current and previous inputs. So the parameter estimation equation can be formulated only at those time instants when the output is available. The estimated model is then used to predict the missing values to get a complete data set. Next, this complete data set is used to estimate the parameters of the continuous time model using the procedure described in section 3.3. This model is then used to predict the missing outputs. This procedure is carried on iteratively by replacing the prediction from previous model by that from the current one until some convergence criteria are met. The iteration algorithm is presented graphically in figure 3.1 and the different steps of the iteration procedure are detailed below.

3.2.2 Input only modelling

For the purpose of initial prediction, we consider a model that expresses the output in terms of only the input. A number of such input only approaches, both in discrete-time and continuous-time, are available in the literature. Different basis function methods may be used to serve this purpose. In this work, we use one of the orthogonal basis function models, namely a Laguerre polynomial model in continuous time for the initial prediction.

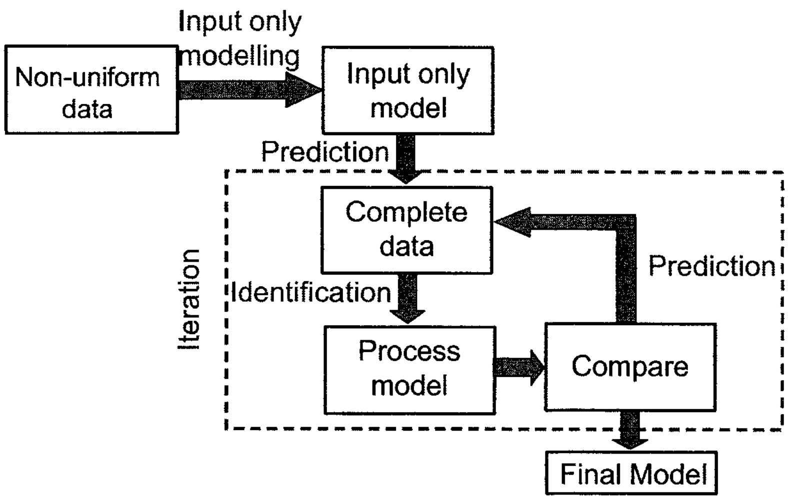


Figure 3.1: *Graphical representation of the iterative prediction algorithm for identification from non-uniformly sampled data.*

Laguerre polynomial model

The use of Laguerre functions in identification goes back to Wiener (1956). In the transform domain, the Laguerre functions are given by (Lee 1932)

$$L_j(s) = \sqrt{2\kappa}\frac{(s-\kappa)^j}{(s+\kappa)^{j+1}} \tag{3.1}$$

where, κ is the parameter of the Laguerre model to be specified by the user. Let $z_j(t)$ be the output of the $j-th$ Laguerre function, with $u(t)$ as its input, i.e.

$$Z_j(s) = L_j(s)U(s) \tag{3.2}$$

where, $Z_j(s)$ and $U(s)$ represent the Laplace transform of $z_j(t)$ and $u(t)$, respectively. The output of a stable plant with input $u(t)$ can be approximated by a truncated $l-th$ order Laguerre polynomial model

$$y(t) = \sum_{j=0}^{l} \alpha_j z_j(t) \tag{3.3}$$

where, $\alpha = [\alpha_0, \alpha_1 \cdots \alpha_l]^T$, is the parameter vector for the Laguerre model. Theories and proofs of the convergence of the Laguerre model can be found in (Makila 1990, Parington 1991, Wang and Cluett 1995).

We denote the time instants when the output is available by t_i^{obs} with $i = 1, 2 \cdots M$, where, M is the number of available output data. Also if the time instants when the output is missing are denoted by t_k^{mis} with $k = 1, 2 \cdots N - M$, with N being the length of the input vector, we can represent the incomplete output vector as

$$Y_{incomplete} = \{Y_{obs} \;\; Y_{mis}\} \tag{3.4}$$

with

$$Y_{obs} = \begin{bmatrix} y(t_1^{obs}) \\ y(t_2^{obs}) \\ \cdots \\ y(t_M^{obs}) \end{bmatrix} \qquad Y_{mis} = \begin{bmatrix} y(t_1^{mis}) \\ y(t_2^{mis}) \\ \cdots \\ y(t_{N-M}^{mis}) \end{bmatrix} \tag{3.5}$$

In the initial prediction stage using the Laguerre polynomial model, the estimation equation (eqn(3.3)) is formulated only at the time instants when the output is available to give

$$y(t_i^{obs}) = \sum_{j=0}^{l} \alpha_j z_j(t_i^{obs}) \tag{3.6}$$

Next eqn(3.6) can be formulated for t_i^{obs} with $i = 1, 2 \cdots M$ to give an equation in least squares form

$$Y_{obs} = \mathbf{Z}_{obs}\alpha \tag{3.7}$$

where,

$$\mathbf{Z}_{obs} = \begin{bmatrix} z_0(t_1^{obs}) & z_1(t_1^{obs}) & \cdots & z_l(t_1^{obs}) \\ z_0(t_2^{obs}) & z_1(t_2^{obs}) & \cdots & z_l(t_2^{obs}) \\ \cdots & \cdots & \cdots & \cdots \\ z_0(t_M^{obs}) & z_1(t_M^{obs}) & \cdots & z_l(t_M^{obs}) \end{bmatrix} \tag{3.8}$$

Finally, the parameter vector can be obtained as

$$\hat{\alpha} = (\mathbf{Z}_{obs}^T \mathbf{Z}_{obs})^{-1} \mathbf{Z}_{obs}^T Y_{obs} \tag{3.9}$$

The missing elements of the output can be predicted using

$$\hat{y}(t_i^{mis}) = \sum_{j=0}^{l} \hat{\alpha}_j z_j(t_i^{mis}) \tag{3.10}$$

The estimated value of the missing elements can then be inserted into the output vector to get a complete data set

$$Y_{complete} = \{Y_{obs} \;\; \hat{Y}_{mis}\} \tag{3.11}$$

This complete data is then used for the identification of the transfer function model of the process.

3.3 Parameter estimation using complete data

As shown in the previous section, using the prediction from the input only model, we get a complete data set with regularly sampled input and output. Now, we can use the linear filter method described in the previous chapter to estimate the model parameters. As stated earlier, we describe the identification method for the filter

$$F(s) = \frac{1}{sA(s)} \tag{3.12}$$

To describe the necessary formulation, let us start with the differential equation model described by eqn(2.2)

$$\mathbf{a}_n \mathbf{y}^{(n)}(t) = \mathbf{b}_m \mathbf{u}^{(m)}(t-\delta) + e(t) \tag{3.13}$$

For the purpose of simplicity in presentation, we will assume that the process input and output are initially at rest. It has been shown in chapter 2 and chapter 4 how one can handle the initial conditions. Based on this assumption of initial steady state, for the causal filter $F(s)$ we can write

$$\mathbf{a}_n \mathbf{s}^n F(s) Y(s) = \mathbf{b}_m \mathbf{s}^m F(s) U(s) e^{-\delta s} + F(s) E(s) \tag{3.14}$$

For the filter in eqn(3.12), we denote $\underline{Y}(s)$ by

$$\underline{Y}(s) = \frac{Y(s)}{A(s)} \tag{3.15}$$

By defining $\underline{U}(s)$ in the same way as $\underline{Y}(s)$ is defined in eqn(3.15), we can express eqn(3.14) as

$$\mathbf{a}_n \mathbf{s}_+^{n-1} \underline{Y}(s) = \mathbf{b}_m \mathbf{s}_+^{m-1} \underline{U}(s) e^{-\delta s} + \xi(s) \tag{3.16}$$

where, the subscript $(\bullet_+)$ means that the $\mathbf{s}^{n-1}$ vector has been augmented by $\frac{1}{s}$, i.e.,

$$\mathbf{s}_+^{n-1} = \left[s^{n-1} \; s^{n-2} \ldots s^0 \; \frac{1}{s} \right] \tag{3.17}$$

Using partial fraction expansion, the transfer function of the filter, $1/sA(s)$, can be expressed as

$$\frac{1}{sA(s)} = \frac{C(s)}{A(s)} + \frac{1}{s} \tag{3.18}$$

where, $C(s) = -(a_n s^{n-1} + a_{n-1} s^{n-2} + \cdots + a_1)$. Eqn(3.18) can be used to represent the filtered input

$$\begin{aligned} U_F(s) &= \frac{C(s)}{A(s)} U(s) + \frac{1}{s} U(s) \\ &= C(s) \underline{U}(s) + U_I(s) \end{aligned} \tag{3.19}$$

Applying eqn(3.19), we can restructure eqn(3.16) to give a standard least squares form

$$\begin{aligned}\underline{Y}_I(s) &= -\bar{\mathbf{a}}_n\mathbf{s}^{n-1}\underline{Y}(s) + \bar{\mathbf{b}}_m\mathbf{s}^{m-1}\underline{U}(s)e^{-\delta s} \\ &\quad + b_0\left[C(s)\underline{U}(s) + U_I(s)\right]e^{-\delta s} + \xi(s) \end{aligned} \qquad (3.20)$$

where,
$\bar{\mathbf{a}}_n$: $\mathbf{a}_n$ with its last column removed, $\bar{\mathbf{a}}_n \in \mathbb{R}^{1\times n}$
$\bar{\mathbf{b}}_m$: $\mathbf{b}_m$ with its last column removed, $\bar{\mathbf{b}}_n \in \mathbb{R}^{1\times m}$

Taking inverse Laplace Transform, we get the equivalent time domain expression for any sampling instant $t = t_k$

$$\begin{aligned}\underline{y}_I(t_k) &= -\bar{\mathbf{a}}_n\underline{\mathbf{y}}^{(n-1)}(t_k) + \bar{\mathbf{b}}_m\underline{\mathbf{u}}^{(m-1)}(t_k-\delta) \\ &\quad + b_0\left[\underline{u}_c(t_k-\delta) + u_I(t_k-\delta)\right] + \zeta(t_k) \end{aligned} \qquad (3.21)$$

with

$$\underline{y}_I(t_k) = \mathfrak{L}^{-1}\left[\frac{\underline{Y}(s)}{s}\right] \qquad (3.22)$$

$$\underline{u}_c(t_k-\delta) = \mathfrak{L}^{-1}\left[C(s)\underline{U}(s)e^{-\delta s}\right] \qquad (3.23)$$

$$u_I(t_k-\delta) = \mathfrak{L}^{-1}\left[\frac{1}{s}U(s)e^{-\delta s}\right] \qquad (3.24)$$

The integral of the input and of the delayed input for any time $t = t_k$ is given by

$$u_I(t_k) = \int_0^{t_k} u(t)dt \qquad (3.25)$$

$$u_I(t_k-\delta) = \int_0^{t_k} u(t)dt - \int_{t_k-\delta}^{t_k}[u(t)-u(t_k)]dt - u(t_k)\delta \qquad (3.26)$$

Eqn(3.26) can be presented graphically as in figure 3.2. Here, in this figure, the integrated delayed input, $u_I(t_k-\delta)$, is the area under the input signal up to time $t_k - \delta$, while the integrated input signal,$u_I(t_k)$, is the area under the input curve up to time t_k. Also the 2nd and 3rd term of eqn(3.26) represent the areas as shown by the legends in figure 3.2. From the figure, it is seen that by subtracting these 2 areas from $u_I(t_k)$, we get $u_I(t_k-\delta)$.

Applying eqn(3.26) in eqn(3.21) and rearranging it to give a standard least squares form we get

$$\begin{aligned}\underline{y}_I(t_k) &= -\bar{\mathbf{a}}_n\mathbf{y}^{(n-1)}(t_k) + \mathbf{b}_m\underline{\mathbf{u}}_+^{(m-1)}(t_k-\delta) \\ &\quad + b_0u(t_k)\delta + \zeta(t_k) \end{aligned} \qquad (3.27)$$

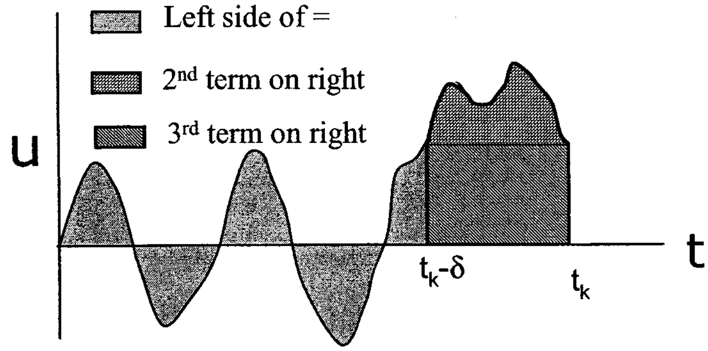


Figure 3.2: *Graphical representation of eqn(3.26) Note that the input is not piecewise constant.*

where,

$$\underline{\mathbf{u}}_+^{(m-1)}(t_k-\delta) = \begin{bmatrix} \underline{u}^{(m-1)}(t_k-\delta) \\ \cdots \\ \underline{u}(t_k-\delta) \\ u_+(t_k-\delta) \end{bmatrix} \tag{3.28}$$

$$u_+(t_k-\delta) = \underline{u}_c(t_k-\delta) + u_I(t_k) - \int_{t_k-\delta}^{t_k} [u(t)-u(t_k)]dt \tag{3.29}$$

Or equivalently

$$\gamma(t_k) = \phi^T(t_k)\theta + \zeta(t_k) \tag{3.30}$$

where,

$$\gamma(t_k) = \underline{y}_I(t_k) \tag{3.31}$$

$$\phi(t_k) = \begin{bmatrix} -\underline{\mathbf{y}}^{(n-1)}(t_k) \\ \underline{\mathbf{u}}_+^{(m-1)}(t_k-\delta) \\ u(t_k) \end{bmatrix} \tag{3.32}$$

$$\theta = [\bar{\mathbf{a}}_n \ \mathbf{b}^m \ b_0\delta] \tag{3.33}$$

Eqn(3.30) can be written for $t_k = t_{d+1}, t_{d+2} \cdots t_N$ and combined to give

$$\mathbf{\Gamma} = \mathbf{\Phi}\theta + \zeta \tag{3.34}$$

3.3.1 Parameter estimation

Solution of eqn(3.34) gives the parameter vector θ. From θ one can directly get $\bar{\mathbf{a}}_n$ and $\mathbf{b}_m$. δ is obtained as $\delta = \theta(n+m+2)/\theta(n+m+1)$. But in estimating θ by

Algorithm 3.1: Linear filter algorithm for simultaneous estimation of the delay and model parameters from complete data.

Step 1-Initialization: Choose an initial estimate $A_0(s)$ and δ_0.

Step 2-LS step: Construct Γ and Φ by replacing $A(s)$ and δ by $A_0(s)$ and δ_0 and get the LS solution of θ

$$\theta^{LS} = (\Phi^T\Phi)^{-1}\Phi^T\Gamma \tag{3.35}$$

$\theta^1 = \theta^{LS}$. Get $A_1(s)$, $B_1(s)$ and δ^1 from θ^1. Set $i = 1$.

Step 3-IV step: $i = i + 1$. Construct Γ, Φ and Ψ by replacing $A(s)$, $B(s)$ and δ by $A_{i-1}(s)$,$B_{i-1}(s)$ and δ_{i-1} and get the IV solution of θ

$$\theta^i = (\Psi^T\Phi)^{-1}\Psi^T\Gamma \tag{3.36}$$

Obtain $A_i(s)$,$B_i(s)$ and δ_i from θ^i and repeat **step 3** until A_i and δ_i converge.

Step 4-Termination: When A_i and δ_i converge, the corresponding θ^i is the final estimate of parameters.

---

solving eqn(3.34), there are two problems. First, we need to know $A(s)$ and δ, which are unknowns. This obvious problem can be solved by applying an iterative procedure that adaptively adjusts initial estimates of $A(s)$ and δ until they converge. Second, The least squares estimate of θ that minimizes the sum of the squared errors is given by

$$\hat{\theta}^{LS} = \left[\mathbf{\Phi}^T\mathbf{\Phi}\right]^{-1}\mathbf{\Phi}^T\mathbf{\Gamma} \tag{3.37}$$

As discussed in chapter 2, the LS solution is not unbiased even for a white measurement noise and we need a bias elimination scheme. Here we will use the same instrumental variable (IV) method as used in chapter 2 where the instrumental variable is built from an auxiliary model (Young 1970). The instrument vector is given by

$$\psi(t_k) = \begin{bmatrix} -\hat{\underline{\mathbf{x}}}^{(n-1)}(t_k) \\ \underline{\mathbf{u}}_+^{(m-1)}(t_k - \delta) \\ u(t_k) \end{bmatrix} \tag{3.38}$$

where

$$\hat{x}(t) = \hat{G}(p, \hat{\theta}_{-}^{LS})u(t-\delta) \tag{3.39}$$

and $\hat{G}(p, \hat{\theta}_{-}^{LS})$ is the process model estimated from least squares solution. The IV-based bias-eliminated parameters are given by

$$\hat{\theta}^{IV} = \left[\mathbf{\Psi}^T \mathbf{\Phi}\right]^{-1} \mathbf{\Psi}^T \mathbf{\Gamma} \tag{3.40}$$

The iterative identification algorithm for simultaneous estimation of the delay and other parameters from a complete data set is summarized in *Algorithm 3.1.* The iterative prediction algorithm for identification from irregularly sampled output is summarized in *Algorithm 3.2.* The initial prediction step of the iterative algorithm

---

Algorithm 3.2: Iterative prediction algorithm for parameter estimation from irregularly sampled output.

Step 1-Initial Prediction: Using only the observed output, estimate the parameters of the input only model using eqn(3.9). Predict the missing element of the output using eqn(3.10). Use these predicted values, $\hat{Y}_{mis}^{0}$ to replace Y_{mis} in eqn(3.11). $i = 0$.

Step 2-Iterative Prediction: $i = i + 1$. Estimate the continuous time model parameters using the complete data set $Y_{complete} = \{Y_{obs} \quad \hat{Y}_{mis}^{i-1}\}$ and applying Algorithm 3.1. Use the estimated model, θ^i, to get the $i-th$ prediction of the missing values, $\hat{Y}_{mis}^{i}$. Replace $\hat{Y}_{mis}^{i-1}$ by $\hat{Y}_{mis}^{i}$

Step 3-Comparison: Compare MSE_{obs}^{i} with MSE_{obs}^{i-1}. If their is significant improvement, go back to step 2 and repeat the iteration.

Step 4-Termination: When MSE_{obs}^{i} converges, the corresponding θ^i is the final estimate of parameters.

---

involves choice of the parameters of the Laguerre polynomial model, namely κ and l. Generally, we choose κ on the basis of the knowledge of process cut-off frequency. A value slightly higher than the cut-off frequency is chosen. The order of the polynomial model, l, is chosen as few order higher than the order of the transfer function model.

3.3.2 Criterion of convergence

The proposed iterative procedure is based on the idea of iterative prediction. Consequently, a natural option for criterion of convergence is the prediction error. As the output has missing elements, we can define the mean squared error at $i-th$ stage of iteration based on the observed output and their predicted values

$$MSE_{obs}^{i} = \frac{1}{M}\sum_{k=1}^{M}\left[y(t_k^{obs}) - \hat{y}_i(t_k^{obs})\right]^2 \tag{3.41}$$

where, $\hat{y}_i$ is the prediction of the model obtained in the $i-th$ stage of iteration. Convergence of this MSE criterion is equivalent to the convergence of the model prediction and the model parameters.

3.4 Simulation results

To demonstrate the applicability of the proposed methods, we consider here a second order process having the following transfer function

$$G(s) = \frac{-4s+1}{9s^2+2.4s+1}e^{-0.615s} \tag{3.42}$$

A complete data set of 2000 samples with a uniform sampling interval of 30 milliseconds (ms) is generated using a random binary signal (RBS) as the input. Discrete time white noise is added to obtain a noisy output signal. The signal to noise ratio (NSR) is 10%. Table 3.1 summarizes the parameter estimation results from 100 Monte Carlo simulations (MCS) when the entire data set is used for identification.

To test the performance of the algorithm proposed for irregular data, we generate three sets of irregular data that differ in terms of their amount of missing data. For case (i) every 3rd sample is taken out to generate a data set for 33% missing data (ii) every 2^{nd} for 50% missing and (iii) every 2^{nd} and 3^{rd} for 67%. The model estimated using the iterative algorithm is compared with the model estimated using only the available data i.e. data at the time instants when both input and output are available. To compare different models with a single index we define a total error criterion that is a combined measure of bias and variance. We denote it by E_{total} where

$$E_{total} = \frac{1}{N_\theta}\sum_{i=1}^{N_\theta}\frac{(\hat{\theta}_i-\theta_i)^2+\text{var}(\hat{\theta}_i)}{\theta_i^2} \tag{3.43}$$

θ_i is the true values of the i^{th} parameter and $\hat{\theta}_i$ is its estimated value. N_θ is the number of parameters. Figure 3.3 shows the total error for the results from 100

Table 3.1: *Estimation results using the entire data set.*

		Estimated parameters	
Parameter	True value	Mean	Variance
a_2	9.00	9.0068	0.0387
a_1	2.40	2.4309	0.0465
b_1	-4.00	-4.0201	0.0570
b_0	1.00	1.0109	0.0068
δ	0.615	0.6302	0.0253

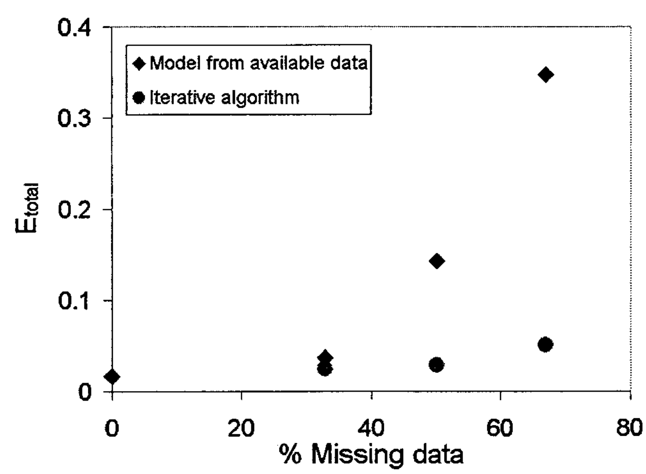


Figure 3.3: *Improvement of model quality using the iterative algorithm for the simulation example.*

MCS runs. The estimated values of the parameters are the means of 100 estimates. The error corresponding to 0% missing data refers to the model estimated using the entire data set and can be taken as the benchmark. When 33% of the data are missing, the model estimated using only the available data has error comparable with the benchmark value and the iterative algorithm has little room to improve. This suggests that the available data are enough to give a good model. Consequently the error level of the model estimated using the iterative algorithm remains almost the same. However, when more data are missing the error corresponding to the model estimated using the available data is much higher than the benchmark value and the iterative algorithm reduces the error to a level comparable with the benchmark.

3.5 Experimental evaluation

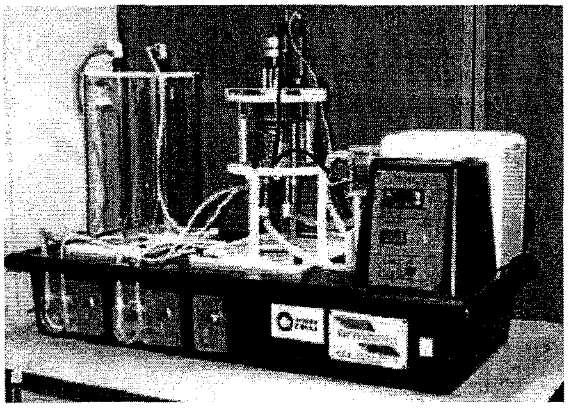


Figure 3.4: *Photograph of the mixing process.*

The iterative prediction algorithm is evaluated using an experimental data set from a mixing process. The set-up consists of a continuous stirred tank used as a mixing chamber having two input streams fed from two feed tanks. Salt solution and pure water run from the feed tanks and mixed together in the mixing chamber. A constant volume and a constant temperature of the solution in the mixing tank are maintained. Also the total inlet flow is kept constant. The input to the process is the flow rate of the salt solution as fraction of total inlet flow. The output is the concentration of salt in the mixing tank. We assume a uniform concentration of salt throughout the solution in the tank. The concentration is measured in terms of the electrical conductivity of the solution. A photograph of the set-up is shown in figure 3.4.

The input signal is a random binary signal. The sampling period is 20 seconds. A total of 955 data points are used for this study. To study the effect of % data missing and evaluate the performance of the iterative prediction algorithm, missing data were chosen on a random basis and the algorithm was applied. The study is carried out for 30%, 50% and 70% missing data. To generate a certain data set, say with 30% of its elements missing, 30% of the available output data are taken out on a random basis. The identification algorithm is then applied with the remaining 70% data points. The same procedure is applied 100 times with a different data set chosen each time containing only 70% of the total data. Finally we get 100 estimates of the parameters. The total error is then calculated from the estimated mean and variance of the 100 estimates. To calculate the bias error, the model estimated using the entire data set is taken as the nominal or true value. Figure 3.5 shows the performance of the proposed iterative algorithm for the mixing process data. While the error levels for models estimated only from the available data points are high, the iterative algorithms gives final estimates of the parameters with a much lower levels of error.

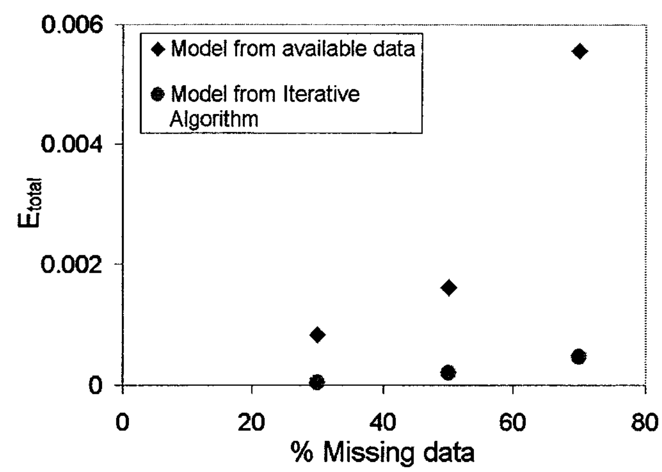


Figure 3.5: *Improvement of model quality using the iterative algorithm for the mixing process.*

3.6 Concluding remarks

Identification from non-uniform data has been considered in discrete-time identification but mainly for multi-rate data. In continuous-time identification, it is assumed that the methods are capable of dealing with non-uniform data provided that appropriate numerical techniques are used. However, the inherent assumption of the numerical methods on the inter-sample behavior of the signals that results in certain arbitrary interpolation, may introduce errors in the estimation of continuous-time parameters. We present a simple algorithm to deal with non-uniformly sampled output data. It has been demonstrated using simulated and experimental data that the quality of the model estimated using the proposed model based prediction algorithm is much better than the quality of the model estimated using only the available output data.

Chapter 4

Identification from Step Response[1]

4.1 Introduction

Step response based methods are most commonly used for system identification, especially in the process industries. The most comprehensive survey of application of identification and parameter estimation techniques to physical and chemical processes was conducted by Gustavsson (1973). A number of identification applications e.g. on extraction process, distillation unit, blast furnace, paper-making process, digesting unit, cement kiln etc. are reported in that survey where step input excitation was used and it was concluded that the step was found to be the most popular input used for identification in physical and chemical processes.

The theoretical development of step response based identification techniques also has a long history. The first method to estimate the parameters of a first order plus time delay (FOPTD) model from step response was proposed by Küpfmüller (1928) who indeed first introduced the idea of step response. This graphical technique, described by Oldenbourg and Sartorius (1948) and later by Rake (1980) and Unbehauen and Rao (1987), involves drawing a tangent to the inflection point of the response curve and formed the basis for a number of similar methods both for first and second order models. Strejc (1959) proposed an improvement of Küpfmüller's method in which the parameters are estimated on the basis of two points suitably chosen on either side of the flexion point. Oldenbourg and Sartorius (1948) describes another method that estimates two non-identical time constants of a second order model from a measure of lengths of different segments of the flexion tangent. There are a number of methods that measure the time required for the output to reach certain percent of its final

[1]An article on this topic has been submitted for publication:

Ahmed, S., B. Huang and S. L. Shah (2006). Novel Identification Method from Step Response, *Submitted to Control Engineering Practice* steady state value and use nomographs to calculate the parameters of different order models. Tahl-Larsen (1956) described a procedure that from the time required to reach 10, 40 and 80 percent of the final value estimates the parameters of a third order model with non-identical poles using a set of graphs. There are also a number of graphical methods available that estimate the parameters of models of any order having real poles and of second order having imaginary poles. For details of such methods readers are referred to (Rake 1980, Seborg *et al.* 1989, Unbehauen and Rao 1987). Sundaresan *et al.* (1978) outlined the limitations of the graphical methods and presented an algorithm that estimates the time delay and the time constants of both under-damped and oscillatory second order processes from the first moment of the normalized step response curve, the slope of the tangent at the inflection point and the time corresponding to the point of intersection between the tangent and the final value of the response curve. Some other developments on the graphical method have been reported in (Huang and Clement 1982, Huang and Huang 1993, Rangaiah and Krishnaswamy 1994).

A group of methods that involves estimation of the area under the response curve, has also been the subject of extensive research. Such methods, often termed as the area methods, have been reported in (Bi *et al.* 1999, Hwang and Lai 2004, Rake 1980, Wang and Zhang 2001*b*). The method by Wang and Zhang (2001*b*) estimates the model parameters and delay simultaneously and is robust in the presence of noise; however, it may give multiple estimates of the delay and is not applicable when the output is initially at an unsteady state. The method by Hwang and Lai (2004) is based on the pulse response; however, it uses data corresponding to one step of the pulse at a time and can estimate the model parameters and delay when the process output is initially not at steady state. This method may also give multiple delays.

Another class of methods, namely the method of moments, has also emerged as an efficient technique for parameter estimation. Use of different order moments for parameter estimation has been reported in (Ba Hli 1954). This method has also been described in (Unbehauen and Rao 1987). The characteristic area method (Nishikawa *et al.* 1990), often termed simply as an area method (Åström and Hägglund 1995), is indeed a variant of the method of moments. An improved method of moment has been proposed in (Ingimundarson and Hagglund 2000) which is also reported in (Ingimundarson 2003). The method of moments has been detailed in (Åström and Hägglund 1995). Identification from step response has also been considered using the Laguerre network in (Wang and Cluett 1995) and combined with the state variable filter method in (Wang *et al.* 2004*b*). A recursive least square algorithm is presented in (Tuch *et al.* 1994) to estimate the time delay of a continuous-time model.

For estimation of the open loop process parameters from a closed loop test applying a step change in the set-point, Yuwana and Seborg (1982) proposed a method for FOPTD model under proportional only controller where Padé approximation is used for the delay term. Jutan and RodriguezII (1984) proposed some modification of the method including the approximation of the delay term by a functional form. Refinement of the method has also been proposed in (Lee 1989, Chen 1989) and a comparison of performance of these methods was reported in (Taiwo 1993). The method was extended for SOPTD systems by Lee *et al.* (1990). Closed-loop identification using the method of moments has been reported in (Nishikawa *et al.* 1990). Viswanathan and Rangaiah (2000) proposed an optimization technique and Coelho and Barros (2003) proposed an integral equation approach that estimates the open loop model parameters using the estimated controller output signal. The method by Yuwana and Seborg (1982) was extended for unstable processes by Kavdia and Chidambaram (1996) but only for P controllers. For unstable processes with a PID controller, a method is proposed by Ananth and Chidambaram (1999) that uses the coordinates of the peaks of the underdamped closed loop response curve to estimate the parameters.

Identification of real processes is often not quite straight-forward (Gustavsson 1973). A number of problems are involved. In this work the main concern is about two specific issues related to the form of data. To formulate the problem some industrial data are presented in figure 4.1. For proprietary reasons the processes are not described. However the process descriptions are not necessary to understand the objectives of this discussion.

In general, identification methods are developed to deal with variables in deviation form while data from industries are not available in that form. To obtain data in deviation form, the initial steady state values are subtracted from the raw values. However, the initial steady state values are often not known correctly due to two reasons: (i) presence of noise and (ii) often step input is introduced before the system is at the desired steady state. The data presented in figures 4.1(a) and 4.1(b) show that indeed the outputs were at some steady state values before the step inputs were applied. However, due to the presence of noise, it is not easy to determine the steady state value exactly. Figures 4.1(c) and 4.1(d) show other situations where the step

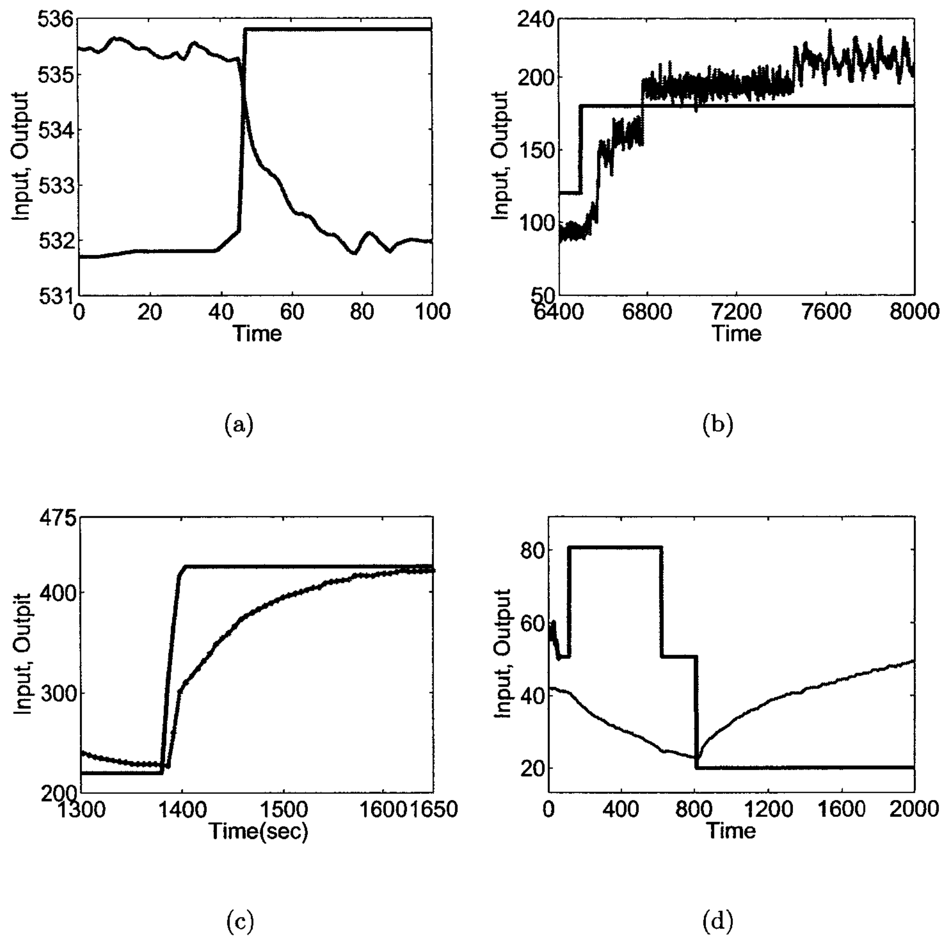


Figure 4.1: *Step responses of different industrial processes.*

inputs were applied before the outputs had reached steady state. For such cases, the initial steady state values are unknown and consequently we cannot convert the data into deviation form.

Another important issue is whether a method is applicable in the presence of initial conditions. A common approach is to formulate the estimation equation at the time instants after the effect of initial conditions becomes negligible. Certainly this is not applicable for step responses. To the best of knowledge of the authors there is no step response based method available in the literature that can handle non-zero initial conditions. In addition, if the input is applied before the system reaches the desired steady state, it is not possible to get the data in deviation form.

In this work we present a new approach for identification from step response that does not need the data in deviation form. This allows us to use industrial data without the required preprocessing. Moreover, the proposed methods can estimate the initial conditions along with the process parameters. So it is not necessary to bring the process to a steady state condition before the step input is applied.

In some cases, such as for unstable processes, it may not be possible to perform open loop tests. For such situations, we consider an identification method that formulates the estimation equation in terms of the open loop model parameters using closed loop step response and setpoint data. Solution of the estimation equation directly gives the open loop model parameters.

4.2 Identification using raw data

4.2.1 Deviation vs. raw form

First, let us see how data in deviation form are obtained from the raw data. Here the subscript ($\bullet_r$) denotes the corresponding variable in raw form and variables without the subscript are in deviation form. These two quantities are related to as follows.

$$y(t_k) = y_r(t_k) - y_{ss} \tag{4.1}$$

where, y_{ss} is the steady state value of the output corresponding to the steady state value of the input before the step is applied. Figure 4.2 describes graphically the raw and deviation form of the variables. To get the variable in deviation form we need to know the value y_{ss} which, as mentioned earlier, is sometimes difficult to measure or may be unknown. However, if we consider y_{ss} as unknown we can simply write

$$y(t_k) = y_r(t_k) - q \tag{4.2}$$

where, q is the initial unknown steady state value of the output. Taking Laplace transform on both sides, we get

$$Y(s) = Y_r(s) - \frac{q}{s} \tag{4.3}$$

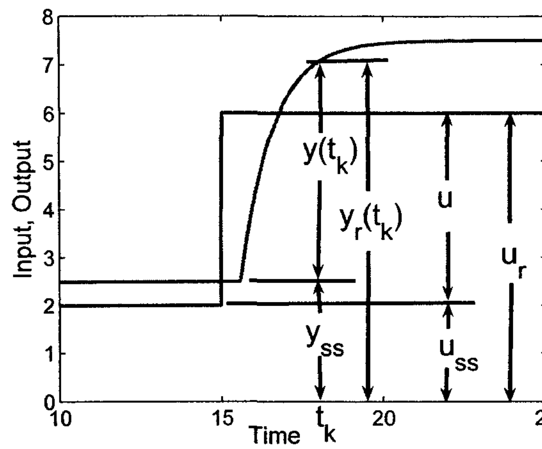


Figure 4.2: *Variables in deviation and raw form.*

4.2.2 Open loop identification

A new identification method is proposed that uses the data in their raw form. The basic idea is to consider the initial steady state of the output as another unknown parameter in the estimation equation. Also the method estimates the initial conditions along with the model parameters. The necessary equations are first derived in terms of the deviation variables. Later using eqn(4.3) the estimation equation will be presented in terms of the raw form of the variables. To describe the necessary mathematical formulation, let us consider a linear single input single output (SISO) system with time delay as described by eqn(2.2)

$$\mathbf{a}_n \mathbf{y}^{(n)}(t) = \mathbf{b}_m \mathbf{u}^{(m)}(t-\delta) + e(t) \tag{4.4}$$

Taking Laplace transformation on both sides of eqn(4.4), we can write

$$\mathbf{a}_n \mathbf{s}^n Y(s) = \mathbf{b}_m \mathbf{s}^m U(s) e^{-\delta s} + \mathbf{c}_{n-1} \mathbf{s}^{n-1} + E(s) \tag{4.5}$$

The notations have been defined in chapter 2. Next, we will devise a linear filter method for the estimation of the parameters. As discussed in chapter 2, filters having different structures can be used for the estimation of the system parameters and the delay. Here, we use the filter described in chapter 3 having the transfer function

$$\frac{1}{sA(s)} \tag{4.6}$$

where, $A(s)(=\mathbf{a}_n \mathbf{s}^n)$ is the denominator of the process transfer function. Now, if we denote $P(s) = \frac{1}{sA(s)}$ and apply the filtering operation on both sides of eqn(4.5) we end up with the formulation

$$\mathbf{a}_n\mathbf{s}^n P(s)Y(s) = \mathbf{b}_m\mathbf{s}^m P(s)U(s)e^{-\delta s} + \mathbf{c}_{n-1}\mathbf{s}^{n-1}P(s) + P(s)E(s) \quad (4.7)$$

Using partial fraction expansion, the transfer function of the filter, $\frac{1}{sA(s)}$, can be expressed as

$$\frac{1}{sA(s)} = \frac{C(s)}{A(s)} + \frac{1}{s} \quad (4.8)$$

where, $C(s) = -(a_n s^{n-1} + a_{n-1}s^{n-2} + \cdots + a_1)$. Using the notations $\underline{Y}(s) = \frac{Y(s)}{A(s)}$, $\underline{Y}^I(s) = \frac{\underline{Y}(s)}{s}$ and similar notations for $U(s)$ and then rearranging the estimation equation to give a standard least-squares form we get the expression

$$\begin{aligned}\underline{Y}^I(s) &= -\bar{\mathbf{a}}_n\mathbf{s}^{n-1}\underline{Y}(s) + \bar{\mathbf{b}}_m\mathbf{s}^{m-1}\underline{U}(s)e^{-\delta s} \\ &\quad + b_0\left[C(s)\underline{U}(s) + U^I(s)\right]e^{-\delta s} + \mathbf{c}_{n-1}\mathbf{s}^{n-1}P(s) + \xi(s) \quad (4.9)\end{aligned}$$

where,
$\bar{\mathbf{a}}_n$: $\mathbf{a}_n$ with its last column removed, $\bar{\mathbf{a}}_n \in \mathbb{R}^{1\times n}$
$\bar{\mathbf{b}}_m$: $\mathbf{b}_m$ with its last column removed,$\bar{\mathbf{b}}_m \in \mathbb{R}^{1\times m}$
Now using eqn(4.3) we can write the above equation in terms of the raw form of the output y

$$\begin{aligned}\underline{Y_r}^I(s) - qP^I(s) &= -\bar{\mathbf{a}}_n\mathbf{s}^{n-1}\underline{Y_r}(s) + \bar{\mathbf{a}}_n\mathbf{s}^{n-1}qP(s) + \bar{\mathbf{b}}_m\mathbf{s}^{m-1}\underline{U}(s)e^{-\delta s} \\ &+ b_0\left[C(s)\underline{U}(s) + U^I(s)\right]e^{-\delta s} + \mathbf{c}_{n-1}\mathbf{s}^{n-1}P(s) + \xi(s) \quad (4.10)\end{aligned}$$

For a step input, if the step size is denoted as h, i.e. $u(t) = u_r(t) - u_{ss} = h$, we have

$$U(S) = \frac{h}{s} \quad (4.11)$$

$$\underline{U}(s) = \frac{U(s)}{A(s)} = \frac{h}{sA(s)} = hP(s) \quad (4.12)$$

Using eqn(4.11) and (4.12) and rearranging eqn(4.10), we get an estimation equation in the Laplace domain

$$\begin{aligned}\underline{Y_r}^I(s) &= -\bar{\mathbf{a}}_n\mathbf{s}^{n-1}\underline{Y_r}(s) + h\bar{\mathbf{b}}_m\mathbf{s}^{m-1}P(s)e^{-\delta s} + b_0\left[hC(s)P(s) + \frac{h}{s^2}\right]e^{-\delta s} \\ &\quad + \left[\mathbf{c}_{n-1} + \bar{\mathbf{a}}_n q\right]\mathbf{s}^{n-1}P(s) + qP^I(s) + \xi(s) \quad (4.13)\end{aligned}$$

Inverse Laplace transform gives the equation in time domain for any time $t = t_k$

$$\begin{aligned}\underline{y_r}^I(t_k) &= -\bar{\mathbf{a}}_n\underline{\mathbf{y}_r}^{(n-1)}(t_k) + h\bar{\mathbf{b}}_m\mathbf{P}^{m-1}(t_k-\delta) + b_0\left[hP_c(t_k-\delta) + h[t_k-\delta]\right] \\ &\quad + \left[\mathbf{c}_{n-1} + \bar{\mathbf{a}}_n q\right]\mathbf{P}^{n-1}(t_k) + qP^I(t_k) + \zeta(t_k) \quad (4.14)\end{aligned}$$

The term $\mathbf{P}_n(t)$ contains the impulse response of the filter and is defined by

$$\mathbf{P}^n(t_k) = [P_n(t_k) \cdots P_0(t_k)]^T \in \mathbb{R}^{(n+1)\times 1} \tag{4.15}$$

$$P_i(t) = \mathfrak{L}^{-1}\left[s^i P(s)\right] \tag{4.16}$$

$$P^I(t) = \mathfrak{L}^{-1}\left[\frac{P(s)}{s}\right] \tag{4.17}$$

$$P_c(t) = \mathfrak{L}^{-1}\left[P(s)C(s)\right] \tag{4.18}$$

Now for the step input

$$h[t_k - \delta] = ht_k - h\delta \tag{4.19}$$

Applying eqn(4.19) in eqn(4.14) and rearranging it we get an estimation equation in least-squares form

$$\begin{aligned} \underline{y_r}^I(t_k) &= -\bar{\mathbf{a}}_n \underline{\mathbf{y}_r}^{(n-1)}(t_k) + h\mathbf{b}_m \mathbf{P}_+^{m-1}(t_k - \delta) + b_0\delta h \\ &\quad + \left[\mathbf{c}_{n-1} + \bar{\mathbf{a}}_n q\right] \mathbf{P}^{n-1}(t_k) + qP^I(t_k) + \zeta(t_k) \end{aligned} \tag{4.20}$$

where,

$$\mathbf{P}_+^{m-1}(t_k - \delta) = \begin{bmatrix} \mathbf{P}^{m-1}(t_k - \delta) \\ P_c(t_k - \delta) + t_k \end{bmatrix} \tag{4.21}$$

Or equivalently

$$\gamma(t_k) = \phi(t_k)\theta + \zeta(t_k) \tag{4.22}$$

where,

$$\gamma(t_k) = \underline{y_r}^I(t_k) \tag{4.23}$$

$$\phi(t_k) = \begin{bmatrix} -\underline{\mathbf{y}_r}^{(n-1)}(t_k) \\ h\mathbf{P}_+^{m-1}(t_k - \delta) \\ h \\ \mathbf{P}^{n-1}(t_k) \\ P^I(t_k) \end{bmatrix} \tag{4.24}$$

$$\theta = [\bar{\mathbf{a}}_n \ \ \mathbf{b}_m \ \ b_0\delta \ \ \mathbf{c}_{n-1} + \bar{\mathbf{a}}_n q \ \ q] \tag{4.25}$$

Eqn(4.22) can be written for t_k, $k = 1, 2 \cdots N$, where N is the total number of available data points, and combined to give the estimation equation

$$\mathbf{\Gamma} = \mathbf{\Phi}\theta + \zeta \tag{4.26}$$

4.2.3 Identification under closed-loop conditions

Due to safety or economic reasons it may not be always possible to open control loops for identification. Also for unstable and marginally stable processes open loop test is not a practical option. In this section a closed loop identification method based on a step change in the set-point is introduced. The method directly estimates the parameters of the open loop transfer function model along with the time delay from closed loop data. We assume that the controller is completely known.

For a process model described by eqn(4.4) and for a known controller, $K(s)$, the fundamental relation between the output and set-point for an initial steady state condition of the setpoint can be described by

$$Y(s) = \frac{\mathbf{b}_m \mathbf{s}^m K(s) e^{-\delta s}}{\mathbf{a}_n \mathbf{s}^n + \mathbf{b}_m \mathbf{s}^m K(s) e^{-\delta s}} R(s) + \frac{\mathbf{c}_{n-1}\mathbf{s}^{n-1} + \mathbf{d}_{m-1}\mathbf{s}^{m-1} e^{-\delta s}}{\mathbf{a}_n \mathbf{s}^n + \mathbf{b}_m \mathbf{s}^m K(s) e^{-\delta s}} + W(s) \quad (4.27)$$

where, $R(s)$ is the Laplace transform of the set-point, $r(t)$, and $W(s)$ is the error term. Now, in equation error form the closed loop expression relating the output to the set-point can be expressed as

$$\mathbf{a}_n \mathbf{s}^n Y(s) = \mathbf{b}_m \mathbf{s}^m K(s) e^{-\delta s}\left[R(s) - Y(s)\right] + \mathbf{c}_{n-1}\mathbf{s}^{n-1} + \mathbf{d}_{m-1}\mathbf{s}^{m-1} e^{-\delta s} + V(s) \quad (4.28)$$

where,

$$\mathbf{d}_{m-1} = \left[d_{m-1}\; d_{m-2} \cdots d_0\right] \in \mathbb{R}^{1\times m} \quad (4.29)$$

$$d_{m-i} = \mathbf{g}_i \mathbf{y}^{m-1}(0), \quad i = 1 \cdots m \quad (4.30)$$

$$\mathbf{g}_i = \left[\mathbf{0}^{1\times(m-i)}\; b_m \cdots b_{m-(i-1)}\right] \in \mathbb{R}^{1\times m} \quad (4.31)$$

$$\mathbf{y}^{(m-1)}(0) = \left[y^{(m-1)}(0)\; y^{(m-2)}(0) \cdots y(0)\right]^T \quad (4.32)$$

Other notations have been defined in chapter 2. Now applying the filtering operation on both the output and the set-point with the filter $P(s) = \frac{1}{sA(s)}$ and rearranging the equation we get

$$\begin{aligned} \underline{Y}^I(s) \;=\; & -\bar{\mathbf{a}}_n \mathbf{s}^{n-1} \underline{Y}(s) + \bar{\mathbf{b}}_m \mathbf{s}^{m-1} K(s)\left[\underline{R}(s) - \underline{Y}(s)\right] e^{-\delta s} \\ & + b_0 K(s)\left[\underline{R}^I(s) - \underline{Y}^I(s)\right] e^{-\delta s} + \mathbf{c}_{n-1}\mathbf{s}^{n-1} P(s) \\ & + \mathbf{d}_{m-1}\mathbf{s}^{m-1} P(s) e^{-\delta s} + \epsilon(s) \end{aligned} \quad (4.33)$$

The controller transfer function $K(s)$ is different for different controller structures. For a PID controller we can write

$$K(s) = K_p + K'(s) \quad (4.34)$$

$$K'(s) = \frac{K_I}{s} + K_D s \quad (4.35)$$

where K_p, K_I and K_D are the proportional, integral and derivative constants, respectively. For P only controller $K'(s) = 0$ and for PI controller $K'(s) = \frac{K_I}{s}$. Following these notations and using eqn(4.8) we get

$$\begin{aligned} K(s)\underline{R}^I(s) &= K(s)\frac{R(s)}{sA(s)} \\ &= K_P\left[R^I(s) + C(s)\underline{R}(s)\right] + K'(s)\underline{R}^I(s) \end{aligned} \tag{4.36}$$

For a step change in set-point of magnitude h, $R(s) = \frac{h}{s}$. Applying eqn(4.36), we can write eqn(4.33) for a step change in the set-point as

$$\begin{aligned} \underline{Y}^I(s) &= -\bar{\mathbf{a}}_n\mathbf{s}^{n-1}\underline{Y}(s) + \bar{\mathbf{b}}_m\mathbf{s}^{m-1}K(s)\left[hP(s) - \underline{Y}(s)\right]e^{-\delta s} \\ &+ b_0\left[K_Ph/s^2 + hC(s)P(s) + K'(s)hP(s)/s - K(s)\underline{Y}^I(s)\right]e^{-\delta s} \\ &+ \mathbf{c}_{n-1}\mathbf{s}^{n-1}P(s) + \mathbf{d}_{m-1}\mathbf{s}^{m-1}P(s)e^{-\delta s} + \epsilon(s) \end{aligned} \tag{4.37}$$

In closed loop operation, for a proper controller structure the overall gain of the loop is unity. So in this case it becomes straightforward to get the data in deviation form from their raw values and we can write eqn(4.3) as

$$Y(s) = Y_r(s) - \frac{h_0}{s} \tag{4.38}$$

where, h_0 is the initial steady state value of the set-point. Using eqn(4.38) we can write eqn(4.37) after rearrangement

$$\begin{aligned} \underline{Y_r}^I(s) - h_0P^I(s) &= -\bar{\mathbf{a}}_n\mathbf{s}^{n-1}\left[\underline{Y_r}(s) - h_0P(s)\right] \\ &+ \bar{\mathbf{b}}_m\mathbf{s}^{m-1}K(s)\left[hP(s) - (\underline{Y_r}(s) - h_0P(s))\right]e^{-\delta s} \\ &+ b_0\left[K_Ph/s^2 + K_PhC(s)P(s) + hK'(s)P^I(s)\right. \\ &\quad \left. -K(s)\underline{Y_r}^I(s) + h_0K(s)P^I(s)\right]e^{-\delta s} \\ &+ \mathbf{c}_{n-1}\mathbf{P}^{n-1}(s) + \mathbf{d}_{m-1}\mathbf{P}^{m-1}(s)e^{-\delta s} + \epsilon(s) \end{aligned} \tag{4.39}$$

Taking inverse Laplace transform the above equation can be expressed in time domain at any time $t = t_k$

$$\begin{aligned} \underline{y_r}^I(t_k) - h_0P^I(t_k) &= -\bar{\mathbf{a}}_n\left[\underline{\mathbf{y_r}}^{(n-1)}(t_k) - h_0\mathbf{P}^{n-1}(t_k)\right] \\ &+ \bar{\mathbf{b}}_m\left[(h + h_0)\mathbf{P}_K^{m-1}(t_k - \delta) - \underline{\mathbf{y_r}}_K^{(m-1)}(t_k - \delta)\right] \\ &+ b_0\left[K_Ph[t_k - \delta] + K_PhP_c(t_k - \delta) + hP_{K'}^I(t_k - \delta)\right. \\ &\quad \left. + h_0P_K^I(t_k - \delta) - \underline{y_r}_K^I(t_k - \delta)\right] \\ &+ \mathbf{c}_{n-1}\mathbf{P}^{n-1}(t_k) + \mathbf{d}_{m-1}\mathbf{P}^{m-1}(t_k - \delta) + \varepsilon(t_k) \end{aligned} \tag{4.40}$$

Here, $P_K(t) = \mathfrak{L}^{-1}[K(s)P(s)]$ and other similar terms are defined in the same way. Now using the equation $h[t_k - \delta] = ht_k - h\delta$ we can get the estimation equation in a least-squares form

$$\gamma_+(t_k) = \phi_+^T(t_k)\theta_+ + \varepsilon(t_k) \tag{4.41}$$

where,

$$\gamma_+(t_k) = \underline{y_r}^I(t_k) - h_0 P^I(t_k) \tag{4.42}$$

$$\phi_+(t_k) = \begin{bmatrix} -\mathbf{y_r}^{(n-1)}(t_k) + h_0\mathbf{P}^{n-1}(t_k) \\ h\mathbf{P}_K^{m-1}(t_k-\delta) - \underline{\mathbf{y_r}}_K^{(m-1)}(t_k-\delta) \\ \Omega(t_k) - \underline{y_r}_K^I(t_k-\delta) \\ K_P h \\ \mathbf{P}^{n-1}(t_k) \\ \mathbf{P}^{m-1}(t_k-\delta) \end{bmatrix} \tag{4.43}$$

$$\Omega(t_k) = K_P h t_k + K_P h P_c(t_k-\delta) + hP^I_{K'}(t_k-\delta) + h_0 P^I_K(t_k-\delta) \tag{4.44}$$

$$\theta_+ = [\bar{\mathbf{a}}_n \;\; \bar{\mathbf{b}}_m \;\; b_0 \;\; b_0\delta \;\; \mathbf{c}_{n-1} \;\; \mathbf{d}_{m-1}] \tag{4.45}$$

Eqn(4.41) can be written for $t = t_k, k = 1, 2 \cdots N$, where N is the total number of available data points. To formulate the estimation equation for $t_k < \delta$, we need output data before the step input is applied. Hence we suggest recording some output data before the setpoint is changed. Combination of the N equations gives the estimation equation as

$$\mathbf{\Gamma}_+ = \mathbf{\Phi}_+\theta_+ + \varepsilon \tag{4.46}$$

4.2.4 Parameter estimation

The parameter vector can be obtained by solving eqn(4.26) for open loop data or eqn(4.46) for closed loop set-point and output data. However, there are two problems associated with the solution. First, for both of the cases we need to know $A(s)$ and δ, which are unknowns. This problem can be solved by applying an iterative procedure that adaptively adjust an initial estimate of $A(s)$ and δ until they converge. Second, the least-squares solution does not give unbiased estimate in the presence of general forms of measurement noise such as colored noise. To solve the bias problem, the instrumental variable (IV) method discussed in chapter 2 is used. The same bootstrap estimation of IV type where the instrumental variable is built from an auxiliary model (Young 1970) is considered. For the open loop method the instrumental variable is defined as

$$\psi(t_k) = \begin{bmatrix} -\underline{\hat{\mathbf{y}}}_r^{(n-1)}(t_k) \\ \phi(n : 2n+m+2, 1) \end{bmatrix} \tag{4.47}$$

where, $\hat{y}_r(t) = \hat{y}(t) + \hat{q}$, $\hat{y}(t) = \mathfrak{L}^{-1}[\hat{Y}(s)]$ and

$$\hat{Y}(s) = \frac{\hat{\mathbf{b}}_m\mathbf{s}^m}{\hat{\mathbf{a}}_n\mathbf{s}^n}U(s)e^{-\hat{\delta}s} + \frac{\hat{\mathbf{c}}_{n-1}\mathbf{s}^{n-1}}{\hat{\mathbf{a}}_n\mathbf{s}^n} \tag{4.48}$$

For identification under closed loop condition, following the above procedure, the instrument matrix is obtained by replacing the $y(t)$ in $\phi_+(t)$ by $\hat{y}(t)$ i.e.,

$$\psi_+(t_k) = \begin{bmatrix} -\underline{\hat{\mathbf{y}}_{\mathbf{r}}}^{(n-1)}(t_k) + h_0\mathbf{P}^{n-1}(t_k) \\ h\mathbf{P}_K^{m-1}(t_k-\delta) - \underline{\hat{\mathbf{y}}_{\mathbf{r}}}_K^{(m-1)}(t_k-\delta) \\ \Omega(t_k) - \underline{\hat{y}_r}_K^I(t_k-\delta) \\ \phi_+(n+m+2:2n+m+2,1) \end{bmatrix} \tag{4.49}$$

where, $y_r(t_k) = \hat{y}(t_k) + h_0$ and

$$\hat{Y}(s) = \frac{\hat{\mathbf{b}}_m\mathbf{s}^m K(s)e^{-\delta s}}{\hat{\mathbf{a}}_n\mathbf{s}^n + \hat{\mathbf{b}}_m\mathbf{s}^m K(s)e^{-\delta s}}R(s) + \frac{\hat{\mathbf{c}}_{n-1}\mathbf{s}^{n-1} + \hat{\mathbf{d}}_{m-1}\mathbf{s}^{m-1}e^{-\delta s}}{\hat{\mathbf{a}}_n\mathbf{s}^n + \hat{\mathbf{b}}_m\mathbf{s}^m K(s)e^{-\delta s}} \tag{4.50}$$

The iterative IV scheme is embedded within the iteration steps of the proposed method and no additional step is required.

From θ or θ_+we directly get the parameters $\bar{\mathbf{a}}_n$, $\mathbf{b}_m$, δ, q and $\mathbf{c}_{n-1}$. To retrieve $\mathbf{y}^{(n-1)}(0)$ from $\mathbf{c}_{n-1}$, eqn(2.6) can be written for $i = 1\cdots n$ to give

$$(\mathbf{c}_{n-1})^T = \mathbf{H}\mathbf{y}^{(n-1)}(0) \tag{4.51}$$

where, $\mathbf{H} = [(\mathbf{h}_1)^T\ (\mathbf{h}_2)^T \cdots (\mathbf{h}_n)^T]^T \in \mathbb{R}^{n\times n}$. Finally

$$\mathbf{y}^{(n-1)}(0) = (\mathbf{H})^{-1}(\mathbf{c}_{n-1})^T \tag{4.52}$$

4.2.5 Convergence of the iterative scheme

Extensive simulation study shows that the iterative procedure converges monotonically except for processes showing inverse response. Also the effect of the initial conditions on the response curve is similar to the inverse response for some cases. For both of these cases the iteration scheme diverse monotonically. To make the diverging scheme converge we use the same *ad hoc* procedure as described in (Ahmed *et al.* 2006) and in chapter 2 which has also been detailed in *Appendix A*.

4.3 Simulation study

4.3.1 Open loop identification

Example 1: Effect of initial condition

To demonstrate the effect of initial conditions on the estimation of the process model, a first order process having the following transfer function is used

$$G(s) = \frac{1.25}{20s+1}e^{-7s} \quad (4.53)$$

Figure 4.3 shows the response of the process to three successive steps. At the beginning of every step the process is very close to steady state. Data presented here are free of noise. For noisy data it is even harder to determine the steady state condition. Now to apply most of the methods available in literature, a preprocessing of the data is required. An approximated steady state value is first subtracted from the raw measurements to get the data in deviation form. To show the effect of this preprocessing, we will present results obtained using the MATLAB SYSID Toolbox. Besides other requirements such as regular sampling, SYSID Toolbox can handle data only in deviation form. If an initial steady state value is estimated from the data and data are preprocessed using that value, for the three steps, the toolbox gives three different models. Figure 4.4(a) shows the step response of the three models obtained using the estimated deviation form data from the three steps. From the figure we see that although the process reached very close the steady state values before the steps were applied, the estimated models differ from the true model as well as among themselves. In particular the gain values are different. On the other hand, if we use the proposed method, we get three models having almost the same parameters. The step response of the three model estimated using the proposed method are shown in figure 4.4(b). Here we see that the responses coincide and overlap with the step response of the true process.

It is worthwhile to mention here that in MATLAB there are some options to preprocess the data e.g., to remove mean and to choose a particular segment of data. The often used quick start option indeed detrends the data and chooses a suitable segment of the step response to do the identification. Though we are not presenting here any results using MATLAB's preprocessing steps, the SYSID toolbox gave very different models when the data from the three steps were used and the data were preprocessed using the quick start option. Simply removing the means also fails to

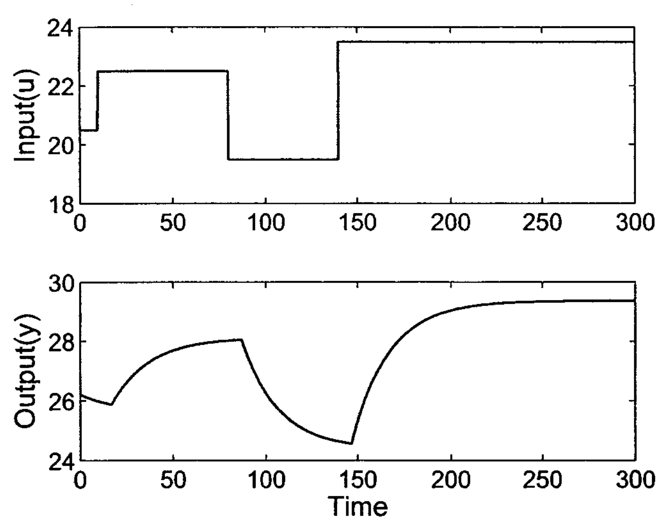


Figure 4.3: *Output response of the process (example 1) to three successive steps in the input.*

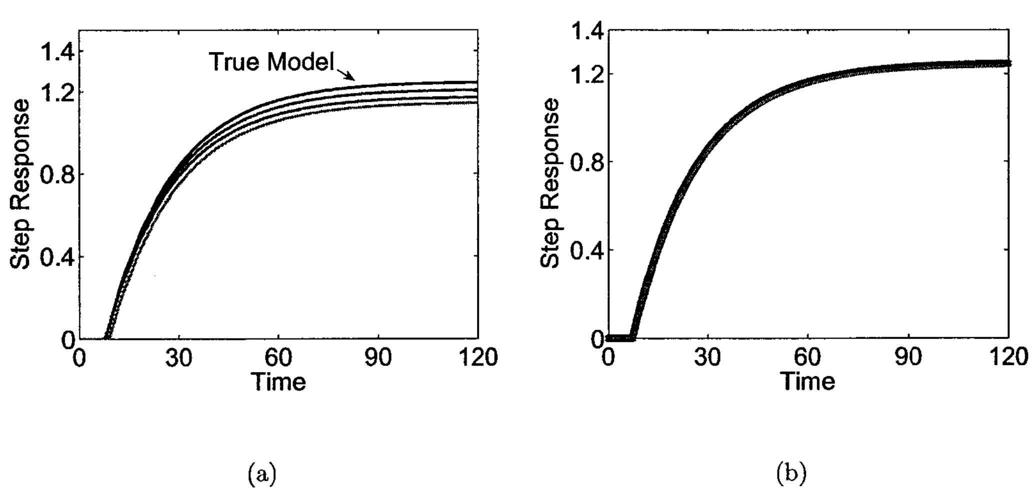


Figure 4.4: *Step responses of the estimated models using(a) MATLAB SYSID Toolbox (b) proposed method (example 1).*

produce consistent results for step response based identification.

Figure 4.5 shows the step responses of the same process when it is initially far

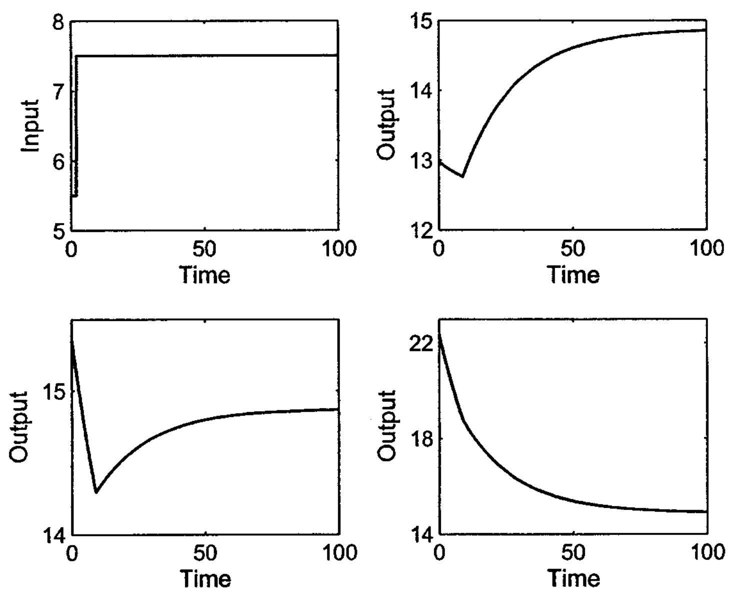


Figure 4.5: *Step responses from initial conditions far away from steady state (example 1).*

away from the steady state conditions. It is readily understandable that any sort of preprocessing by subtraction of an approximated steady state value may produce misleading results. In some cases the estimated gain may even have a wrong sign. However, using the proposed method we get models whose step responses coincide with that of the true process as shown in figure 4.6.

4.3.2 Identification under closed loop condition

Example 2: Unstable process

We consider here a second order process with a PI controller. The process and controller are represented by the following transfer function.

$$G(s) = \frac{s+1}{s^2+s-2}e^{-0.04s} \tag{4.54}$$

$$K(s) = 10+\frac{15}{s} \tag{4.55}$$

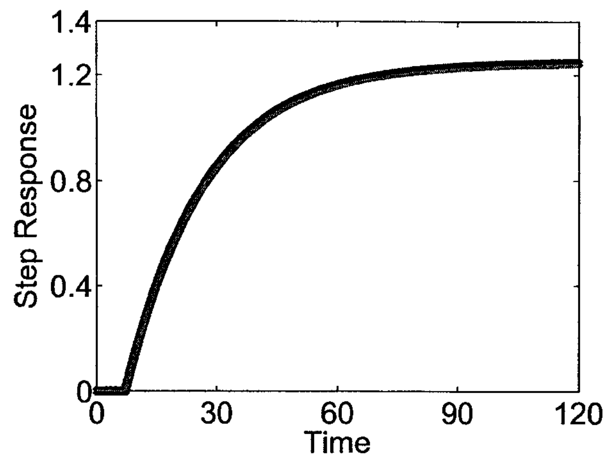
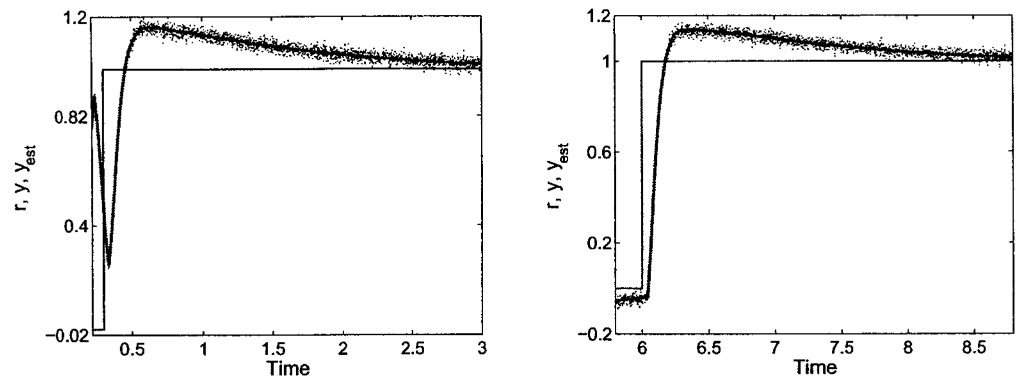


Figure 4.6: *Step responses of the estimated models using data when process initially at far away from steady state (example 1).*

This open loop unstable process has been considered in (Garnier *et al.* 2000), however, without any delay. The sampling interval was set to 1 ms. Figures 4.7(a) and 4.7(b) show the closed loop response of the unstable process and that of the estimated model for the same controller. The identification data set shows that the process was at an unsteady state when the step change was made in the set-point. The validation data show that the method gives a good estimate of the model parameters in the presence of initial condition. To study the effect of noise on the parameter estimates, 100 Monte Carlo simulations were carried out for a NSR of 10%. For this study a zero initial condition was assumed. Figure 4.8 shows the Bode diagram of the 100 estimated model for both least squares (LS) and instrumental variable (IV) estimation. It is seen that although the quality of the estimates are satisfactory for both cases the IV estimates are better than the LS estimates.

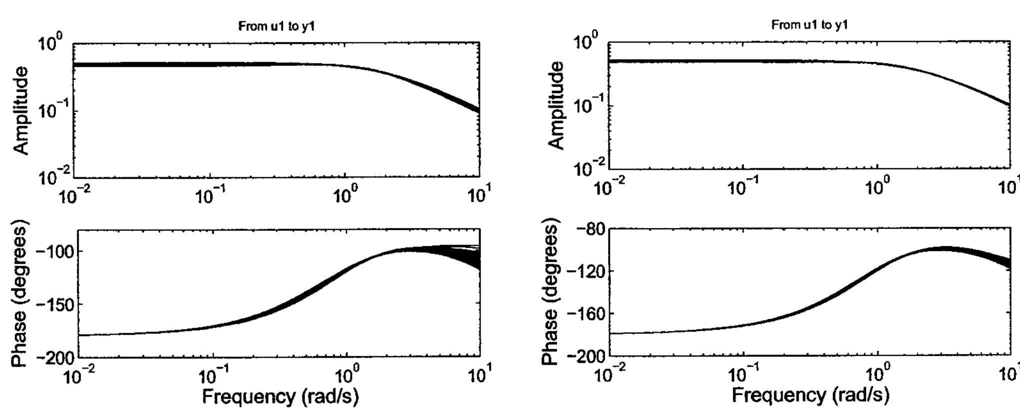
Example 3: Nonlinear unstable bio-reactor

Continuous bio-reactors are typical nonlinear unstable processes for which open loop identification is not possible. The processes have significant time delays arising from the measurement procedure. In this example a linear transfer function model with time delay is estimated from a closed loop step response of the nonlinear model of a bio-reactor. The following dynamic equations, steady state models and the



(a) (b)

Figure 4.7: *Closed loop step response of the process and model (example 2).(a)Identification data (b) Validation data.*



(a) (b)

Figure 4.8: *Bode diagram of the 100 Monte Carlo estimates (a)Least square estimates (b) Instrumental variable estimates (example 2).*

corresponding parameters of a bio-reactor are considered:

$$\frac{dx}{dt} = (\mu - D)x \tag{4.56}$$

$$\frac{ds}{dt} = D(s_F - s) - \frac{\mu}{y_0}x \tag{4.57}$$

$$\mu = \frac{\mu_m s}{K_s + s + s^2/K_i} \tag{4.58}$$

Here, x and s are the concentrations of the cell and substrate, respectively, μ is the specific growth rate, μ_m is the maximum specific growth rate, y_0 is the yield, K_s and K_i are the constants of the substrate inhibition model and D is the dilution rate which is the manipulated variable to control the concentration of cell in the reactor. The values of the parameters are

$$s_F = 4g/g \quad \mu_m = 0.53h^{-1} \quad y_0 = 0.4g/g \quad K_s = 0.12g/g$$

A delay of $1h$ is considered in the measurement of x. The reactor exhibits an unstable steady state at $(x = 0.9951, s = 1.5122)$ for a nominal value of dilution rate $D = 0.36h^{-1}$. The closed loop response of x, for a change in the set-point from 0.9951 to 1.1941, is obtained for a PID controller $k_c(1 + \frac{1}{\tau_I s} + \tau_D s)$ with $k_c = -0.7356, \tau_I = 4$ and $\tau_D = 0.2$. To avoid the derivative kick, the derivative action is applied only to the output and not on the error signal. The model and model parameters of the bio-reactor are taken from (Agrawal and Lim 1986, Ananth and Chidambaram 1999)

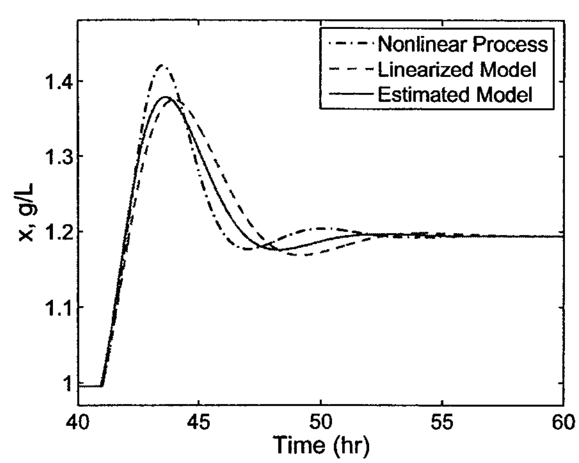


Figure 4.9: *Closed loop step response of the nonlinear bio-reactor model, linearized model and the estimated linear model (example 3).*

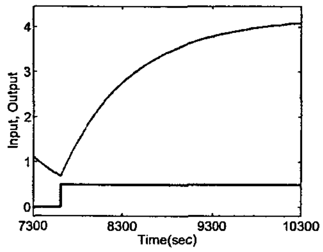
A first order plus time delay model with an unstable pole was estimated from the measured closed loop response. The closed loop response of the estimated model is plotted along with the closed loop response of the nonlinear model in figure 4.9. The figure also shows the closed loop response of the linearized model given in (Ananth and Chidambaram 1999). It can be seen from the figure that the match between the response of the estimated model and that of the nonlinear model is better than the match between the response of the linearized and nonlinear model.

4.4 Experimental evaluation

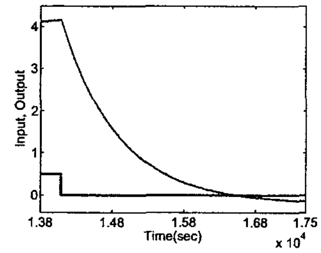
4.4.1 Open loop identification

A number of step tests from different unsteady initial conditions are performed in a laboratory scale mixing process. In chapter 3 a brief description and a pictorial representation of the process is provided.

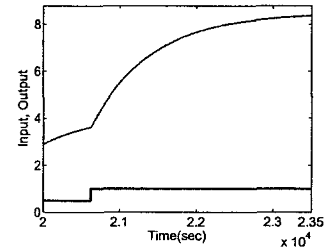
Figures 4.10(a), 4.10(b) and 4.10(c) show the concentration profile of the salt solution in the tank resulting from a step change in the feed flow rate. It can be seen from the figures that the initial concentration in the tank were not at steady states when the step changes were made. Figure 4.11 shows the step response and frequency responses of the three models estimated from the three sets of data. It can be concluded from the responses that the different data sets result almost the same estimated models.



(a)

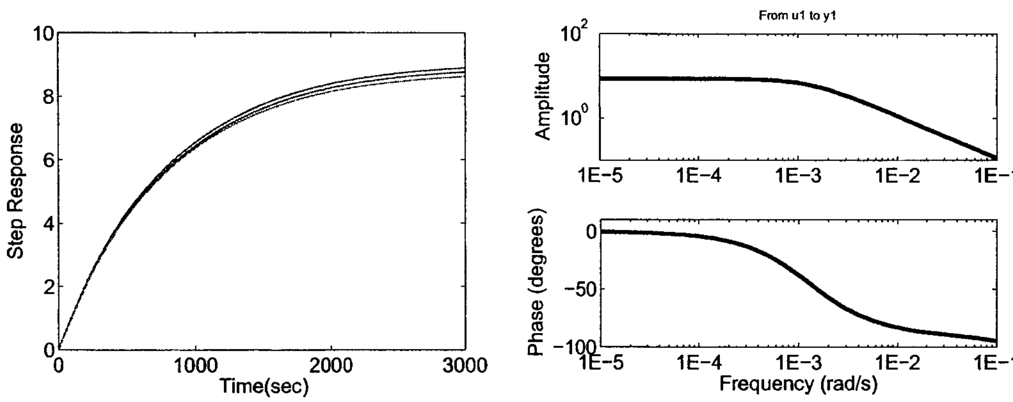


(b)



(c)

Figure 4.10: *Step response of the mixing process with different initial conditions.*



(a) (b)

Figure 4.11: *Step and frequency responses of the three identified model of the mixing process.*

4.4.2 Identification under closed loop condition

The proposed closed loop identification technique is applied for the identification of a continuous stirred tank heating (CSTH) process shown in figure 4.12. The cylindrical glass tank is equipped with steam coil with a controlled input facilitating the manipulation of steam flow to control the temperature of water in the tank. Also the level of water is controlled by manipulating the inlet water flow. The water outlet and condensate flow is controlled only manually. A number of thermocouples are placed at different distance in the tank outlet flow line that introduce time delay in the system. The set-up is under Emersons Delta-V distributed control system (DCS).

In this exercise, the set-point for the temperature of water in the tank is changed from 30^0C to 40^0C and the temperature of the outlet water was measured and recorded at 5 seconds intervals. A PI controller having a gain of 4.85 and reset time of 100 seconds was in place for the control loop to manipulate the steam valve. The level of water in the tank was controlled to be constant. A second order plus time delay (SOPTD) model was estimated as the open loop transfer function between temperature and steam flow. Figure 4.13 shows the closed loop response of the process and the estimated model for the PI controller. It can be concluded that the two responses match quite well.

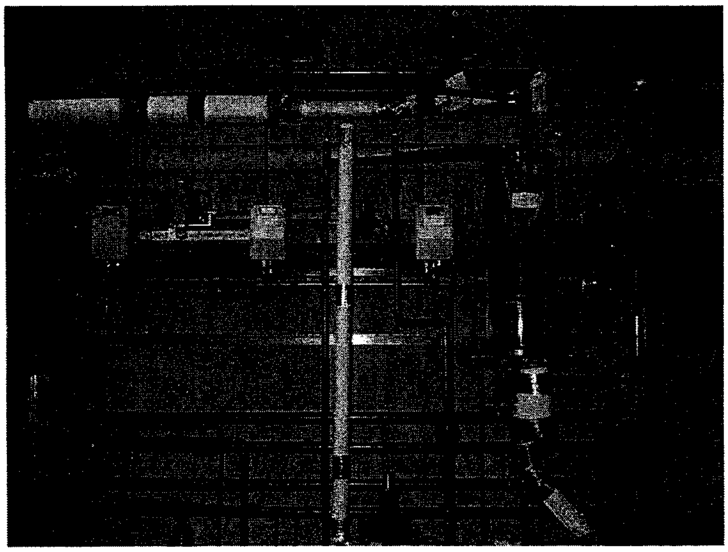


Figure 4.12: *Part of the CSTH process.*

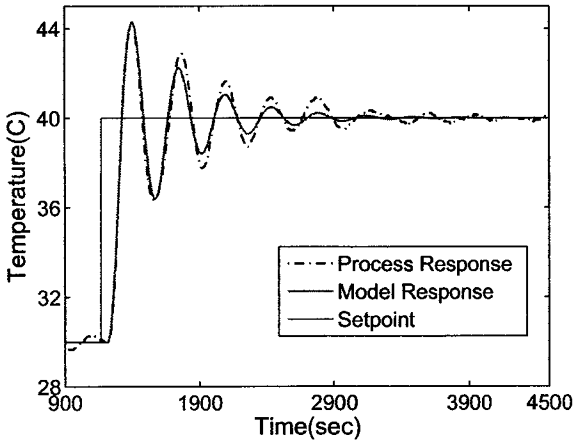


Figure 4.13: *Closed loop response of the heating tank process and its estimated model for a step change in steam flow.*

4.5 Conclusions

Identification from step response is a popular and commonly used method. In spite of this there are challenges in applying these methods in real life implementations. In this chapter two specific issues related to step response based identification have been addressed. Detailed mathematical derivations have been presented to show how, under both open loop and closed loop framework, process model parameters and the delay can be estimated from raw data even when the process is not initially at steady state. Formulation of the estimation equation in terms of raw form of the variables is an unique feature of the proposed algorithms that allows the use of industrial data without much preprocessing. Through simulations, the applicability of the methods has been demonstrated for a diverse group of processes including a nonlinear process. Finally, the performance of the methods is evaluated by experimental evaluations under both open loop and closed loop conditions.

Chapter 5

Identification of MIMO systems

5.1 Introduction

Most industrial processes are multivariate in nature and it is the complex interaction of different variables that makes control of such processes difficult. This challenging problem has acted as a major driving force for research into multivariate control that has resulted in an upsurge in process control interest both in academia and in industry. Much of the research has focused on new developments in model based control technologies that require a model of the multivariate process. The performance of any such technology depends mainly on the quality of the model used. However, there is an impression in the process control community that the development in the field of system identification is behind the development of the control algorithms (Zhu 1998). This impression apparently contradicts the fact that system identification is one of the most active areas of control research. Over the last several decades extensive work on system identification has been done resulting in publication of a large number of books and numerous research articles. However, an astonishing fact is that most of the identification results developed so far are not used by industrial control engineers. Some reasons for this failure in technology transfer have been pointed out by Zhu (1998). We believe that the limitations of the available methods act as a major barrier in their applications. Methods requiring a specific test procedure or those applicable for processes with specific order cannot find general acceptability due to obvious reasons.

The theoretical developments of MIMO system identification has been mainly based on the state space framework. Unfortunately, state space identification methods have not found widespread application in process industries. Outside the domain of state space, most of the identification methods for MIMO systems available in the literature results in a nonparametric frequency response as in (Melo and Friedly 1992) instead of the transfer functions. Another group of methods as in (Mei *et al.* 1992, Wang and Zhang 2001*a*) first obtain the frequency response and then estimate the step response of the individual input-output channels from which one can get the transfer functions. The procedure to obtain the frequency response from a transfer function is trivial whereas to obtain the transfer function from a frequency response is not straightforward. For the purpose of simulation, prediction or controller design the transfer function models are much more useful than the nonparametric frequency response models. For the same model accuracy, a parametric model needs much shorter test time; or put another way, for the same test data, a parametric model can be much more accurate (Zhu 2000). Regarding test signals, most of the methods use the idea of sequential step test. In this technique a step input is applied to one manipulated variable while the other inputs are kept constants. However, data from a single variable test may not contain good information about the multivariable character of the process. Also, step signals may not have enough excitation energy to obtain dynamic information of the process (Zhu 1998). Sequential relay experiment that has gained much popularity in the MIMO identification research also suffers from the limitations of single variable test. Contrary to the single variable thinking, several authors have proposed the idea of simultaneous excitation (Zhu 2000). The sequential test procedures requires proper supervision. Another issue in MIMO identification is that methods are specific for open loop or closed loop operation or for relay feedback experiment. Ideally, if possible, the identification test should be done under closed-loop condition. One advantage of closed-loop test is obvious: the controller helps to keep control variables within their limit. What is less obvious is that a model identified from closed-loop operation gives better control performance (Hjalmarsson *et al.* 1996) provided that the identification method can handle closed-loop data properly (Zhu 1998).

In view of the limitations of the existing methods, we present a general framework for continuous-time MIMO process identification that is applicable to both closed loop and open loop identification with minor modification. We consider direct approach for closed loop identification which requires no knowledge of the controller and indifferent on the controller structure. With respect to the test signals, both sequential testing and simultaneous testing can be applied in this procedure. More importantly, the results of the algorithm are the continuous-time transfer functions of the individual input-output channels of the process that can be used for controller design, simulation or prediction or from which we can directly get the nonparametric models. Additional advantage is that the time delays are estimated simultaneously with the other process parameters.

This work is motivated by an earlier work of Li *et al.* (2005) where the authors developed a closed-loop identification procedure for two-input two-output (TITO) processes with known decentralized controllers using the idea of sequential step test. In line with this development, we present here a general algorithm for MIMO process with any order including processes for which the number of inputs is different from the number of outputs. The developed technique is applicable for open loop as well as closed loop identification. Also the processes can be excited simultaneously through all its channels and the input signals are not necessarily step inputs. The method proposed by (Li *et al.* 2005) approximates the time delay by Taylor series expansion which may introduce significant error in the parameter and delay estimation. The algorithm developed in this work does not use any approximation for the time delay.

The basic idea behind the identification scheme is to decompose the MIMO data matrix into a number of equivalent SISO data matrices. The main advantage extracted from this decomposition is that the parameters of each individual transfer function are estimated separately. This disintegrate the complex parameter estimation equation of the MIMO process into a set of independent equations each corresponding to the individual transfer function of a single input output channel. However, a single experiment is not enough for this algorithm to be applied. Rather, we have to perform N_u (number of inputs) similar experiments either sequentially or separately with different input signals each time. The requirement is that the experiments should start at steady state conditions of the process. It is to be noted that the sequential step testing or relay testing are equivalent to the multiple test procedure adopted in this method.

In this work we present an identification method applicable under both open loop and closed loop operation. However, only the direct approach of closed loop identification is considered which is similar to open loop identification except that the process input and output data are collected from closed loop experiments. This gives rise to some issues in parameter estimation. Before going into the mathematical derivation a brief discussion on these issues is presented.

5.2 Issues in closed loop identification

Closed loop identification has been categorized in the literature in different ways. Gustavsson *et al.* (1977) classified closed loop identification techniques based on the assumptions on the availability of knowledge of the controller. This classification was later adopted by Soderstrom and Stoica (1989) and Forssell and Ljung (1999) where three major groups are considered:

1. The direct approach: Ignore the feedback and identify the open loop system using the measurements of the inputs and the outputs assuming no knowledge of the controller or the set-point signals or extra inputs.

2. The indirect approach: Assume complete knowledge of the controllers and using extra inputs or set-point signals a closed-loop transfer functions are estimated from which the open loop process parameters are extracted.

3. The joint input-output approach: Assume an unknown regulator with a certain structure and regard the input and output jointly as the output of a system driven by some extra input or set-point signals and noise. Use an identification methods to determine the open loop parameters from an estimate of this augmented system.

The indirect approach of closed loop identification not only needs complete knowledge of the controller but also assumes its simple structure. The joint input-output approach, too, is suitable for controllers having simple structures. However, in practice controllers may have complex structure or may be nonlinear. Also the input constraint may induce some nonlinear relation between the controller output and the error signal. Because of the limited applicability of these two approaches, when the objective of the identification exercise is to obtain the open loop model, the direct approach is the obvious choice. According to Ljung (1999), this approach should be seen as the natural approach to closed loop data analysis.

However, there are some difficulties associated with the direct approach. When closed loop data are used for open loop identification, a major difficulty arises because the closed loop data contains two relationships. The forward path relationship from the input to the output caused by the process and the feedback path relationship from the output to the input caused by the controller. The next part of the discussion is extracted from (Doma *et al.* 1996). The feedback path can cause a bias in the open loop model even when independent external test signals are added to the system. This is particularly important in nonparametric models, such as step response models, because the identified model can easily become a mixture of the open loop dynamics and the negative inverse of the controller dynamics.

The extent of this bias is determined by many factors, but two of them relate to process operation. The tuning of the controller determines the strength of the feedback path relationship. A very de-tuned controller means that the control actions will be so sluggish that the forward path dominates the data. In this case the closed loop data is almost open loop data and the bias caused by the feedback path is very small. The other factor that influences the extent of the bias is the magnitude of disturbances that may affect the process during the data collection period. This is a signal-to-noise issue. If the magnitude of the disturbance is small compared to the magnitude of the external test signal, then the feedback path is also small compared to the forward path. In the extreme, there will be no bias in the open loop model identified if there are no disturbances.

A practical closed loop identification method can not demand that the controller be de-tuned during data collection or simply wait for periods of small disturbances. Fortunately, the bias caused by the feedback path can be reduced by filtering the data. In addition to filtering the data, the bias can be further reduced by choosing the correct structure of the open loop model. Feedback controllers are designed to have dynamics that are different from the open loop process dynamics. In the extreme case, a deadbeat controller has dynamics that contain the exact inverse of the open loop process dynamics. Feedback controllers invariably respond without delay when the dependent variables deviate from their setpoints, whereas a physical process normally has some delay. Hence, the presence of deadtime is an important difference between the controller dynamics and the open loop process dynamics. Since the feedback path relationship originates in the controller, choosing a structure for the open loop model that matches the forward path dynamics will force the identification method to ignore the feedback path relationship. This is an important aspect of parametric model identification where the structure of the model is explicitly chosen. Nonparametric models such as step response models have virtually no structure. This makes them vulnerable to the feedback path relationship in closed loop data.

The theoretical developments on the bias issue in closed loop identification can be found in (Forssell and Ljung 1999, Goodwin and Welsh 2002, Karimi and Landau 1998, Van den Hof 1998).

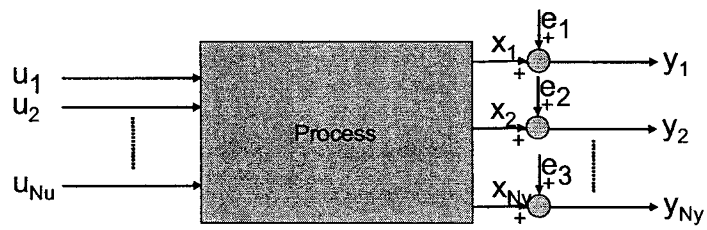


Figure 5.1: *Block diagram of a MIMO open-loop process with N_u inputs and N_y outputs.*

5.3 Mathematical formulation

5.3.1 From MIMO to SISO

The basic idea in the proposed approach is to decompose the MIMO data matrix into equivalent SISO data matrices by doing some algebraic manipulation. The main advantage of this decomposition arises from the fact that in the parameter estimation steps, parameters of each transfer function are estimated at one time. Thus a set of estimation equations are solved each giving the parameters corresponding to a specific transfer function. The mathematical manipulation for open loop and closed loop setups is described below.

Open loop identification

Consider the MIMO system in open loop described in figure 5.1. where, $u_j,\ j = 1, 2\cdots N_u$ are the inputs, $x_i,\ i = 1, 2\cdots N_y$ are the noise-free outputs and $y_i,\ i = 1, 2\cdots N_y$ are the noisy measurements of the outputs. So, the dimension of the MIMO system is $N_u \times N_y$. The basic relations between the outputs and the inputs can be described using the Laplace domain expressions

$$X_i = \sum_{j=1}^{N_u} G_{ij} U_j \tag{5.1}$$

$$Y_i = X_i + E_i, \qquad i = 1, 2\cdots N_y. \tag{5.2}$$

where, G_{ij} is the transfer function between the $i-th$ output and $j-th$ input. We will use upper case letters for variables in the Laplace domain and the corresponding lower case letters for that in the time domain. However, the indices (s) and (t) denoting the two domains will be frequently omitted for simplicity. Here E_i represents the disturbance term and we assume that the disturbances in the output signal that is not from the input excitation can be lumped into the additive terms E_i.

As mentioned earlier, for a process with N_u inputs, N_u different experiments are to be performed. One or more or all of the inputs can be changed at each experiment. Irrespective of how the inputs are changed, the mathematical formulation will be presented based on the assumption that all the inputs are changed during each experiment. Described below is the procedure for the $l-th$ experiment. N_u similar experiments are to be carried out using different sets of inputs each time.

Experiment l: Excite the process through its different input channels with the input signals u_j^l, $j = 1, 2 \cdots N_u$ and record the input and output measurements.

The input output relation for the $l-th$ experiment can be described by

$$X_i^l = \sum_{j=1}^{N_u} G_{ij} U_j^l, \qquad i = 1, 2 \cdots N_y. \tag{5.3}$$

Eqn(5.3) can be written for all the outputs i.e. for $i = 1, 2 \cdots N_y$ and can be combined and represented in a matrix form

$$\begin{bmatrix} X_1^l \\ X_2^l \\ \cdots \\ X_{N_y}^l \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1N_u} \\ G_{21} & G_{22} & \cdots & G_{2N_u} \\ \cdots & \cdots & \cdots & \cdots \\ G_{N_y1} & G_{N_y2} & \cdots & G_{N_yN_u} \end{bmatrix} \begin{bmatrix} U_1^l \\ U_2^l \\ \cdots \\ U_{N_u}^l \end{bmatrix} \tag{5.4}$$

N_u experiments result in a set of N_u equations similar to eqn(5.4). Combination of all the equations in a single expression gives

$$\mathbf{X} = \mathbf{G}\mathbf{U} \tag{5.5}$$

where,

$$\mathbf{X} = \begin{bmatrix} X_1^1 & X_1^2 & \cdots & X_1^{N_u} \\ X_2^1 & X_2^2 & \cdots & X_2^{N_u} \\ \cdots & \cdots & \cdots & \cdots \\ X_{N_y}^1 & X_{N_y}^2 & \cdots & X_{N_y}^{N_u} \end{bmatrix} \in \mathbb{R}^{N_y \times N_u} \tag{5.6}$$

$$\mathbf{U} = \begin{bmatrix} U_1^1 & U_1^2 & \cdots & U_1^{N_u} \\ U_2^1 & U_2^2 & \cdots & U_2^{N_u} \\ \cdots & \cdots & \cdots & \cdots \\ U_{N_u}^1 & U_{N_u}^2 & \cdots & U_{N_u}^{N_u} \end{bmatrix} \in \mathbb{R}^{N_u \times N_u} \tag{5.7}$$

$$\mathbf{G} = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1N_u} \\ G_{21} & G_{22} & \cdots & G_{2N_u} \\ \cdots & \cdots & \cdots & \cdots \\ G_{N_y1} & G_{N_y2} & \cdots & G_{N_yN_u} \end{bmatrix} \in \mathbb{R}^{N_y \times N_u} \tag{5.8}$$

Eqn(5.5) can be solved for $\mathbf{G}$ to give

$$\mathbf{G} = \mathbf{X}\mathbf{U}^{-1} \tag{5.9}$$

$\mathbf{U}^{-1}$ can be expressed as

$$\mathbf{U}^{-1} = \frac{\mathbf{\Pi}}{det(\mathbf{U})} \tag{5.10}$$

$$\mathbf{\Pi} = \begin{bmatrix} \Pi_1^1 & \Pi_2^1 & \cdots & \Pi_{N_u}^1 \\ \Pi_1^2 & \Pi_2^2 & \cdots & \Pi_{N_u}^2 \\ \cdots & \cdots & \cdots & \cdots \\ \Pi_1^{N_u} & \Pi_2^{N_u} & \cdots & \Pi_{N_u}^{N_u} \end{bmatrix} \tag{5.11}$$

where, $\mathbf{\Pi} = adj(\mathbf{U})$, the adjoint matrix of $\mathbf{U}$ with Π_i^j is the cofactor of U_i^j, the element corresponding to the $i-th$ row and $j-th$ column of $\mathbf{U}$. $det(\bullet)$ is the determinant of the corresponding matrix. So we have

$$\mathbf{G} = \frac{\mathbf{X\Pi}}{det(\mathbf{U})} \tag{5.12}$$

The (i,j)-th element of $\mathbf{X\Pi}$ is given by

$$\mathbf{X\Pi}(i,j) = \sum_{k=1}^{N_u} X_i^k \Pi_k^j \tag{5.13}$$

Also we have

$$det(\mathbf{U}) = \sum_{k=1}^{N_u} U_i^k \Pi_i^k \qquad \text{for } 1 \le i \le N_u \tag{5.14}$$

From eqn(5.9) we get the expressions for individual transfer function

$$G_{ij} = \frac{\mathbf{X\Pi}(i,j)}{det(\mathbf{U})} = \frac{\sum_{k=1}^{N_u} X_i^k \Pi_k^j}{\sum_{k=1}^{N_u} U_i^k \Pi_i^k} \tag{5.15}$$

The following are the explicit expressions for the individual transfer functions for a 2×2 process.

$$G_{11} = \frac{X_1^1 U_2^2 - X_1^2 U_2^1}{U_1^1 U_2^2 - U_1^2 U_2^1} = \frac{X_{11}}{U} \tag{5.16}$$

$$G_{12} = \frac{X_1^2 U_1^1 - X_1^1 U_1^2}{U_1^1 U_2^2 - U_1^2 U_2^1} = \frac{X_{12}}{U} \tag{5.17}$$

$$G_{21} = \frac{X_2^1 U_2^2 - X_2^2 U_2^1}{U_1^1 U_2^2 - U_1^2 U_2^1} = \frac{X_{21}}{U} \tag{5.18}$$

$$G_{22} = \frac{X_2^2 U_1^1 - X_2^1 U_1^2}{U_1^1 U_2^2 - U_1^2 U_2^1} = \frac{X_{22}}{U} \tag{5.19}$$

From the above expression we can see that the MIMO system has been transformed into single input single output systems with the same input signal passing through the different process transfer functions.

However, X_i^l are not available, rather we have the output measurements as Y_i^l. If we assume that the noise terms in Y_{ij} can be combined to a single term V_{ij}, we have

$$Y_{ij} = X_{ij} + V_{ij} \tag{5.20}$$

or

$$Y_{ij} = G_{ij}U + V_{ij} \tag{5.21}$$

Eqn(5.21) forms the basis for the estimation of the transfer function G_{ij}. Here for all of the transfer functions the input expressed in the Laplace domain is

$$U = U_1^1 U_2^2 - U_1^2 U_2^1 \tag{5.22}$$

The equivalent expression in the time domain is

$$u = u_1^1 * u_2^2 - u_1^2 * u_2^1 \tag{5.23}$$

where, the symbol $*$ corresponds to the convolution operation. Similarly the output expressions can be expressed in the Laplace domain

$$\begin{aligned} Y_{11} &= Y_1^1 U_2^2 - Y_1^2 U_2^1 && (5.24) \\ Y_{12} &= Y_1^2 U_1^1 - Y_1^1 U_1^2 && (5.25) \\ Y_{21} &= Y_2^1 U_2^2 - Y_2^2 U_2^1 && (5.26) \\ Y_{22} &= Y_2^2 U_1^1 - Y_2^1 U_1^2 && (5.27) \end{aligned}$$

The expressions in the time domain for the outputs for the 2×2 process are

$$\begin{aligned} y_{11} &= y_1^1 * u_2^2 - y_1^2 * u_2^1 && (5.28) \\ y_{12} &= y_1^2 * u_1^1 - y_1^1 * u_1^2 && (5.29) \\ y_{21} &= y_2^1 * u_2^2 - y_2^2 * u_2^1 && (5.30) \\ y_{22} &= y_2^2 * u_1^1 - y_2^1 * u_1^2 && (5.31) \end{aligned}$$

Input output expressions for (2×3), (3×2), (3×3) and (4×4) processes are given in *Appendix B*.

h

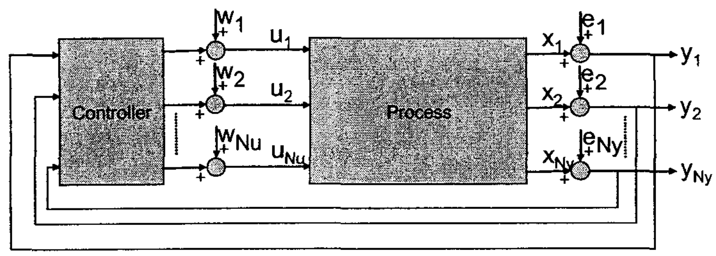


Figure 5.2: *Block diagram of a MIMO process under closed loop with dither inputs.*

Identification under closed loop conditions

Closed loop identification can be performed by exciting the process through the set-point channels or an extra signal, known as the dither signal, can be added to the input signal with the set-point constant. We will adopt the dither signal option. The set-up is described in figure 5.2. We consider the direct approach for closed loop identification where the process model is estimated directly using the process input and output information and ignoring the feedback mechanism. The fundamental problem with closed loop data is the correlation between the unmeasurable noise and the input which results in a bias error in the least squares solution of the parameters. However, the bias problem is not specific to closed loop identification. Open loop identification methods e.g., the linear filter method also gives a biased estimates unless some bias correction scheme is in place. The point here is that any open loop identification algorithm that incorporates in itself a bias elimination mechanism can be directly applied to direct closed loop identification with necessary modification in the bias elimination scheme.

In fact the direct closed loop identification approach differs from the open loop method only in the bias elimination scheme. All the mathematical derivation and estimation procedure remains the same.

5.3.2 Parameter estimation

The previous section presents how the MIMO system can be decomposed into equivalent SISO systems. It has been shown that the decomposition procedure finally yields an estimation equation for individual transfer functions that can be described in the form of a general expression

$$Y(s) = G(s)U(s) + V(s) \tag{5.32}$$

It has been shown that irrespective of the dimension of the process or the process configuration i.e open or closed loop, an equation similar to the above expression can be derived. They differ in how the equivalent inputs and outputs data are obtained. Here, we describe U and Y as the SISO equivalent of the MIMO process for an input output channel. The expressions for the equivalent input and output differ based on the process dimension.

To estimate the model parameters one can use any SISO identification technique. We will use the linear filter method described in chapter 3.

Linear filter method

Without going into further details of the method, following eqn(3.37), we directly give the the estimation equation as

$$\hat{\theta}^{LS} = \left[\mathbf{\Phi}^T\mathbf{\Phi}\right]^{-1}\mathbf{\Phi}^T\mathbf{\Gamma} \tag{5.33}$$

where,

$$\mathbf{\Phi} = [\phi(t_1)\ \ \phi(t_2)\cdots\phi(t_N)]^T \tag{5.34}$$

$$\mathbf{\Gamma} = [\gamma(t_1)\ \ \gamma(t_2)\cdots\gamma(t_N)]^T \tag{5.35}$$

with

$$\gamma(t_k) = \underline{y}_I(t_k) \tag{5.36}$$

$$\phi(t_k) = \begin{bmatrix} -\underline{\mathbf{y}}^{(n-1)}(t_k) \\ \underline{\mathbf{u}}_+^{(m-1)}(t_k - \delta) \\ u(t_k) \end{bmatrix} \tag{5.37}$$

As mentioned earlier, the least squares solution does not give an unbiased estimate. One of the most effective solution to the bias problem associated with the least square algorithms is to use the instrumental variable (IV) method since they do not require any a priori knowledge of the noise statistics. However, choice of the instrument is important for the IV method to be effective. The next section describes how to obtain the instruments for the different set-ups.

5.3.3 Choice of instruments

Gilson and Garnier (2003) have demonstrated effective use of an IV scheme for direct identification of systems under closed loop. It has been proved in (Gilson and Hof 2001) that in the discrete-time case, a tailor-made instrumental variable method is equivalent to the bias-eliminated least-squares method that gives an asymptotically unbiased estimates of model parameters. This result also holds in the continuous-time case (Gilson and Garnier 2003).

In chapter 2 we presented an IV method to get a bias free estimate of the open loop process parameters. A bootstrap estimation of IV type was adopted where the instrumental variables are built from an auxiliary model. However, the IV method mentioned above cannot be applied directly for the closed loop method because the input signal is now correlated with the noise.

Open loop method

For open loop identification, the input is not correlated with output. However, for colored noise the least square estimate is not bias free even for open loop identification. Following the procedure discussed in section 3.3.1, the instrumental variable for open loop identification can be defined as

$$\psi(t_k) = \begin{bmatrix} -\underline{\hat{\mathbf{x}}}^{(n-1)}(t_k) \\ \underline{\mathbf{u}}_{+}^{(m-1)}(t_k - \delta) \\ u(t_k) \end{bmatrix} \tag{5.38}$$

where,

$$\hat{X} = \hat{G}(s, \hat{\theta}^{LS})U \tag{5.39}$$

Closed loop method

To develop an IV method we will use the knowledge of the extra input signal known as the dither signal. So the required data set $\{y_i^l(t_k)\ u_j^l(t_k)\ w_j^l(t_k)\}\ k = 1, 2 \cdots N,\ j = 1, 2 \cdots N_y,\ j = 1, 2 \cdots N_u, l = 1, 2 \cdots N_u$ should contain the extra input signals w_j along with the process inputs and outputs. To generate the IV matrix we first define the part of the input due to the dither signal. For a 2×2 process this can be defined as

$$w = w_1^1 * w_2^2 - w_1^2 * w_2^1 \tag{5.40}$$

The part of the outputs y due to w is given by

$$\hat{Z} = \hat{G}_{ij}(s, \hat{\theta}^{LS})W \tag{5.41}$$

Finally the instrumental variable can be defined.

$$\psi(t_k) = \begin{bmatrix} -\hat{\underline{\mathbf{z}}}^{(n-1)}(t_k) \\ \underline{\mathbf{w}}_{+}^{(m-1)}(t_k - \delta) \\ w(t_k) \end{bmatrix} \tag{5.42}$$

5.4 Simulation study

To demonstrate the performance of the proposed methods and study the effects of different experimental conditions, a benchmark processes commonly used in the literature for identification of MIMO systems, namely the Wood and Berry distillation column (Wood and Berry 1973) having the following transfer function matrix is considered

$\begin{bmatrix} \frac{12.8e^{-s}}{} & \frac{-18.9e^{-3s}}{} \end{bmatrix}$

column (Wood and Berry 1973) having the following transfer function matrix is considered

T

$$\mathbf{G}(s) = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ 6.6e^{-7s} & -19.4e^{-3s} \end{bmatrix} \tag{5.43}$$

strong interaction and significant time delays, has long been used in identification literature (Melo and Friedly 1992, Li *et al.* 2005) as well as in the field of controller design (Huang *et al.* 2003*b*, Wang *et al.* 1997).

5.4.1 Simulation conditions

To obtain necessary data for the purpose of identification, a SIMULINK block representing the MIMO process was created. The individual transfer functions were represented by continuous-time transfer functions followed by time delay blocks. For the closed-loop simulation, controllers were applied using PID controller blocks. For all the cases, the inputs were multi-sine signals generated using the idinput function in MATLAB. The sampling time for all the inputs and outputs was 0.2 seconds which is approximately $\frac{1}{50}th$ of the smallest time constants. For continuous-time identification sampling intervals are recommended as $\frac{\tau}{50}$ (Garnier *et al.* 2003), where τ is the time constant of the process. For each experiment a total of 2000 data points for each of the variables were generated. The outputs were corrupted using white noise sequences of appropriate variances to give the desired noise to signal ratio (NSR).

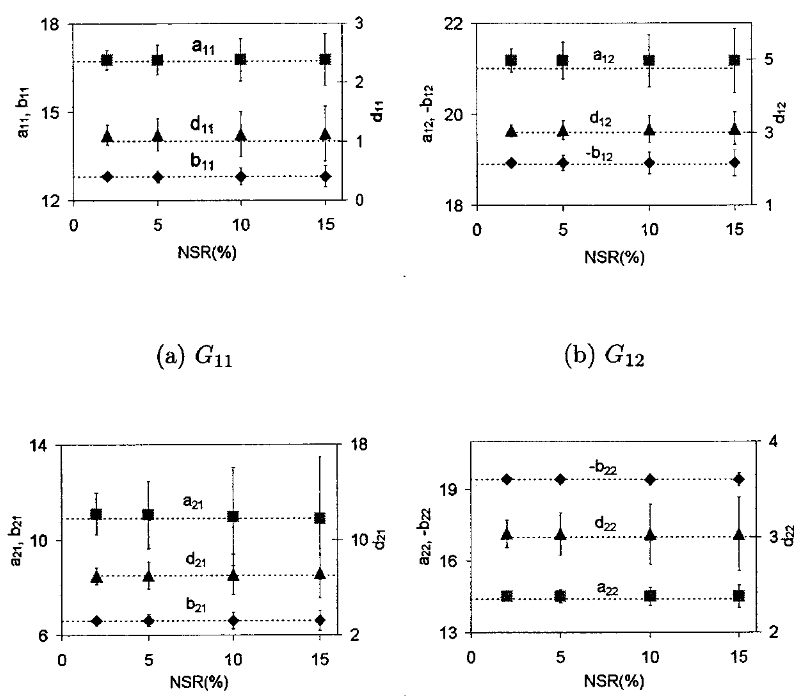
For the closed-loop experiments, a decentralized controller was used. A proportional only controller used by Melo and Friedly (1992) was used in this simulation study.

$$\mathbf{G}_c(s) = \begin{bmatrix} 0.1 & 0 \\ 0 & -0.05 \end{bmatrix} \tag{5.44}$$

5.4.2 Results

Open loop identification

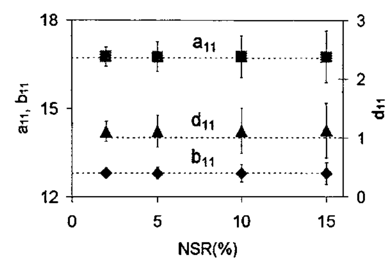
The 2×2 process was subject to open-loop experiment with different NSR. Presented below are the identification results of 100 Monte Carlo simulations (MCS). The mean value of the 100 estimates of the parameters are shown along with their standard deviation. Each subfigure presents the parameters of an individual transfer function of the Wood-Berry column.



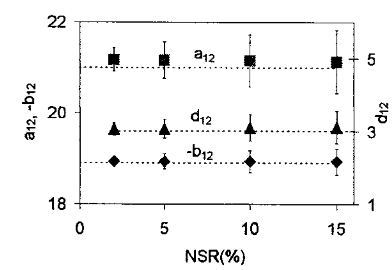
(a) G_{11} (b) G_{12}

(c) G_{21} (d) G_{22}

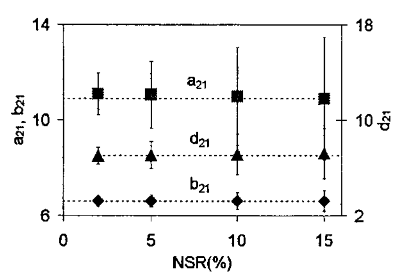
Figure 5.3: *Results of Open loop instrumental variable method.*



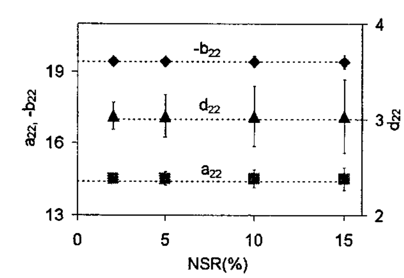
(a) G_{11}



(b) G_{12}



(c) G_{21}



(d) G_{22}

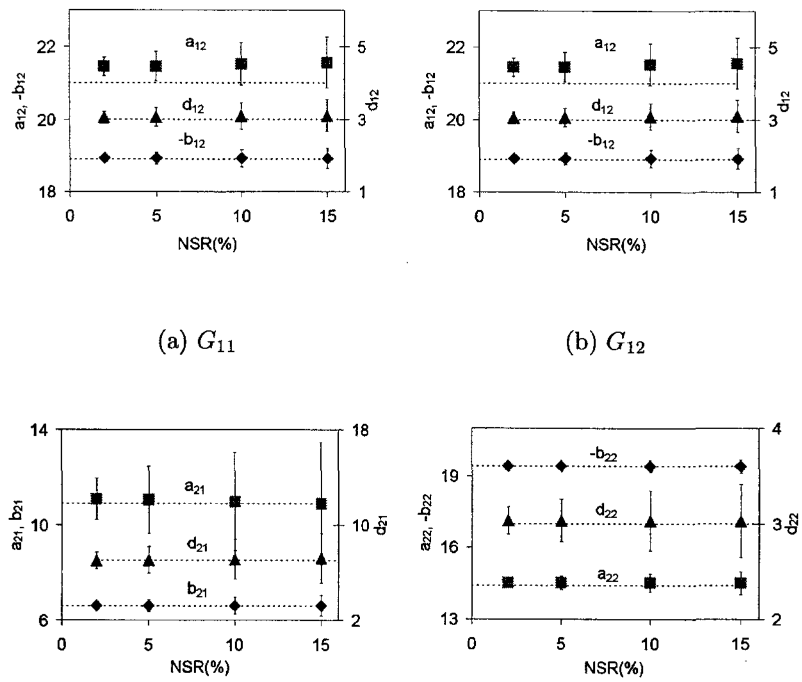
Figure 5.4: *Results of Open loop least squares method.*

Closed loop identification

Following the procedure described in section 5.3.1, the process was subjected to a closed loop experiment. The multi-sine inputs used in the open-loop tests were used for the closed-loop experiments as dither signals.

Comparison of the IV and LS methods

To compare the performances between the instrumental variable (IV) and the least squares (LS) based methods we use the total error criterion defined by eqn(3.43). Figure 5.7 represent the errors as a function of the NSR for the open-loop and closed-loop methods. Here it is seen that for the open-loop identification the IV and the LS methods have the same level of errors. However, for closed-loop identification, the IV method gives better performance than the LS method.



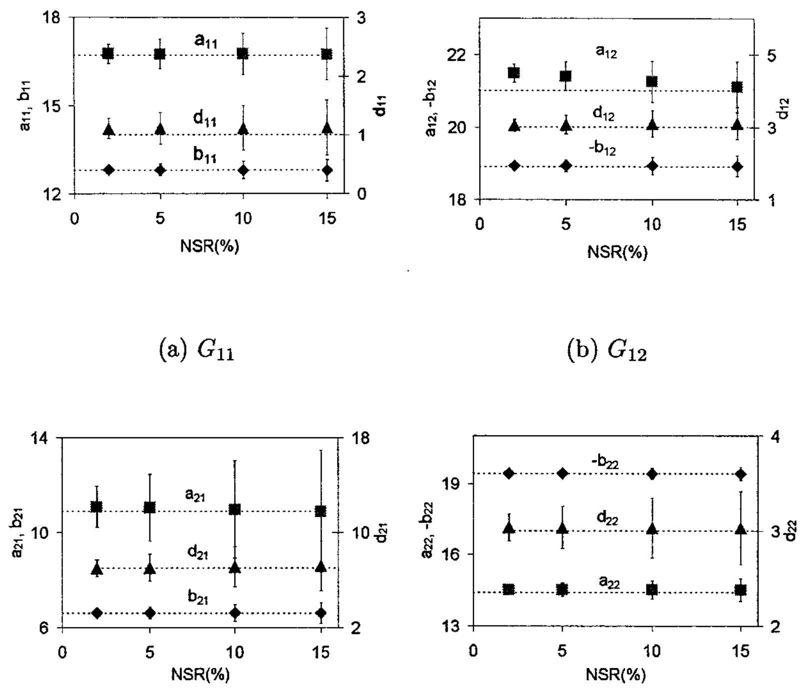
(a) G_{11} (b) G_{12}

(c) G_{21} (d) G_{22}

Figure 5.5: *Results of closed-loop instrumental variable method.*

5.5 Conclusions

Over the last few decades, significant developments of continuous-time single input single output (SISO) model identification has taken place whereas the identification of multiple input multiple output (MIMO) model has remained as a challenging task. In this chapter, a new identification technique for MIMO models has been introduced that uses a SISO algorithm in the parameter estimation stage. The main challenge of MIMO model identification is the dimensionality of the parameter vector. In this procedure, the MIMO data matrix is decomposed into a set of SISO data matrices each of which corresponds to one input output channel of the MIMO model. The SISO equivalents of the MIMO data are then used to estimate the parameters of each transfer function individually. Thus the problem of high dimensionality of the MIMO parameter estimation equation is overcome and the entire parameter vector is finally obtained from the solutions of a set of estimation equations. The explicit expressions for the equivalent inputs and outputs of different order models have been provided.

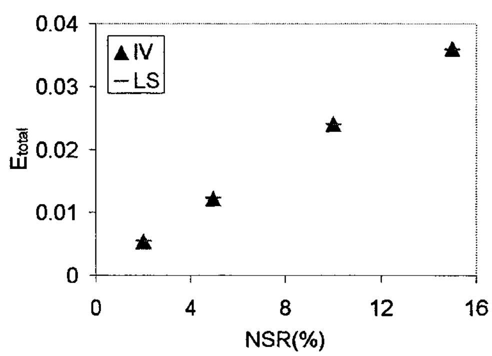


(a) G_{11} (b) G_{12}

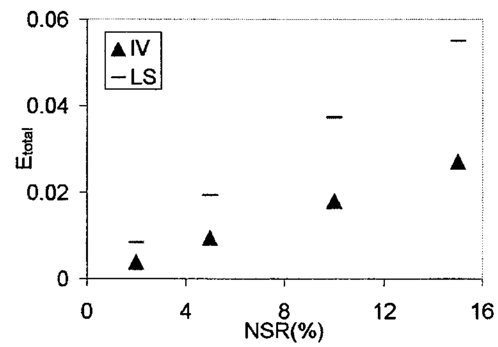
(c) G_{21} (d) G_{22}

Figure 5.6: *Results of closed-loop least squares method.*

Using a simulation example the applicability of the method has been demonstrated for both open loop and closed loop identification. The robustness of the method to noise has also been tested.



(a) Open-loop method



(b) Closed-loop method

Figure 5.7: *Total errors as a function of NSR for different estimation methods*

Chapter 6

Model Validation

6.1 Introduction

Model validation is considered as an integral part of system identification. Once a model of a process is identified, it is important to have model validation as the last quality control station before the model is delivered to the user (Ljung and Guo 1997). Also, it is equally important to continuously validate the model after it is placed for use and to maintain its quality by detecting changes in the parameters. Model validation is naturally related to detection of parameter changes (Huang *et al.* 2003*a*).

Methods for model validation through residual analysis have been reported in the literature; see e.g. (Box and Jenkins 1976, Ljung 1999, Ljung and Guo 1997, Soderstrom and Stoica 1989) and the references therein. The residual analysis techniques involve correlation test procedures consisting of tests using the autocorrelation function of the residuals and the cross-correlation functions between the residuals and the inputs. These tests are based on the principle that if the model structure is correct and the estimated parameters are unbiased, the residuals should form an independent random sequence and should be unpredictable from all past inputs, outputs and residuals (Mao and Billings 2000). From statistical efficiency point of view, Benveniste *et al.* (1987) have suggested an asymptotic local approach for model validation. It is shown by (Basseville 1998) that regular residuals as used by most model validation algorithms are not sufficient statistic. Basseville (1998) further shows that by utilizing the information of the gradient of the prediction error, the local detection algorithm provides more information than the regular residual analysis and is asymptotically sufficient statistic.

In recent years, new developments of model validation algorithms based on the local approach have been reported. Huang and Tamayo (2000) and Huang *et al.* (2003*a*) have presented model validation method for industrial model predictive control systems. The problem of multivariate model prediction in the presence of time variant disturbance dynamics has been addressed in (Huang 2000). Huang (2001) also presented an on-line model validation algorithm for processes under closed-loop. Model validation in a prediction error framework has been presented by (Gevers *et al.* 2003).

The references mentioned above mainly deal with the validation problem of discrete-time models. The literature on continuous-time model validation is sparse and to the best of the knowledge of the author, the change detection of the time delay has not been reported in the literature. In this chapter we present validation methods for SISO and MIMO continuous-time models having time delay. The methods treat the delay as another system parameter and capable of detecting its change.

The validation scheme is based on the local approach for detecting changes in the model parameters. Before presenting the validation methods, we first briefly describe the local approach for change detection as presented in (Basseville 1998) and (Basseville and Nikiforov 1993).

6.2 The local approach for change detection

The detection algorithms for additive and non-additive changes differ greatly in terms of complexity. For additive changes, it means the transformation of observations into residuals and detecting the change in the mean value of a Gaussian process. This is also termed as *basic problem.* But for non-additive changes, neither transformation into innovations nor to parity checks provide a sufficient statistics. Rather the sufficient statistics is the likelihood ratio. But using the likelihood ratio for change detection is computationally complex (Basseville and Nikiforov 1993). Also it is a non-linear function. The *local approach* is one of the tools that simplify the complex change detection problem into a basic problem. The relevant sufficient statistic here is the efficient score (Basseville and Nikiforov 1993). The efficient score is recovered by linearizing the log-likelihood ratio over a local point, namely the nominal model of the process.

Before going into the mathematical statement of the problem, it is necessary to point out the difference between model validation and change detection. Model validation is seen as an off-line process where the objective is to verify whether the model estimated previously is valid. On the other hand, change detection algorithms are often run on-line to detect changes in the model parameters. Mathematically both of the problems can be formulated as a test between two simple hypotheses say H_0 versus H_1. These two problems take slightly different forms.

Model validation:

$$\begin{aligned} H_0 : \theta &= \theta_0 && \text{for } k = 1, \ldots, N \\ H_1 : \theta &= \theta_0 + \frac{\tilde{\theta}}{\sqrt{N}} && \text{for } k = 1, \ldots, N \end{aligned}$$

On-line change detection:

$$\begin{aligned} H_0 &: \theta = \theta_0 && \text{for } k = 1, \ldots, N \\ H_1 &: \text{there exist some } \tau \in (0,1) \text{ such that} \\ &\quad \theta = \theta_0 && \text{for } 1 \leq k < \tau N \\ &\quad \theta = \theta_0 + \frac{\tilde{\theta}}{\sqrt{N}} && \text{for } \tau N \leq k < N \end{aligned}$$

Here, θ_0 is the parameter vector of the true system, unknown but represented by observations, θ, which is obtained under normal operating conditions. $\tilde{\theta}$ is an arbitrary vector with dimension same as that of θ. So, if the current parameter vector matches that at normal run, it is concluded that model is valid, otherwise, it is concluded that the model no longer represents the system. The formulation of $\tilde{\theta}$ tells that the change may take place in any of the parameters. The role of the local approach is that it transforms the hypothesis test problem into the monitoring problem of the mean of a Gaussian vector. In the local approach it is assumed that the two hypotheses get closer to each other when N grows to infinity and this is the asymptotic point of view.

Detailed mathematical derivation of the local approach can be found in (Basseville 1998, Basseville and Nikiforov 1993, Zhang *et al.* 1998). Here, we provide the summary.

Primary residuals: A vector-valued function $\rho(\theta, x_t)$ is a valid primary residual if

$$E_\theta \rho(\theta, x_t) = 0 \quad \text{for} \quad \theta = \theta_0$$

and

$$E_\theta \rho(\theta, x_t) \neq 0 \quad \text{for} \quad \theta \in \omega(\theta_0)\backslash\theta_0$$

and if it is differentiable in θ. Here, $\omega(\theta_0)\backslash\theta_0$ is read as a neighborhood of θ_0 exclusive of θ_0 and $x_t = [u(t)\ y(t)]^T$. *Improved Residual:* For a given primary residual $\rho(\theta, x_t)$, the improved residual is defined as:

$$\xi_N(\theta) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} \rho(\theta, x_t) \tag{6.1}$$

where N is the size of the sample. For sufficiently high N

$$\begin{aligned} \xi_N(\theta) &\sim \quad N(0, \Sigma(\theta_0)) \quad \text{if } \theta = \theta_0 \\ \xi_N(\theta) &\sim \quad N(-M(\theta_0)\tilde{\theta}, \Sigma(\theta_0)) \quad \text{if } \theta \neq \theta_0 \end{aligned}$$

where,

$$\begin{aligned} M(\theta_0) &= E\left(\frac{\partial}{\partial \theta} \rho(\theta, x_t) \bigg|_{\theta=\theta_0} \right) \\ &\approx \frac{1}{N} \sum_{t=1}^{N} \left(\frac{\partial}{\partial \theta} \rho(\theta, x_t) \bigg|_{\theta=\theta_0} \right) \\ \Sigma(\theta_0) &= \sum_{t=-\infty}^{\infty} Cov\left(\rho(\theta_0, x_1), \rho(\theta_0, x_t)\right) \\ &\approx \frac{1}{N} \sum_{t=1}^{N} \rho(\theta_0, x_t)\rho^T(\theta_0, x_t) \\ &\quad + \sum_{i=1}^{I} \frac{1}{N-i} \sum_{t=1}^{N-i} (\rho(\theta_0, x_t)\rho^T(\theta_0, x_{t+i}) \\ &\quad + \rho(\theta_0, x_{t+i})\rho^T(\theta_0, x_t)) \end{aligned}$$

We see that the residuals are different from the residuals defined in other fields. For example, residuals of regression analysis for a variable is a scalar. But here the residuals are vector valued and from their definition are sufficient statistics.

6.2.1 The detection problem

According to the above results, detection of changes in the model parameters, θ, is asymptotically equivalent to the detection of change in the mean of a Gaussian vector. The generalized likelihood ratio (GLR) test to detect change in the mean of a Gaussian vector is the χ^2 (chi-square) test. Basseville (1998) has shown that the GLR test of H_1 against H_0 can be written as

$$\chi^2_{global} = \xi_N(\theta_0)^T\Sigma^{-1}(\theta_0)M(\theta_0)\left(M(\theta_0)^T\Sigma^{-1}(\theta_0)M(\theta_0)\right)^{-1}M(\theta_0)^T\Sigma^{-1}(\theta_0)\xi_N(\theta_0) \tag{6.2}$$

If $M(\theta_0)$ is a square matrix, then this test can be further simplified to

$$\chi^2_{global} = \xi_N(\theta_0)^T\Sigma^{-1}(\theta_0)\xi_N(\theta_0) \tag{6.3}$$

χ^2_{global} has a central χ^2 distribution under H_0 and a noncentral χ^2 distribution under H_1. The degree of freedom of χ^2_{global} is the row dimension of θ. A threshold value, χ^2_α, can be found from a standard χ^2 table or can be calculated from training data based on the false alarm rate α specified by the user. If χ^2_{global} is found to be larger than the threshold value, it is concluded that there is a change in the parameter vector.

6.2.2 The isolation problem

Once a change in the parameter vector is detected, it is often desired to know which or which sets of parameters have changed. This is called the isolation problem.

To isolate the parameter or set of parameters undergone a change, θ is to be partitioned into θ_a and θ_b and $\tilde{\theta}$ is to be partitioned into $\tilde{\theta}_a$ and $\tilde{\theta}_b$ accordingly. Then the isolation problem can be formulated as the following hypothesis test:

$$H_0 : \tilde{\theta}_a = 0 \qquad \text{versus} \qquad H_1 : \tilde{\theta}_a \neq 0 \tag{6.4}$$

The test is read as no change in parameters in the set of θ_a versus change in parameters in the set of θ_a. To test the hypothesis, two tests can be performed, namely the sensitivity test and the minmax test. The sensitivity test assumes $\tilde{\theta}_b = 0$ while in the minmax test $\tilde{\theta}_b$ is treated as nuisance, i.e. $\tilde{\theta}_b$ is not necessarily zero.

Sensitivity test

To perform the sensitivity test, the matrix $M(\theta_0)$ is partitioned into two matrices M_a and M_b, i.e.

$$M(\theta_0) = [M_a \;\; M_b] \tag{6.5}$$

This partition corresponds to the partition of θ i.e. the column dimension of M_a is the same as the row dimension of θ_a and the column dimension of M_b is the same as the row dimension of θ_b. Based on this partition, the sensitivity test is written as

$$\tilde{\chi}_a^2 = \tilde{\xi}_a^T F_{aa}^{-1} \tilde{\xi}_a \tag{6.6}$$

where,

$$\tilde{\xi}_a = M_a^T \Sigma^{-1}(\theta_0) \xi_N(\theta_0) \tag{6.7}$$

$$F_{aa} = M_a^T \Sigma^{-1}(\theta_0) M_a \tag{6.8}$$

$\tilde{\chi}_a^2$ has a χ^2 distribution with n_a degrees of freedom.

Minmax test

The Minmax test is performed by defining

$$F = M^T(\theta_0)\Sigma^{-1}(\theta_0)M(\theta_0) \tag{6.9}$$

and then partitioning it into

$$F = \begin{bmatrix} F_{aa} & F_{ab} \\ F_{ba} & F_{bb} \end{bmatrix} = \begin{bmatrix} M_a^T\Sigma^{-1}(\theta_0)M_a & M_a^T\Sigma^{-1}(\theta_0)M_b \\ M_b^T\Sigma^{-1}(\theta_0)M_a & M_b^T\Sigma^{-1}(\theta_0)M_b \end{bmatrix} \tag{6.10}$$

Then defining

$$\tilde{\xi}_a = M_a^T \Sigma^{-1}(\theta_0) \xi_N(\theta_0) \tag{6.11}$$

$$\tilde{\xi}_b = M_b^T \Sigma^{-1}(\theta_0) \xi_N(\theta_0) \tag{6.12}$$

$$\xi_a^* = \tilde{\xi}_a - F_{ab} F_{bb}^{-1} \tilde{\xi}_b \tag{6.13}$$

$$F_a^* = F_{aa} - F_{ab} F_{bb}^{-1} F_{ba} \tag{6.14}$$

the test is written as

$$\chi_a^{2*} = \xi_a^{*T} F_a^{*-1} \xi_a^* \tag{6.15}$$

where, χ_a^{2*} has a χ^2 distribution with n_a degrees of freedom. By comparing χ_a^{2*} with a pre-specified threshold, it can be determined whether the parameters in θ_a have changed. In practice, as recommended by , minmax test for all possible sub-vector $\tilde{\theta}_a$ are performed and the one having the largest value indicates the non-zero sub-vector.

6.3 Validation of continuous-time models using the local approach

An important feature of the local approach is that the algorithm can be developed along the same line as model identification. Many established methods to enhance identification algorithms can be readily transferred to the local detection approach. The main task here is to find appropriate primary and improved residuals that are capable of detecting small changes while robust to varying disturbances.

Validation of discrete-time models has been reported in the literature and different aspects of the validation schemes have been investigated. Here we will present local approach for validation of continuous-time models. As mentioned earlier, the main task is to find the appropriate residuals. Now, the residual is dependent on the error that is minimized in the identification procedure. As commonly done, the error is defined as the equation error or the output error. Here, we will derive the expressions for residuals for both equation error and output error and study the performance of the resulting methods in terms of their efficiency in detection the changes in model parameters.

6.3.1 Equation Error (EE) approach

Consider a SISO system given by

$$\begin{aligned} Y(s) &= G(s)e^{-\delta s}U(S) + E(s) \\ G(s) &= \frac{B(s)}{A(s)} \end{aligned} \tag{6.16}$$

where s is the Laplace operator and $A(s)$ and $B(s)$ are polynomials in s and $G(s) = \frac{A(s)}{B(s)}$ is the process transfer function.

$$\begin{aligned} A(s) &= a_n s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0 \\ B(s) &= b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0; \; n \geqslant m \end{aligned}$$

The parameter vector for this process is defined as $\theta = [a_n \;\; a_{n-1} \cdots a_0 \;\; b_m \;\; b_{m-1} \cdots b_0 \;\; \delta]^T$. In EE based approaches, the error term is defined as

$$E_{EE}(s) = A(s)Y(s) - B(s)e^{-\delta s}U(S) \tag{6.17}$$

Evaluation of the above error involves differentiation of the input and output. However, it is a general practice in continuous-time identification to avoid direct differentiation of the noisy data. Rather, different linear dynamic (LD) operations are performed. In the linear filter approach, the differentiation operation is performed on filtered data. For the purpose of model validation, we also want to avoid direct differentiation and following the linear filter approach, we define the equation error as

$$\underline{E}_{EE}(s) = A(s)\underline{Y}(s) - B(s)e^{-\delta s}\underline{U}(S) \tag{6.18}$$

where, $\underline{Y}(s)$ and $\underline{U}(s)$ are the filtered output and input respectively. The filter may be defined in different ways as mentioned in chapter 2.

The cost function that is minimized for the parameter estimation is defined as

$$J_{EE} = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}\underline{e}_{EE}^{2}(t,\theta) \tag{6.19}$$

where, $\underline{e}_{EE}(t) = \mathfrak{L}^{-1}\underline{E}_{EE}(s)$ is the equation error expressed in the time domain. Now, the gradient of J_{EE} can be calculated as

$$\frac{\partial J_{EE}}{\partial\theta} = \frac{1}{N}\sum_{t=1}^{N}\phi(t)\underline{e}_{EE}(t,\theta) \tag{6.20}$$

where,

$$\phi(t) = \frac{\partial\underline{e}_{EE}(t,\theta)}{\partial\theta} \tag{6.21}$$

To obtain the expression for the gradient, we use the Laplace domain expression of the equation error. The expression for ϕ in the Laplace domain can be obtained as

$$\begin{aligned}\Phi(s) &= \frac{\partial}{\partial\theta}\left[A(s)\underline{Y}(s) - B(s)e^{-\delta s}\underline{U}(S)\right] \\ &= \left[s^{n}\underline{Y}(s)\cdots\underline{Y}(s) \quad -s^{m}e^{-\delta s}\underline{U}(s)\cdots -e^{-\delta s}\underline{U}(s) \;\; B(s)se^{-\delta s}\underline{U}(S)\right]^{T}\end{aligned}$$

Taking inverse Laplace transform, we get the time domain expression for ϕ

$$\phi(t) = \left[\underline{y}^{(n)}(t)\cdots\underline{y}(t) \;\; \underline{u}^{(m)}(t^{*})\cdots\underline{u}(t^{*}) \;\; \underline{u}_{B}^{(1)}(t^{*})\right] \tag{6.22}$$

where, $t^{*} = (t-\delta)$ and $\underline{u}_{B}(t) = L^{-1}\left[B(s)\underline{U}(s)\right]$.

The estimated parameter θ_0 must satisfy the following equation

$$\frac{1}{N}\sum_{t=1}^{N}\phi(t)\underline{e}_{EE}(t,\theta_0) = 0 \tag{6.23}$$

which gives the following results

$$\frac{1}{N}\sum_{t=1}^{N}\phi(t)\underline{e}_{EE}(t,\theta) = 0 \quad \text{if} \quad \theta = \theta_0$$

$$\frac{1}{N}\sum_{t=1}^{N}\phi(t)\underline{e}_{EE}(t,\theta) \neq 0 \quad \text{if} \quad \theta \in \omega(\theta_0)\backslash\theta_0$$

Hence, we find the primary residual for equation error model

$$\rho(\theta, x_t) = \phi(t)\underline{e}_{EE}(t,\theta)$$

Next, we define the improved residual

$$\xi_N(\theta) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} \phi(t) \underline{e}_{EE}(t,\theta) \tag{6.24}$$

The expression for $M(\theta_0)$ can be obtained as

$$\begin{aligned} M(\theta_0) &\approx \frac{1}{N} \sum_{t=1}^{N} \left(\frac{\partial}{\partial \theta} \rho(\theta, x_t) \bigg|_{\theta=\theta_0} \right) \\ &= \frac{1}{N} \sum_{t=1}^{N} \left(\frac{\partial}{\partial \theta} \left(\phi(t) e_{EE}(t,\theta) \right) \bigg|_{\theta=\theta_0} \right) \\ &= \frac{1}{N} \sum_{t=1}^{N} \phi(t,\theta) \phi^T(t,\theta) + \frac{1}{N} \sum_{t=1}^{N} \phi'(t,\theta) \underline{e}_{EE}(t,\theta) \end{aligned} \tag{6.25}$$

where,

$$\begin{aligned} \Phi'(s) &= \frac{\partial}{\partial \theta} \left[s^n \underline{Y}(s) \cdots \underline{Y}(s) \quad - s^m e^{-\delta s} \underline{U}(s) \cdots - e^{-\delta s} \underline{U}(s) \;\; B(s) s e^{-\delta s} \underline{U}(S) \right]^T \\ &= \begin{pmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & s^{m+1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & s \\ 0 & \cdots & 0 & s^{m+1} & \cdots & s & -B(s)s^2 \end{pmatrix} e^{-\delta s} \underline{U}(s) \end{aligned} \tag{6.26}$$

So,

$$\phi'(t) = \begin{pmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \underline{u}^{(m+1)}(t^*) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \underline{u}^{(1)}(t^*) \\ 0 & \cdots & 0 & \underline{u}^{(m+1)}(t^*) & \cdots & \underline{u}^{(1)}(t^*) & \underline{u}_B^{(2)}(t^*) \end{pmatrix} \tag{6.27}$$

6.3.2 Output Error (OE) approach

In OE based methods the parameters of an appropriately structured system model are chosen so that they minimize a suitably defined measure or norm of the error between the model output and the observed output of the system to be identified (Unbehauen and Rao 1987),i.e.

$$E_{OE}(s) = Y(s) - \frac{B(s)}{A(s)} e^{-\delta s} U(s) \tag{6.28}$$

where, $G(s) = \frac{B(s)}{A(s)}$is the estimated model. Now the goal of an OE algorithm is to minimize the following objective function

$$J_{OE} = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}e_{OE}^2(t,\theta) \tag{6.29}$$

The gradient of J_{OE} can be calculated as

$$\frac{\partial J_{OE}}{\partial\theta} = \frac{1}{N}\sum_{t=1}^{N}\varphi(t,\theta)e_{OE}(t,\theta) \tag{6.30}$$

where,

$$\varphi(t,\theta) = \frac{\partial e_{OE}(t,\theta)}{\partial\theta} \tag{6.31}$$

The estimated parameter θ_0 must satisfy the following equation

$$\frac{1}{N}\sum_{t=1}^{N}\varphi(t)e_{OE}(t,\theta_0) = 0 \tag{6.32}$$

to yield the following results:

$$\frac{1}{N}\sum_{t=1}^{N}\varphi(t)e_{OE}(t,\theta) = 0 \quad \text{if} \quad \theta = \theta_0$$
$$\frac{1}{N}\sum_{t=1}^{N}\varphi(t)e_{OE}(t,\theta) \neq 0 \quad \text{if} \quad \theta \in \omega(\theta_0)\backslash\theta_0$$

Hence, we find the primary residual for output error model as:

$$\rho(\theta, x_t) = \varphi(t)e_{OE}(t,\theta)$$

Next, we define the improved residual as:

$$\xi_N(\theta) = \frac{1}{\sqrt{N}}\sum_{t=1}^{N}\varphi(t)e_{OE}(t,\theta) \tag{6.33}$$

where

$$\begin{aligned}\varphi(s,\theta) &= -\frac{\partial}{\partial\theta}\frac{B(s)}{A(s)}e^{-\delta s}U(s)\\ &= \frac{e^{-\delta s}}{A(s)}\left[\frac{B(s)}{A(s)}s^nU(s)\cdots\frac{B(s)}{A(s)}U(s) \quad -s^mU(s)\cdots -U(s) \quad B(s)sU(s)\right]^T\end{aligned}$$

The time domain expression can be presented as

$$\varphi(t,\theta) = \left[\underline{\hat{y}}^{(n)}(t^*)\cdots\underline{\hat{y}}(t^*) \quad -\underline{u}^{(m)}(t^*)\cdots -\underline{u}(t^*) \quad \hat{y}^{(1)}(t^*)\right]^T \tag{6.34}$$

$$
\begin{aligned}
M(\theta) &\approx \frac{1}{N}\sum_{t=1}^{N}\left(\frac{\partial}{\partial\theta}\rho(\theta,x_t)\right) \\
&= \frac{1}{N}\sum_{t=1}^{N}\varphi(t,\theta)\varphi^T(t,\theta) + \frac{1}{N}\sum_{t=1}^{N}\varphi'(t,\theta)e(t,\theta)
\end{aligned}
\tag{6.35}
$$

$$
\begin{aligned}
\varphi'(s,\theta) &= \frac{\partial}{\partial\theta}\left(\frac{e^{-\delta s}}{A(s)}\left[\frac{B(s)}{A(s)}s^nU(s)\cdots\frac{B(s)}{A(s)}U(s) \quad -s^mU(s)\cdots - U(s) \;\; B(s)sU(s)\right]^T\right) \\
&= \begin{pmatrix}
\frac{-2B(s)s^{2n}}{[A(s)]^3} & \cdots & \frac{-2B(s)s^{n}}{[A(s)]^3} & \frac{s^{n+m}}{[A(s)]^2} & \cdots & \frac{s^{n}}{[A(s)]^2} & \frac{-B(s)s^{n+1}}{[A(s)]^2} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\frac{-2B(s)s^{n}}{[A(s)]^3} & \cdots & \frac{-2B(s)}{[A(s)]^3} & \frac{s^{m}}{[A(s)]^2} & \cdots & \frac{1}{[A(s)]^2} & \frac{-B(s)s}{[A(s)]^2} \\
\frac{s^{n+m}}{[A(s)]^2} & \cdots & \frac{s^{m}}{[A(s)]^2} & 0 & \cdots & 0 & \frac{s^{m+1}}{A(s)} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\frac{s^{n}}{[A(s)]^2} & \cdots & \frac{1}{[A(s)]^2} & 0 & \cdots & 0 & \frac{s}{A(s)} \\
\frac{-B(s)s^{n+1}}{[A(s)]^2} & \cdots & \frac{-B(s)s}{[A(s)]^2} & \frac{s^{m+1}}{A(s)} & \cdots & \frac{1}{A(s)} & \frac{-B(s)s^{2}}{A(s)}
\end{pmatrix} e^{-\delta s}U(s)
\end{aligned}
$$

$$
\varphi'(t,\theta) = \begin{pmatrix}
-2\underline{\underline{\hat{y}}}^{(2n)}(t^*) & \cdots & -2\underline{\underline{\hat{y}}}^{(n)}(t^*) & \underline{\underline{u}}^{(n+m)}(t^*) & \cdots & \underline{\underline{u}}^{(n)}(t^*) & -\underline{\hat{y}}^{(n+1)}(t^*) \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
-2\underline{\underline{\hat{y}}}^{(n)}(t^*) & \cdots & -2\underline{\underline{\hat{y}}}(t^*) & \underline{\underline{u}}^{(m)}(t^*) & \cdots & \underline{\underline{u}}(t^*) & -\underline{\hat{y}}^{(1)}(t^*) \\
\underline{\underline{u}}^{(n+m)}(t^*) & \cdots & \underline{\underline{u}}^{(m)}(t^*) & 0 & \cdots & 0 & \underline{u}^{(m+1)}(t^*) \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\underline{\underline{u}}^{(n)}(t^*) & \cdots & \underline{\underline{u}}(t^*) & 0 & \cdots & 0 & \underline{u}^{(1)}(t^*) \\
-\underline{\hat{y}}^{(n+1)}(t^*) & \cdots & -\underline{\hat{y}}^{(1)}(t^*) & \underline{u}^{(m+1)}(t^*) & \cdots & \underline{u}^{(1)}(t^*) & -\hat{y}^{(2)}(t^*)
\end{pmatrix}
$$

Here $t^* = (t-\delta)$, $\underline{u}^{(i)}(t) = \mathfrak{L}^{-1}\left[\frac{s^i}{A(s)}U(s)\right]$, $\underline{\underline{u}}^{(i)}(t) = \mathfrak{L}^{-1}\left[\frac{s^i}{[A(s)]^2}U(s)\right]$, and $\underline{\hat{y}}^{(i)}(t) = \mathfrak{L}^{-1}\left[\frac{s^i}{A(s)}\hat{Y}(s)\right]$,$\underline{\underline{\hat{y}}}^{(i)}(t) = \mathfrak{L}^{-1}\left[\frac{s^i}{[A(s)]^2}\hat{Y}(s)\right]$.

6.3.3 MISO model validation

This section describes how we can obtain the primary and improved residuals for MISO systems. Let us consider a MISO system given by the following model:

$$
Y(s) = G_1(s)e^{-\delta_1 s}U_1(s) + \cdots + G_k(s)e^{-\delta_k s}U_k(s) + V(s) \tag{6.36}
$$

$$
G_i(s) = \frac{B_i(s)}{A_i(s)} \qquad i = 1, 2\cdots k
$$

$$
A_i(s) = a_{i,n_i}s^{n_i} + a_{i,n_i-1}s^{n_i-1} + \cdots\cdots + a_{i,1}s + a_{i,0}
$$

Algorithm 6.1: Continuous-time SISO model validation algorithm.

(1) Error estimation: Estimate the error term using eqn(6.18) (for the EE method) or eqn(6.28) (for OE method).

(2) Error gradient: Obtain the vector representing the error gradient, ϕ(EE) using eqn(6.22) or φ (OE) using eqn(6.34).

(3) Residuals: Calculate the improved residuals $\xi_N(\theta_0)$ using eqn(6.24) (EE) or eqn(6.33) (OE).

(4) Residual gradient: Obtain the matrix ϕ' (eqn(6.27)) or φ' (eqn(6.36)) and calculate the residual gradient $M(\theta_0)$ using eqn(6.25) (EE) or eqn(6.35) (OE).

(5) Calculation of χ^2: Calculate the χ^2_{global} value using eqn(6.2) or eqn(6.3), whichever is appropriate.

(6) Detection test: Compare the χ^2_{global} value with a pre-specified threshold. If χ^2_{global} is smaller than the threshold, the model passes the validation. Otherwise conclude that the model is in error and proceed with the isolation test.

(6) Isolation test: Calculate $\tilde{\chi}^2_a$ for sensitivity test using eqn(6.6) for every possible subset of parameter $\tilde{\theta}_a$ and compare it with the appropriate threshold. If $\tilde{\chi}^2_a$ is larger than the threshold the corresponding $\tilde{\theta}_a$ is said to be in error. For the minmax test, calculate χ^{2*}_a using eqn(6.15) for every possible $\tilde{\theta}_a$ and the one having the greatest value is said to be in error.

$$B_i(s) = b_{i,m_i}s^{m_i} + b_{i,m_i-1}s^{m_i-1} + \cdots\cdots + b_{i,1}s + b_{i,0}$$

The parameter vector θ is given by

$$\theta = [a_{1,n_1} \cdots a_{1,0} \;\; b_{1,m_1} \cdots b_{1,0} \; \delta_1 \cdots\cdots a_{k,n_k} \cdots a_{k,0} \;\; b_{k,m_k} \cdots b_{k,0} \; \delta_k]^T$$

Defining

$$\theta^{(i)} = [a_{i,n_i} \cdots a_{i,0} \;\; b_{i,m_i} \cdots b_{i,0} \; \delta_k]^T$$
$$\theta = [\theta^{(1)^T} \theta^{(2)^T} \cdots \theta^{(k)^T}]^T$$

The goal of an OE algorithm is to minimize the following objective function

$$J = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}(y(t) - \hat{y}(t|\theta))^2 = \frac{1}{N}\sum_{t=1}^{N}\frac{1}{2}e^2(t,\theta) \tag{6.37}$$

where

$$\hat{Y}(s|\theta) = G_1(s)e^{-\delta_1 s}U_1(s) + \cdots + G_k(s)e^{-\delta_k s}U_k(s)$$

Defining

$$\hat{Y}_i(s|\theta^{(i)}) = G_i(s)e^{-\delta_i s}U_i(s)$$

We get

$$\hat{y}(t|\theta) = \hat{y}_1(t|\theta^{(1)}) + \hat{y}_2(t|\theta^{(2)}) + \cdots \hat{y}_k(t|\theta^{(k)})$$

The gradient of J can be calculated as

$$\frac{\partial J}{\partial \theta} = \frac{1}{N}\sum_{t=1}^{N}\psi(t,\theta)e(t,\theta) \tag{6.38}$$

where

$$\psi(t,\theta) = -\frac{\partial \hat{y}(t|\theta)}{\partial \theta} \tag{6.39}$$

The gradient of J is zero for $\theta = \theta_0$ and non-zero in the neighborhood of θ_0. So, we can define the primary and improved residuals as:

$$\rho(\theta, x_t) = \psi(t)e(t,\theta)$$

$$\xi_N(\theta) = \frac{1}{\sqrt{N}}\sum_{t=1}^{N}\psi(t)e(t,\theta)$$

where,

$$\psi(t,\theta) = -\frac{\partial \hat{y}(t|\theta)}{\partial \theta} = \begin{pmatrix} -\frac{\partial \hat{y}_1(t|\theta^{(1)})}{\partial \theta^{(1)}} \\ -\frac{\partial \hat{y}_2(t|\theta^{(2)})}{\partial \theta^{(2)}} \\ \vdots \\ -\frac{\partial \hat{y}_k(t|\theta^{(k)})}{\partial \theta^{(k)}} \end{pmatrix} = \begin{pmatrix} \psi_1(t,\theta^{(1)}) \\ \psi_2(t,\theta^{(2)}) \\ \vdots \\ \psi_k(t,\theta^{(k)}) \end{pmatrix} \tag{6.40}$$

where

$$\psi_i(t,\theta^{(i)}) = -\frac{\partial \hat{y}_i(t|\theta^{(i)})}{\partial \theta^{(i)}}$$

$\psi_i(t,\theta^{(i)})$ can be found from eqn(6.34)

$$\psi_i(t,\theta^{(i)}) = \left[\underline{\hat{y}}_i^{(n_i)}(t^{*i})\cdots\underline{\hat{y}}_i(t^{*i}) \quad -\underline{u}_i^{(m_i)}(t^{*i})\cdots-\underline{u}_i(t^{*i}) \;\; \hat{y}_i^{(1)}(t^{*i})\right]^T$$

where $t^{*i}=(t-\delta_i)$, $\underline{u}_i^{(j)}(t)=\mathfrak{L}^{-1}\left[\frac{s^j}{A_i(s)}U_i(s)\right]$ and $\underline{\hat{y}}_i^{(j)}(t|\theta^{(i)})=\mathfrak{L}^{-1}\left[\frac{s^j}{A_i(s)}\hat{Y}_i(s|\theta^{(i)})\right]$

$$\begin{aligned}\psi'(t,\theta) &= \frac{\partial\psi(t,\theta)}{\partial\theta}\\ &= \begin{pmatrix} \psi_1'(t,\theta^{(1)}) & 0 & \cdots & 0\\ 0 & \psi_2'(t,\theta^{(2)}) & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \cdots & \psi_k'(t,\theta^{(k)})\end{pmatrix}\end{aligned} \tag{6.41}$$

where

$$\psi_i'(t,\theta^{(i)}) = \frac{\partial\psi_i(t,\theta^{(i)})}{\partial\theta^{(i)}}$$

Following the derivation of eqn(6.36) we can find

$$\psi_i'(t,\theta^{(i)}) = \begin{pmatrix} -2\underline{\underline{\hat{y}_i}}^{(2n)}(t^*) & \cdots & -2\underline{\underline{\hat{y}_i}}^{(n)}(t^*) & \underline{\underline{u_i}}^{(n+m)}(t^*) & \cdots & \underline{\underline{u_i}}^{(n)}(t^*) & -\underline{\hat{y}_i}^{(n+1)}(t^*)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots\\ -2\underline{\underline{\hat{y}_i}}^{(n)}(t^*) & \cdots & -2\underline{\underline{\hat{y}_i}}(t^*) & \underline{\underline{u_i}}^{(m)}(t^*) & \cdots & \underline{\underline{u_i}}(t^*) & -\underline{\hat{y}_i}^{(1)}(t^*)\\ \underline{\underline{u_i}}^{(n+m)}(t^*) & \cdots & \underline{\underline{u_i}}^{(m)}(t^*) & 0 & \cdots & 0 & \underline{u_i}^{(m+1)}(t^*)\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots\\ \underline{\underline{u_i}}^{(n)}(t^*) & \cdots & \underline{\underline{u_i}}(t^*) & 0 & \cdots & 0 & \underline{u_i}^{(1)}(t^*)\\ -\underline{\hat{y}_i}^{(n+1)}(t^*) & \cdots & -\underline{\hat{y}_i}^{(1)}(t^*) & \underline{u_i}^{(m+1)}(t^*) & \cdots & \underline{u_i}^{(1)}(t^*) & -\hat{y}_i^{(2)}(t^*)\end{pmatrix}$$

Here we define $\underline{\underline{u}}_{i,j}(t) = \frac{p^j}{[A_i(p)]^2} u_i(t)$ and $\underline{\underline{\hat{y}}}_{i,j}(t|\theta^{(i)}) = \frac{p^j}{[A_i(p)]^2} \hat{y}_i(t|\theta^{(i)})$
We also get

$$
\begin{aligned}
M(\theta) &\approx \frac{1}{N}\sum_{t=1}^{N} \psi(t,\theta)\psi^T(t,\theta) + \frac{1}{N}\sum_{t=1}^{N}\psi'(t,\theta)e(t,\theta) \\
&= \frac{1}{N}\sum_{t=1}^{N} \begin{pmatrix} \psi_1(t,\theta^{(1)})\psi_1^T(t,\theta^{(1)}) & \psi_1(t,\theta^{(1)})\psi_2^T(t,\theta^{(2)}) & \cdots & \psi_1(t,\theta^{(1)})\psi_k^T(t,\theta^{(k)}) \\ \psi_2(t,\theta^{(2)})\psi_1^T(t,\theta^{(1)}) & \psi_2(t,\theta^{(2)})\psi_2^T(t,\theta^{(2)}) & \cdots & \psi_2(t,\theta^{(2)})\psi_k^T(t,\theta^{(k)}) \\ \vdots & \vdots & \vdots & \vdots \\ \psi_k(t,\theta^{(k)})\psi_1^T(t,\theta^{(1)}) & \psi_k(t,\theta^{(k)})\psi_2^T(t,\theta^{(2)}) & \cdots & \psi_k(t,\theta^{(k)})\psi_k^T(t,\theta^{(k)}) \end{pmatrix} \\
&= \frac{1}{N}\sum_{t=1}^{N} \begin{pmatrix} \psi_1'(t,\theta^{(1)})e(t,\theta) & 0 & \cdots & 0 \\ 0 & \psi_2'(t,\theta^{(2)})e(t,\theta) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \psi_k'(t,\theta^{(k)})e(t,\theta) \end{pmatrix} \\
&= \begin{pmatrix} M_1(\theta) & M_2(\theta) & \cdots & M_k(\theta) \end{pmatrix}
\end{aligned}
$$

where

$$
\begin{aligned}
M_1(\theta) &= \frac{1}{N}\sum_{t=1}^{N} \begin{pmatrix} \psi_1(t,\theta^{(1)})\psi_1^T(t,\theta^{(1)}) + \psi_1'(t,\theta^{(1)})e(t,\theta) \\ \psi_2(t,\theta^{(1)})\psi_1^T(t,\theta^{(1)}) \\ \vdots \\ \psi_k(t,\theta^{(k)})\psi_1^T(t,\theta^{(1)}) \end{pmatrix} \\
M_2(\theta) &= \frac{1}{N}\sum_{t=1}^{N} \begin{pmatrix} \psi_1(t,\theta^{(1)})\psi_2^T(t,\theta^{(2)}) \\ \psi_2(t,\theta^{(1)})\psi_1^T(t,\theta^{(1)}) + \psi_2'(t,\theta^{(2)})e(t,\theta) \\ \vdots \\ \psi_k(t,\theta^{(k)})\psi_2^T(t,\theta^{(2)}) \end{pmatrix} \\
M_k(\theta) &= \frac{1}{N}\sum_{t=1}^{N} \begin{pmatrix} \psi_1(t,\theta^{(1)})\psi_k^T(t,\theta^{(k)}) \\ \psi_2(t,\theta^{(1)})\psi_k^T(t,\theta^{(k)}) \\ \vdots \\ \psi_k(t,\theta^{(k)})\psi_k^T(t,\theta^{(k)}) + \psi_k'(t,\theta^{(k)})e(t,\theta) \end{pmatrix}
\end{aligned}
$$

6.4 Simulation study

In this section, we study the performance of the proposed model validation methods. The measure of performance used in this study are the detection rate and the isolation rate. These two quantities are calculated based on the results of 100 Monte-Carlo simulation runs. The detection rate for a particular parameter, expressed in percentage, is defined as the number of successful determination of the change out of the 100 runs. The isolation rate is defined as the percent of times a detected change is pointed towards the corresponding parameter successfully.

The following elements are considered as potential factors influencing the performance of any validation algorithm

- Degree of change in the parameters.
- Presence of noise.
- Data length.

It is obvious that a large change in parameters will be easily detected and isolated while a high NSR will inhibit detection and result in poor isolation. Now, for the sake of brevity, we will present the performance in terms of a ratio between change in parameters and the NSR, both expressed in percentage, which we will denote as $\frac{\Delta\theta}{NSR}$.

An important implementation issue for the local approach is the choice of threshold for the χ^2 test. From the application point of view, the selection of the threshold is as important as the generation of the residuals. Details on the effect of threshold based on simulation and real industrial data have been provided in (Cheng *et al.* 2003). Theoretical value of the threshold can be obtained from a standard χ^2 table for a specified false alarm rate and for the degrees of freedom of the corresponding problem. In real life implementations, it is a common practise to choose the threshold from a training data set due to noise characteristics. In our study we choose the threshold using the training data.

For the equation error approach, we need to choose a filter. For the current study we chose a filter of the form $\frac{\lambda}{(s+\lambda)^{n+1}}$, which, in the linear filter literature, is known as the normalized PMF filter.

6.4.1 A SISO example

To test the performance of the EE and OE methods and evaluate the effects of the above mentioned factors, we take the following process as example

$$G(s) = \frac{1}{20s+1}e^{-7s} \tag{6.42}$$

We will denote the parameter vector for this process as $\theta = [a \;\; b \;\; \delta] = [20 \;\; 1 \;\; 7]$. A RBS signal was used as input to the process. The sampling interval was chosen to be 1 second. A discrete-time white noise sequence with specific NSR was added to the noise-free output. A training set of 5000 data points were used to numerically evaluate the parameter $\Sigma(\theta_0)$ and also to determine the threshold. Test data of different lengths were used to estimate the χ^2 value and comparing it with the threshold, validity of the model is concluded. Once a change is detected, a minmax test is performed for the purpose of isolation. The threshold estimated from the training data based on 2% false alarm rate was found to be 12.7. The theoretical value found from the standard χ^2 table was found to be 9.9.

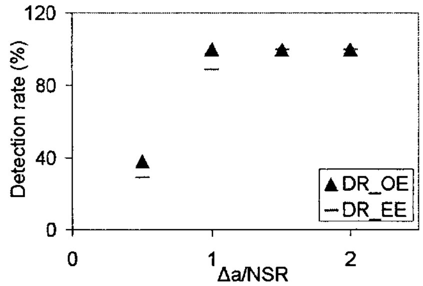
Figures 6.1 show the detection and isolation performance of the EE and OE methods for different value of the ratio $\frac{\Delta\theta}{NSR}$. 1000 data points were used for this test. It can be concluded from the detection and isolation results that any change in the gain parameter for this first order system is readily detected and isolated. The rate of detection and isolation for the time constant is also high. The methods are less sensitive to small changes in time delay. However, when we have either large change in parameters or the number of available data is more, both the detection and isolation is very close to 100% for all of the parameters including the delay.

Also it is demonstrated by the simulation results that the OE based algorithm shows better performance than the EE algorithm except for the case of time delay. For the EE method, the detection and isolation rates are slightly higher for small changes in the delay.

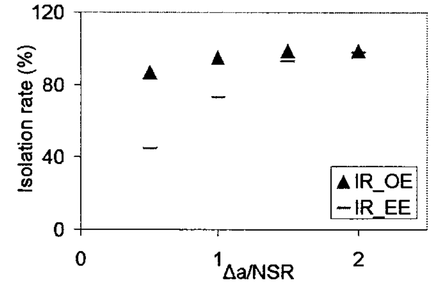
Figures 6.2 show the effect of available data points on the efficiency of the EE and OE method in detecting and isolating a change. As can be seen from the graphs, except for the case of isolation of the change in the time delay, for $\frac{\Delta\theta}{NSR} = 1$, using 1000 data points, changes in the parameters can be detected and isolated at a rate of almost 100%. For the detection and isolation of the delay more data points are needed.

Effect of noise

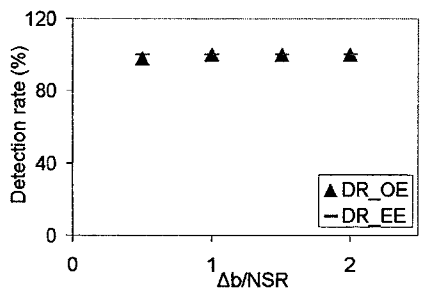
In this section, we present some results on the effect of noise level and noise dynamics on the performance of the EE and OE based methods. The performance is measured on the basis of the rate of false alarm. The false rate is defined as the detection rate when there is no change in the process model based on 100 Monte Carlo simulation runs.



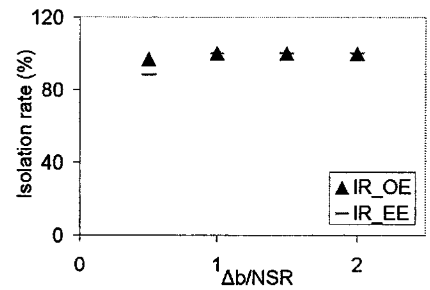
(a) Detection rate: a



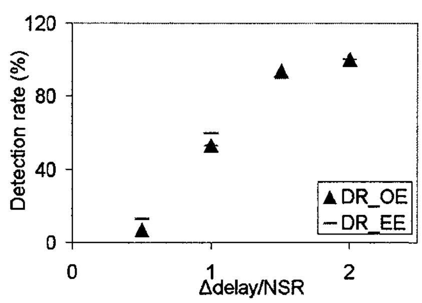
(b) Isolation rate: a



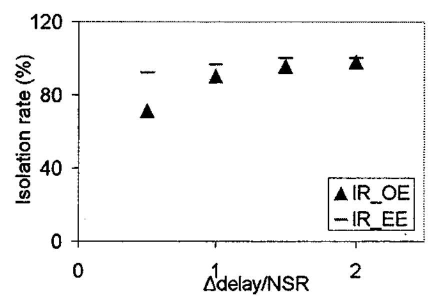
(c) Detection rate: b



(d) Isolation rate: b

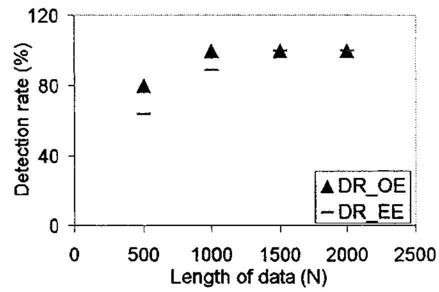


(e) Detection rate: δ

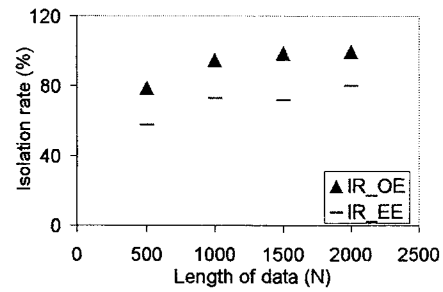


(f) Isolation rate: δ

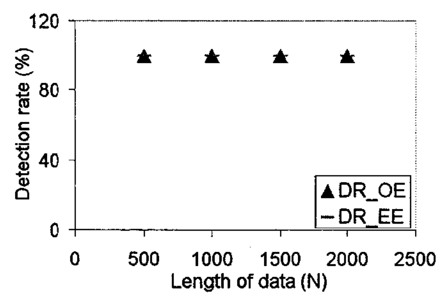
Figure 6.1: *Detection and isolation rate as a function of* $\frac{\Delta\theta}{NSR}$ *(N=1000) for the parameters of the SISO model.*



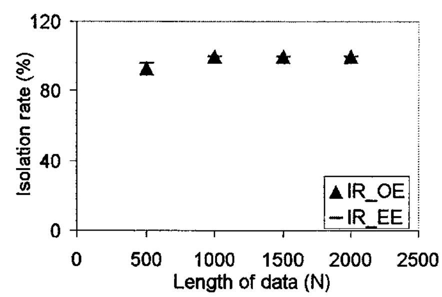
(a) Detection rate: a



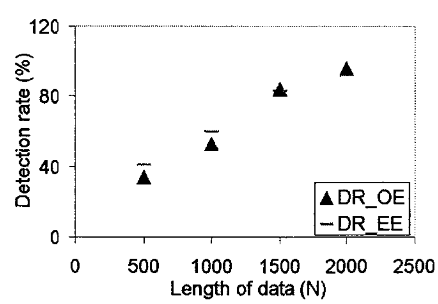
(b) Isolation rate: a



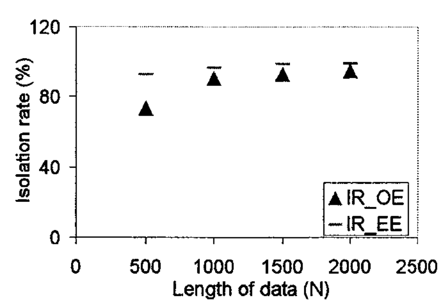
(c) Detection rate: b



(d) Isolation rate: b



(e) Detection rate: δ



(f) Isolation rate: δ

Figure 6.2: *Detection and isolation rate as a function of data length (N)*$(\frac{\Delta\theta}{NSR} = 1)$ *for the parameters of the SISO model.*

Table 6.1 shows the false alarm rate of the OE and EE method when the noise is white. The threshold was chosen numerically based on the χ^2 values obtained using the OE method and the training data and for a false rate or 2%.

Table 6.1: *False alarm rate for white noise.*

NSR	5%	10%	15%	20%
OE	2	2	2	2
EE	3	3	4	4

When colored noise sequences were added to the test data, for the threshold mentioned above, the false alarm rates of the OE and EE method were found to be very different. Table 6.2 shows the results of false rate for colored noise. The colored noise sequences were generated by filtering white noise sequences with the following discrete-time filter.

$$G(q^{-1}) = \frac{0.1}{1 - 0.9q^{-1}} \tag{6.43}$$

Table 6.2: *False alarm rate for colored noise.*

NSR	5%	10%	15%	20%
OE	0	0	0	0
EE	5	9	16	22

6.4.2 Multivariate model validation

A MIMO process can be considered as a set of MISO processes and in this section, we present simulation results on the validation performance of the proposed method using a MISO process having the following output equation.

$$Y(s) = \frac{1.25e^{-7s}}{20s+1}U_1(s) + \frac{1e^{-4s}}{30s+1}U_2(s) + E(s) \tag{6.44}$$

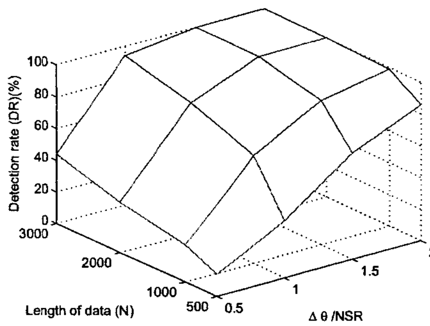
The parameter vector for this process is

$$\begin{aligned}\theta &= [a_{1,1}\ b_{1,1}\ \delta_{1,1}\ a_{2,1}\ b_{2,1}\ \delta_{2,1}]^T \\ &= [20\ \ 1.25\ \ 7\ \ 30\ \ 1\ \ 4]^T\end{aligned}$$

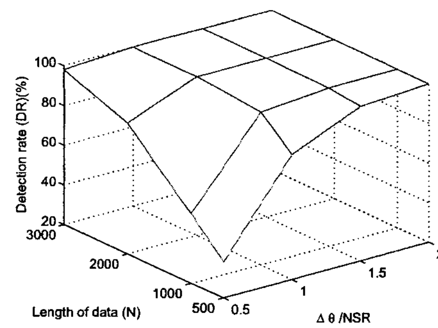
RBS signals were chosen for both u_1 and u_2. The sampling interval was 1 second. A random noise signal corresponding to a NSR of 10% was added with the noise free output to generate the noisy output signal. Results obtained from 100 Monte-Carlo simulation runs are presented in Figures 6.3 and Figures 6.4. The threshold was selected for this example according to a false alarm rate of 5%. The detection and isolation rates are near 100% for the gain parameters for $\frac{\Delta\theta}{NSR} \geq 1$ and $N \geq 100$. Although such a small change of other parameters cannot be detected and isolated at 100% rate, when either the change is larger or the length of available data points are more, the detection and isolation rate can reach as high as 100% for all parameters.

6.5 Concluding remarks

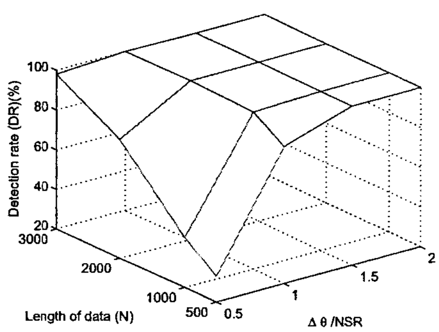
Local approach based model validation methods for discrete-time models have been studied by several researchers. We present here validation methods, also based on the local approach, for continuous-time models with time delay. The ability to detect change in the delay is a unique feature of the proposed algorithms. The different formulation, namely the equation error and output error formulation of the validation problem of single input single output models have been provided and comparative study based on simulation results have been presented. Comparatively better performance in detection and isolation and insensitivity to disturbance give the output error based method preference over the equation error method. Also validation method for multiple input single output models have been presented with simulation results. The proposed methods can efficiently detect and isolate changes in different parameters provided that the change is significant compared to the noise level and an informative data set is available.



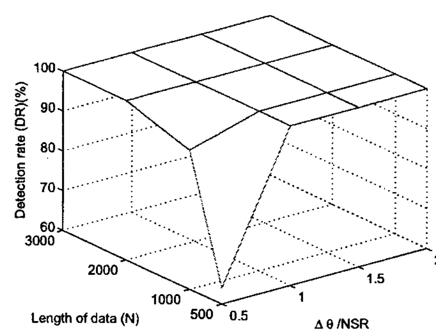
(a) $a_{1,1}$



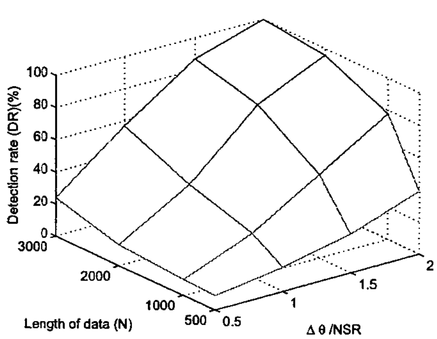
(b) $a_{2,1}$



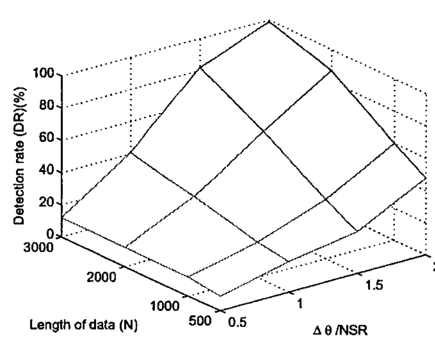
(c) $b_{1,1}$



(d) $b_{2,1}$

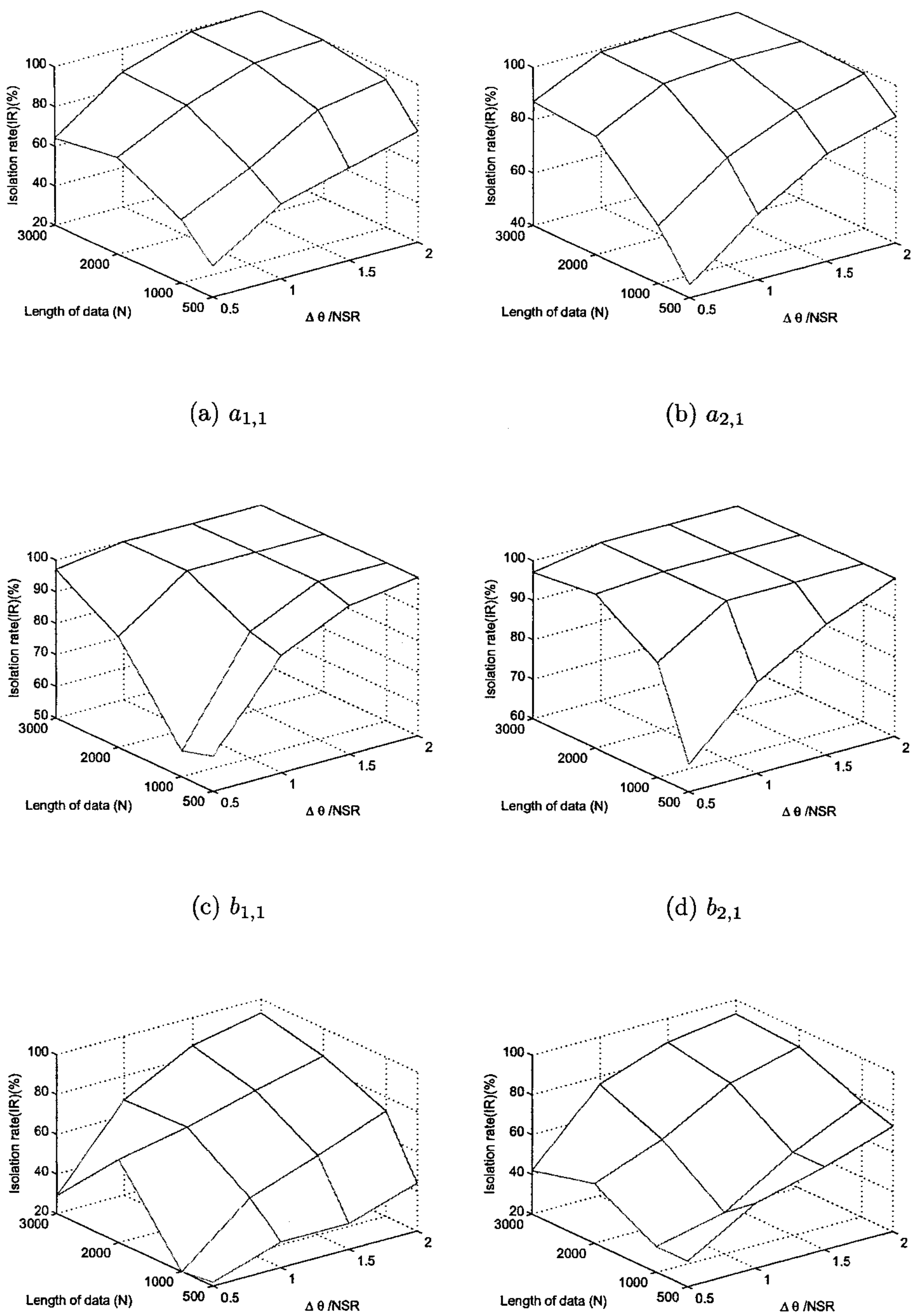


(e) $\delta_{1,1}$



(f) $\delta_{2,1}$

Figure 6.3: *Detection rate as a function of data length (N) and* $\frac{\Delta\theta}{NSR}$ *for the parameters of the MISO model.*



(a) $a_{1,1}$ (b) $a_{2,1}$

(c) $b_{1,1}$ (d) $b_{2,1}$

(e) $\delta_{1,1}$ (f) $\delta_{2,1}$

Figure 6.4: *Isolation rate as a function of data length (N) and $\frac{\Delta\theta}{NSR}$ for the parameters of the MISO model.*

Chapter 7

Conclusions

7.1 Contributions of this thesis

The main contributions of this thesis can be summarized as follows:

- A new linear filter method has been proposed for simultaneous estimation of the time delay and parameters of continuous-time transfer function models. In order to implement the idea of bringing the delay term within the parameter vector of an estimation equation, a novel structure of a linear filter is proposed that facilitates the estimation of the delay and model parameters following an iterative procedure. The method can estimate fractional time delays and is applicable regardless of the order of the model.

- In an effort to meet the requirements of the application world, methods tailored to cope with industrial data have been proposed. A simple procedure based on the idea of iterative prediction has been presented for identification of continuous-time models from irregularly sampled output data.

- Novel methods for identification from both open loop and closed loop step response data have been presented. The methods do not require data in their deviation forms. Consequently, raw industrial data without the required preprocessing can be used directly for identification.

- A continuous-time multiple input multiple output (MIMO) model identification method is introduced that uses the idea of decomposition of the MIMO data matrix into its single input single output (SISO) equivalents. The decomposition allows the use of SISO identification methods for the identification of MIMO models while the requirements for the MIMO test procedure are met.

- Validation techniques for continuous time SISO and MIMO/MISO models are presented. Ability to detect and isolate change in the delay term of the models is a unique feature of the proposed model validation method.

7.2 Recommendations for future work

Research initiatives on the topic of the current research and the following related fields are worthy of further investigation:

- In this thesis, simultaneous time delay and parameter estimation problem has been considered within the frameworks of linear filter methods for continuous-time identification. Other approaches of continuous-time identification may also be explored to search for development of more convenient, simpler, yet numerically robust methods.

- Several methods proposed in this thesis involves iterative solutions. However, the theoretical analysis of the convergence of the iterative algorithms has not been addressed. Initiatives can be taken to analyze the theory of the converging phenomena of the iterative schemes which may also provide more insights of the algorithms.

- Fixed point iteration methods have been used in this work for the iterative solutions of the parameter estimation equations. It is worthwhile to investigate how the statistical properties of the estimated parameters can be obtained theoretically. Also the possibility of obtaining computationally less intensive, numerically more robust and more accurate algorithms can be investigated.

- The multiple input multiple output model identification method presented in this thesis requires that N_u (number of inputs) number of experiments to be performed and the tests should start from steady state conditions of the outputs. It is worthwhile to investigate whether the requirement of initial steady state conditions can be relaxed. This will facilitate the use of data with non-zero initial conditions. But more importantly, data from a single test would be considered enough as one would be able to use different segments of data as representatives of different experiments.

Bibliography

Agrawal, P. and H. C. Lim (1986). Analysis of various control schemes for continuous bioreactors. *Advances in Biochemical Engineering/Biotechnology* **30**, 61–90.

Ahmed, S., B. Huang and S. L. Shah (2006). Parameter and delay estimation of continuous-time models using a linear filter. *Journal of Process Control* **16**(4), 323–331.

Ananth, I. and M. Chidambaram (1999). Closed-loop identification of transfer function model for unstable systems. *Journal of the Franklin Institute* **336**, 1055–1061.

Ba Hli, F. (1954). A general method for time domain network synthesis. *IRE Transactions - Circuit Theory* **1**(3), 21–28.

Basseville, M. (1998). On-board component fault detection and isolation using statistical local approach. *Automatica* **34**(11), 1391–1415.

Basseville, M. and I. V. Nikiforov (1993). *Detection of Abrupt Changes-Theory and Applications.* Prentice-Hall Information and System Science Series. Englewood Cliffs, NJ, USA.

Bastogne, T., H. Garnier and P. Sibille (2001). A PMF-based subspace method for continuous-time model identification. Application to a multivariable winding process. *International Journal of Control* **74**(2), 118–132.

Benveniste, A., M. Basseville and G. V. Moustakides (1987). The asymptotic local approach to change detection and model validation. *IEEE Transactions on Automatic Control* **32**(7), 583–592.

Bi, Q., W. Cai, E. Lee, Q. G. Wang, C. C. Hang and Y. Zhang (1999). Robust identification of first-order plus dead-time model from step response. *Control Engineering Practice* **7**, 71–77.

Björklund, S. (2003). Experimental evaluation of some thresholding methods for estimating time-delays in open-loop. Technical Report - LiTH-ISY-R-2525. Department of Electrical Engineering, Linköping University. Linköping, Sweden.

Björklund, S. (2003). A survey and comparison of time-delay estimation methods in linear systems. Technical Report - Licentiate Thesis no. 1061. Department of Electrical Engineering, Linköping University. Linköping, Sweden.

Box, G. E. P. and G. M. Jenkins (1976). *Time series analysis forecasting and control.* Holden-Day. San Fansisco, USA.

Carlemalm, C., S. Halvarsson, T. Wigren and B. Wahlberg (1999). Algorithm for time delay estimation using a low complexity exhaustive search. *IEEE Transactions on Automatic Control* **44**(5), 1031–1037.

Carter, G. C. (1993). *Coherence and time delay estimation - An Applied Tutorial for Research, Development, Test and Evaluation Engineers.* IEEE Press. New York, USA.

Chen, C. L. (1989). A simple method for online identification and controller tuning. *AIChE Journal* **35**, 2037.

Cheng, L., K. E. Kwok, B. Huang and P. Li (2003). Improved threshold for the local approach in detecting faults. *Industrial & Engineering Chemistry Research* **42**, 1870–1878.

Coelho, F. S. and P. R. Barros (2003). Continuous-time identification of first-order plus dead-time models from step response in closed loop. *Proc. 13th* IFAC *symposium on System Identification, Rotterdam, the Netherlands, Auguat 2003* pp. 413–418.

Diamessis, J. E. (1965). A new method for determining the parameters of physical systems. *Proceedings of the IEEE, February 1965* pp. 205–206.

Doma, M. J., P. A. Taylor and P. J. Vermeer (1996). Closed loop identification of MPC models for MIMO processes using genetic algorithm and dithering one variable at a time: Application to an industrial distillation tower. *Computers & Chemical Engineering* **20**(suppl.), 1035–1040.

Fairman, F. W. (1971). Parameter estimation for a class of multivariate nonlinear processes. *International Journal of Control* **1**(3), 291–296.

Ferretti, G., C. Maffezzoni and R. Scattolini (1991). Recursive estimation of time delay in sampled systems. *Automatica* **27**(4), 653–661.

Fischer, B. R. and A. Medvedev (1999). Laguerre domain estimation of time delays in narrowband ultrasonic echos. *Proc. 14th IFAC World Congress, Beijing, PRC, July 1999* pp. 361–366.

Forssell, U. and L. Ljung (1999). Closed-loop identification revisited. *Automatica* **35**, 1215–1241.

Garnier, H. (2002). Continuous-time model identification of real-life processes with the CONTSID toolbox. *Proc. 15th IFAC Triennial World Congress, Barcelona, Spain, July 2002.*

Garnier, H., M. Gilson and W. X. Zheng (2000). A bias-eliminated least-squares method for continuous-time model identification of closed-loop systems. *International Journal of Control* **73**(1), 38–48.

Garnier, H., M. Mensler and A. Richard (2003). Continuous-time model identification from sampled data: Implementation and performance evaluation. *International Journal of Control* **76**(13), 1337–1357.

Gevers, M. (2003). A personal view on the development of system identification. *Proc. 13th IFAC Symposium on System Identification, Rotterdam, the Netherlands, August 2003* pp. 773–784.

Gevers, M., X. Bombois, B. Codrons, G. Scorletti and B. Anderson (2003). Model validation for control and controller validation in a prediction error framework - Part I: Theory. *Automatica* **39**, 403–415.

Gillberg, J. (2004). Methods for frequency domain estimation of continuous-time models. Technical Report - Licentiate Thesis no. 1133. Department of Electrical Engineering, Linköping University. Linköping, Sweden.

Gillberg, J. and F. Gustafsson (2005). Frequency-domain continuous-time AR modeling using non-uniformly sampled measurements. In: *IEEE Conference on Acoustics, Speech and Signal Processing, Philadelphia, USA, March 2005*. Vol. 4. pp. 105 – 108.

Gillberg, J. and L. Ljung (2004). Frequency domain identification of continuous-time ARMA models: Interpolation and non-uniform sampling. Technical Report - LiTH-ISY-R-2625. Department of Electrical Engineering, Linköping University. Linköping, Sweden.

Gillberg, J. and L. Ljung (2006). Frequency-domain identification of continuous-time output error models from nonuniformly sampled data. In: *Proc. 14th IFAC Symposium on System Identification, Newcastle, Australia, March 2006.*

Gilson, M. and H. Garnier (2003). Continuous-time model identification of systems operating in closed loop. *Proc. 13th IFAC Symposium on System Identification, Rotterdam, the Netherlands, August 2003* pp. 425–430.

Gilson, M. and P. Van Den Hof (2001). On the relation between a bias-eliminated least-squares (BELS) and IV estimator in closed-loop identification. *Automatica* **37**(10), 1593–1600.

Goodwin, G. C. and J. S. Welsh (2002). Bias issue in closed loop identification with application to adaptive control. *Communications in Information and Systems* **2**(4), 349–370.

Grennberg, A. and M. Sandell (1994). Estimation of subsample time delay differences in narrowband ultrasonic echos using the Hilbert transformation. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control* **41**(5), 588–595.

Gustavsson, I. (1973). Survey of applications of identification in chemical and physical processes. *Proc. 3rd IFAC Symposium, the Hague/Delft, the Netherlands* pp. 67–85.

Gustavsson, I., L. Ljung and T. Soderstrom (1977). Identification of processes in closed loop. *Automatica* **3**, 59–75.

Hero, A., H. Messer, J. Goldberg, D. J. Thomson, M. G. Amin, G. Giannakis, A. Swami, J. K. Tugnait, A. Nehorai, A. L. Swindlehurst, J. F. Cardoso, Lang T. and J. Krolik (1998). Highlights of statistical signal and array processing. *IEEE Signal Processing Magazine* **15**(5), 21–64.

Hjalmarsson, H., M. Gevers and F. de Bruyne (1996). For model-based control design, closed-loop identification gives better performance. *Automatica* **32**(12), 1659–1673.

Homssi, L. and A. Tilti (1991). Continuous-time process identification: comparison of eight methods and particular aspects. *Proc. 9th IFAC/IFORS Symposium on System Identification, Budapest, Hungary* pp. 1634–1642.

Houghton, A. W. and C. D. Reeve (1995). Direct finding on spread spectrum signals using the time domain filtered cross spectral density. *IEE Proceedings - Radar, Sonar and Navigation* **144**(6), 315–320.

Hsia, T. C. (1977). *System Identification.* Lexington Books. Toronto.

Huang, B. (2000). Multivariate model validation in the presence of timevariant disturbance dynamics. *Chemical Engineeering Science* **55**, 4583–4595.

Huang, B. (2001). On-line closed-loop model validation and detection of abrupt parameter changes. *Journal of Process Control* **11**, 699–715.

Huang, B., A. Malhotra and E. C. Tamayo (2003*a*). Model predictive control relevant identification and validation. *Chemical Engineeering Science* **58**, 2389–2401.

Huang, B. and E. C. Tamayo (2000). Model validation for industrial model predictive control systems. *Chemical Engineeering Science* **55**, 2315–2327.

Huang, C. and M. Huang (1993). Estimation of the second-order parameters from the process transients by simple calculation. *Industrial & Engineering Chemistry Research* **32**, 228–230.

Huang, C. and W. C. Clement (1982). Parameter estimation for the second-order-plus-dead-time model. *Industrial & Engineering Chemistry Process Design Development* **21**, 601–603.

Huang, H., J. Jeng, C. Chiang and W. Pan (2003*b*). A direct method for multi-loop PI/PID controller design. *Journal of Process Control* **13**, 769–786.

Huselstein, E. and H. Garnier (2002). An approach to continuous-time model identification from non-uniformly sampled data. *Proc. 41st IEEE Conference on Decision and Control, Las Vegas, USA, December 2002* pp. 622–623.

Hwang, S. and S. Lai (2004). Use of two-stage least-squares algorithms for identification of continuous systems with time delay based on pulse response. *Automatica* **40**, 1591–1568.

Imtiaz, S. A., S. L. Shah and S. Narasimhan (2004). Missing data treatment using IPCA and data reconciliation. *Proc. 7th IFAC Symposium on Dynamics and Control of Process Systems, Boston, USA, July 2004.*

Ingimundarson, A. (2003). Dead-time compensation and performance monitoring in process control. PhD thesis. Dep. Automatic Control, Lund Institute of Technology. Lund, Sweden.

Ingimundarson, A. and T. Hagglund (2000). Closed-loop identification of a first-oder plus dead-time model with method of moments. *Proc. IFAC International Symposium on Advanced Control of Chemical Processes, Pisa, Italy.*

Ingimundarson, A. and T. Hagglund (2001). Robust tuning procedure of dead-time compensating controllers. *Control Engineering Practice* **9**, 1195–1208.

Isaksson, A. J. (1993). Identification of ARX-models subject to missing data. *IEEE Transactions on Automatic Control* **38**(5), 813–819.

Isaksson, A. J., A. Horch and G. A. Dumont (2001). Event-triggered deadtime estimation from closed-loop data. *Proc. American Control Conference, Arlington, USA, June 2001* pp. 387–404.

Isaksson, M. (1997). A comparison of some approaches to time delay estimation. Master's thesis. Dep. Automatic Control, Lund Institute of Technology. Lund, Sweden.

Jutan, A. and E. S. RodriguezII (1984). Extension of a new method for on-line controller tuning. *The Canadian Journal of Chemical Engineeering* **62**, 802–807.

Karimi, A. and I. D. Landau (1998). Comparison of closed-loop identification methods in terms of the bias distribution. *System & Control Letters* **34**, 159–167.

Kavdia, M. and M. Chidambaram (1996). On-line controller tuning for unstable systems. *Computers & Chemical Engineering* **20**, 301–305.

Küpfmüller, K. (1928). Uber die dynamik der selbsttatigen verstarkungsregler. *ENT* **5**, 459–467.

Kurz, H. and W. Goedecke (1981). Digital parameter adaptive control of processes with unknown dead time. *Automatica* **17**, 245–252.

Larsson, E. K. and T. Söderström (2002). Identification of continuous-time AR processes from unevenly sampled data. *Automatica* **38**(4), 709–718.

Latour, P., L. B. Koppel and D. R. Coughanowr (1967). Time-optimum control of chemical processes for set-point changes. *Industrial & Engineering Chemistry Process Design and Development* **6**, 452–460.

Lee, J. (1989). On-line PID controller tuning from a single, closed-loop test. *AIChE Journal* **35**(2), 329–331.

Lee, J., W. Cho and T.F. Edgar (1990). An improved technique for PID controller tuning from closed-loop test. *AIChE Journal* **36**(12), 1891–1895.

Lee, Y. W. (1932). Synthesis of electrical networks by means of Fourier transform of Laguerre functions. *Journal of Mathematical Physics* **11**, 83–113.

Li, D., S. L. Shah and T. Chen (2001). Identification of fast-rate models from multirate data. *International Journal of Control* **74**(7), 680–689.

Li, S. Y., W. J. Cai, H. Mei and Q. Xiong (2005). Robust decentralized parameter identification for two-input two-output processes from closed-loop step responses. *Control Engineering Practice* **13**, 519–531.

Ljung, L. (1999). *System Identification*. 2nd ed.. Prentice Hall.

Ljung, L and L. Guo (1997). The role of model validation for assessing the size of the unmodeled dynamics. *IEEE Transactions on Automatic Control* **42**(9), 1230–1239.

Makila, P. M. (1990). Approximation of stable systems by Laguerre filters. *Automatica* **26**(2), 333–345.

Mao, K. Z. and S. A. Billings (2000). Multi-dimensional model validity tests for nonlinear system identification. *International Journal of Control* **73**(2), 132–143.

Mehta, A. A. (1996). Identification of Multivariable, Linear and Nonlinear, Continuous-time Systems from Discrete Input-Output Measurements. PhD thesis. Rensselaer Polytechnique Institute. New York, USA.

Mei, H., S. Li, W. Cai and Q. Xiong (1992). Decentralized closed-loop parameter identification for multivariable processes from step responses. *Industrial & Engineering Chemistry Research* **31**, 274–281.

Melo, D. L. and J. C. Friedly (1992). On-line, closed-loop identification of multivariable systems. *Industrial & Engineering Chemistry Research* **31**, 274–281.

Milinkovic, S. A. (1997). Determination of the glass fiber freezing point by time-delay estimation. *Control Engineering Practice* **5**(7), 943–950.

Missailidis, P. K. (2000). System Identification of Multiple-input Multiple-output Continuous-time Systems using Walsh Functions. PhD thesis. Rensselaer Polytechnique Institute. New York, USA.

Nishikawa, Y., N. Sannomiya, T. Ohta and H. Tanaka (1990). A method for auto-tuning PID control parameters. *Automatica* **20**(3), 321–332.

O'Dwyer, A. (1999). A classification of techniques for the estimation of the model parameters of a time delayed process. In: *Irish Signals and Systems Conference.* National University of Ireland, Galway, Ireland.

O'Dwyer, A. (2000). A survey of techniques for the estimation and compensation of processes with time delay. Technical Report - AOD.00.03. Dublin Institute of Technology. Kelvin St., Dublin, Ireland.

Oldenbourg, R. C. and H. Sartorius (1948). *The Dynamics of Automatic Control.* The American Society of Mechanical Engineers. New York, USA.

Parington, J. R. (1991). Approximation of delay systems by Fourier-Laguerre series. *Automatica* **27**(3), 569–572.

Pintelon, R. and J. Schoukens (1999). Identification of continuous-time systems with missing data. *IEEE Transactions on Instrumentation and Measurement* **48**(3), 736–740.

Preisig, H. A. and D. W. T. Rippin (1993*a*). Theory and application of the modulating function method - I. Review and theory of the method and theory of spline-type modulating functions. *Computers & Chemical Engineering* **17**(1), 1–16.

Preisig, H. A. and D. W. T. Rippin (1993*b*). Theory and application of the modulating function method - II. Algebraic representation of Maletinskys spline-type modulating functions. *Computers & Chemical Engineering* **17**(1), 17–28.

Preisig, H. A. and D. W. T. Rippin (1993*c*). Theory and application of the modulating function method - II. Application to industrial process, a well-stirred tank reactor. *Computers & Chemical Engineering* **17**(1), 29–39.

Raghavan, H. (2004). Quantitative approaches for fault detection and diagnosis in process industries. PhD thesis. Dep. Chemical & Materials Engineering, University of Alberta. Edmonton, Canada.

Raghavan, H., A. Tangirala, R. B. Gopaluni and S. L. Shah (2005). Identification of chemical processes with irregular output sampling. *Control Engineering Practice* **14**(5), 467–480.

Rake, H. (1980). Step response and frequency response methods. *Automatica* **16**, 519–526.

Rangaiah, G. P. and P. R. Krishnaswamy (1994). Estimating second-order plus dead time model parameters. *Industrial & Engineering Chemistry Research* **33**, 1867–1871.

Rao, G. P. and H. Garnier (2002). Numerical illustrations of the relevance of direct continuous-time model identification. *Proc. 15th IFAC Triennial Congress, Barcelona, Spain, July 2002.*

Rao, G. P. and L. Sivakumar (1976). Identification of deterministic time-lag systems. *IEEE Transactions on Automatic Control* **21**, 527–529.

Rao, G.P. and H. Unbehauen (2006). Identification of continuous-time systems. *IEE Proc.- Control Theory & Applications* **153**(2), 185–220.

Roy, B. K., V. N. Bapat and D. C. Saha (1991). Effect of Poisson filter constant on the estimation of continuous-time models. *IEE Proceedings-D* **138**(6), 586–592.

Sagara, S. and Z. Y. Zhao (1990). Numerical integration approach to on-line identification of continuous-time systems. *Automatica* **26**(1), 63–74.

Sagara, S. and Z. Y. Zhao (1991). *Identification of continuous-time systems: Methodology and computer implementation.* Chap. Application of Digital Filtering Techniques, pp. 291–325. Kluwer Academic Publishers.

Saha, D. C. and G. P. Rao (1983). *Identification of Continuous Dynamical Systems - The Poisson Moment Functional Approach.* first ed.. Springer-Verlag.

Schleck, J. R. and D. Hanesian (1978). An evaluation of the Smith linear predictor technique for controlling deadtime dominated processes. *ISA Transaction* **17**(4), 39–46.

Seborg, D. E., T. F. Edgar and D. A. Mellichamp (1989). *Process Dynamics and Control.* John Wiley & Sons.

Shinskey, F. G. (1967). *Process Control Systems.* McGraw-Hill Book Company.

Sinha, N. K. and G. P. Rao (1991). *Identification of continuous-time systems: Methodology and computer implementation.* Kluwer Academic Publishers.

Smith, C. A. and A. B. Corripio (1985). *Principles and Practice of Automatic Process Control.* John Wiley & Sons.

Soderstrom, T. and P. Stoica (1989). *System Identification.* Prentice-Hall, Englewood Cliffs, NJ, USA.

Steiglitz, K. and L. E. McBride (1965). A technique for identification of linear systems. *IEEE Transactions on Automatic Control* **10**(4), 461–464.

Strejc, V. (1959). Approximation aperiodisscher ubertragungscharakteristiken. *Regelungstechnik* **7**, 124–128.

Sundaresan, K. R., C. C. Prasad and P. R. Krishnaswamy (1978). Evaluating parameters from process transients. *Industrial & Engineering Chemistry Process Design and Development* **17**(3), 237–241.

Tahl-Larsen, H. (1956). Frequency response from experimental non-oscillatory transient-response data. *Transactions of the ASME, Part II* **74**, 109–114.

Taiwo, O. (1993). Comparison of four methods of on-line identification and controller tuning. *IEE Proc.-D: Control Theory and Applications* **140**(5), 323–327.

Åström, K. J. and T. Hägglund (1995). *PID Controllers: Theory, Design and Tuning.* 2nd ed.. Instrument Society of America. North Carolina.

Tuch, J., A. Feuer and Z. J. Palmor (1994). Time delay estimation of continuous linear time-invariant systems. *IEEE Transactions on Automatic Control* **39**(4), 823–827.

Unbehauen, H. and G. P. Rao (1987). *Identification of Continuous Systems.* first ed.. North-Holland.

Unbehauen, H. and G. P. Rao (1990). Continuous-time approaches to system identification- A survey. *Automatica* **26**(1), 23–35.

Unbehauen, H. and G. P. Rao (1998). A review of identification in continuous-time systems. *Annual Reviews in Control* **22**, 145–171.

Valstar, J. E. (1963). In flight dynamic checkout. *IEEE Transactions on Aerospace* **1**(2), 213–220.

Van den Hof, P. (1998). Closed-loop issues in system identification. *Annual Reviews in Control* **22**, 173–186.

Viswanathan, P. K. and G. P. Rangaiah (2000). Process identification from closed-loop response using optimization methods. *Trans. IChemE - Part A* **78**, 528–541.

Wang, J., T. Chen and B. Huang (2004*a*). Multi-rate sampled data systems: computing fast models. *Journal of Process Control* **14**(1), 79–88.

Wang, L. and P. Gawthrop (2001). On the estimation of continuous-time transfer functions. *International Journal of Control* **74**(9), 889–904.

Wang, L. and W. R. Cluett (1995). Building transfer function models from noisy step response data using Laguerre network. *Chemical Engineering Science* **50**(1), 149–161.

Wang, L., P. Gawthrop, C. Chessari, T. Podsiadly and A. Giles (2004*b*). Indirect approach to continuous time system identification of food extruder. *Journal of Process Control* **14**, 603–615.

Wang, Q. G. and Y. Zhang (2001*a*). A novel FFT-based robust multivariate process identification method. *Industrial & Engineering Chemistry Research* **40**, 2485–2494.

Wang, Q. G. and Y. Zhang (2001*b*). Robust identification of continuous systems with dead-time from step response. *Automatica* **37**(3), 377–390.

Wang, Q. G., B. Zou, T. H. Lee and Q. Bi (1997). Auto-tuning of multivariate PID controllers from decentralized relay feedback. *Automatica* **33**(3), 319–330.

Whitfield, A. H. and N. Messali (1987). Integral-equation approach to system identification. *International Journal of Control* **45**(4), 1431–1445.

Wiener, N. (1956). *Modern mathematics for the engineers.* Chap. The theory of prediction. McGraw-Hill. Scarborough, CA.

Wood, R. K. and M. W. Berry (1973). Terminal composition control of a binary distillation column. *Chemical Engineering Science* **28**, 1707–1717.

Young, P. C. (1964). In flight dynamic checkout-A discussion. *IEEE Transactions on Aerospace* **2**, 1106–1111.

Young, P. C. (1970). An instrumental variable method for real time identification of noisy process. *Automatica* **6**, 271–287.

Young, P. C. (1981). Parameter estimation of continuous time models- A survey. *Automatica* **17**(1), 23–39.

Young, P. C. (2002). Optimal IV identification and estimation of continuous-time TF models. *Proc. 15th IFAC Triennial World Congress, Barcelona, Spain, July 2002* pp. 337–358.

Young, P. C. and A. Jakeman (1980). Refined instrumental variable method of recursive time-series analysis: Part III-Extensions. *International Journal of Control* **31**(4), 741–764.

Young, P. C., H. Garnier and A. Jarvis (2003). The identification of continuous-time linear and nonlinear models: A tutorial with environmental applications. *Proc. 13th IFAC Symposium on System Identification, Rotterdam, the Netherlands, August 2003* pp. 618–628.

Yuwana, M. and D. E. Seborg (1982). A new method for on-line controller tuning. *AIChE Journal* **28**(3), 434–440.

Zak, D. E., R. K. Pearson, R. Vadigepalli, G. E. Gonye, J. S. Schwaber and F. J. Doyle II (2003). Continuous-time identification of gene expression models. *OMICS A Journal of Integrative Biology* **7**(4), 373–386.

Zhang, Q., M. Basseville and A. Benveniste (1998). Fault detection and isolation in nonlinear dynamic systems: A combined input-output and local approach. *Automatica* **34**(11), 1359–1373.

Zhu, Y. (1998). Multivariate process identification for MPC: the asymptotic method and its applications. *Journal of Process Control* **8**(2), 101–115.

Zhu, Y. (2000). Parametric versus nonparametric models in MPC process identification. *Hydrocarbon Processing* **79**(2), 65–72.

Appendix A

Convergence of fixed point iteration

A.1 Fixed point iteration

To solve an equation $f(x) = 0$ using the procedure of fixed point iteration, the basic step is to express the equation as $x = g(x)$. The function $g(x)$ can be said to define a map on the real line over which x varies, such that for each value of x, the function $g(x)$ maps that point to a new point, $\tilde{x}$. Usually this map results in the points x and $\tilde{x}$ being some distance apart. If there is no motion under the map for some $x = x_p$, we call x_p a fixed point of the function $g(x)$ and which is also a zero of the corresponding function $f(x)$.

Now, suppose that we are able to choose a point x_0, which lies near a fixed point, we might speculate that under appropriate circumstances, we could use the iterative scheme, $x_{n+1} = g(x_n)$, with $n = 0, 1, 2, \cdots$, and continuing it until the difference between successive x_n is as small as required for the desired precision, an approximation of the fixed point of $g(x)$ can be obtained. This is the basic principle of a fixed point iteration scheme.

The main topics of this discussion are

1. The convergence and divergence criteria of a fixed point iteration scheme
2. Methods to make a diverging scheme converge

3. Simple transformation that makes a scheme converge monotonically which otherwise diverges monotonically.

We would make an assumption that within the range of the value of x we are interested in, the function $f(x)$ has only one solution, which is a typical assumption of a fixed point iteration scheme, which means that there is only one fixed point of $g(x)$.

A.2 Convergence criterion

Let us consider the following simple function

$$f(x) = x - m(x - a) \tag{A.1}$$

Now, for solution using fixed point iteration, the straightforward choice for $g(x)$ is

$$g(x) = m(x - a) \tag{A.2}$$

For the first case, we put a value $m = 1.5$. For this case, we denote $g(x)$ by $g_a(x)$.

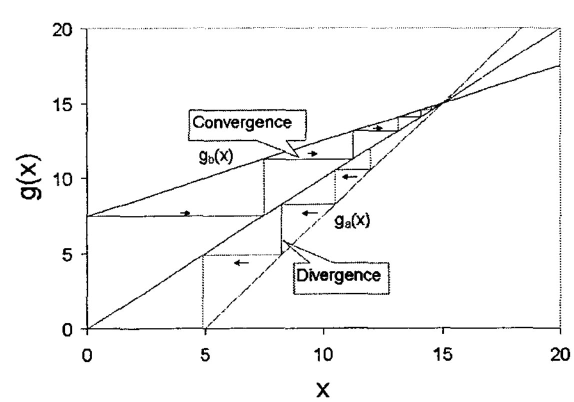


Figure A.1: *Graphical representation of fixed point iteration.*

From figure A.1 we see that, any initial value of x_0, other than x_p, the subsequent values of $x_i's$ are farther away from the fixed point which is the phenomenon of divergence. Here we observe the phenomenon of monotonic divergence.

Now, if $m = 0.5$, we can see that the iteration scheme converges monotonically. For this case, $g(x)$ is denoted by $g_b(x)$ in figure A.1.

In general, we can divide the area over which the function $g(x)$ maps x, into four sub-areas as shown in Fig.A.2. The criteria of convergence and divergence can be summarized as:

Monotonic convergence $g(x)$ bounded between $y = x$ and $y = x_p$.

Oscillating convergence $g(x)$ bounded between $y = x_p$ and $y = -x$.

Monotonic divergence $g(x)$ bounded between $y = x$ and $x = x_p$.

Oscillating divergence $g(x)$ bounded between $y = -x$ and $x = x_p$.

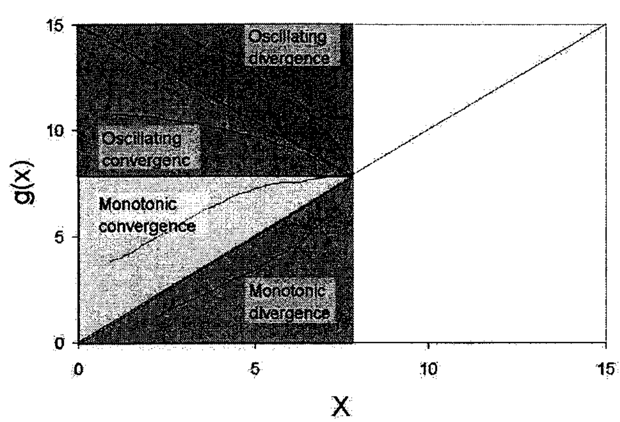


Figure A.2: *Convergence and divergence regions of fixed point iteration.*

A.3 Methods to make a diverging scheme converge

Considering the function $f(x) = x - m(x - a)$, with $m = 1.5$, we saw that expressing it as $x = g_1(x)$ where $g_1(x) = m(x - a)$, does not lead to convergence. For such a case, to develop a converging iteration scheme, an available option is to express the equation in a different way, e.g. $x = g_2(x)$, where $g_2(x)$ is bounded by the region of convergence. Say for the above example we can write

$$x = \frac{1}{m}x + a \tag{A.3}$$

Another option to express $f(x) = 0$ as $x = g_3(x) = x + f(x)$. If $g_3(x)$ maps x in the region of convergence, we can reach the solution. For the current example this can be done as writing $x = x + [x - m(x - a)]$. For the given values of $m = 1.5$ this also maps x in the region of convergence.

Now expressing $f(x) = 0$ as $x = x + f(x)$ does not lead to convergence always. Say for the above example if $m = 3.5$, then $x = x + [x - 3.5(x - a)]$ does not map x in the region of convergence. Such an equation can still be solved using the method of fixed point iteration by expressing it as $x = x + \frac{1}{r}[x - 3.5(x - a)]$. With a suitable choice of r, it is obvious that such a scheme will converge.

Now to put the current problem into perspective, we will consider only the problem of monotonic divergence. From the discussions in the previous section we can summarize the monotonic convergence/divergence criteria as

If a fixed point iteration scheme $x = g_1(x)$ diverges monotonically, another scheme $x = x + \frac{1}{r}[x - g_1(x)]$ will converge monotonically if $g_1(x)$ is bounded by the region $y = x$ and $y = (r+1)x + c$ where c is a constant satisfying that y passes through the fixed point.

A.4 The iterative procedure for parameter and delay estimation

So far we have discussed on the convergence of fixed point iteration in general. Next, we will consider the iterative procedure presented in chapter 2 for simultaneous estimation of the model parameters and the delay. We present the discussion in terms of the answers to the following questions.

Does it belong to the category of one-dimensional fixed point iteration ?

Our answer to this question is yes. Here, though we are considering estimation of all the parameters including δ, the iteration is only on δ. So, from the point of view of fixed point iteration, it can be said that we have expressed the equation as $\delta = g(\delta)$, which is the general formulation of a fixed point iteration scheme for the estimation of a single parameter.

Is the iteration scheme convergent ?

The iteration procedure is monotonically convergent except for non-minimum phase processes. However, for non-minimum phase processes it is monotonically divergent. The theoretical proof of this convergence/divergence phenomena is beyond the scope of this research. The above statement is done based on extensive simulation study. For a large number of processes with different orders and zeros and poles scattered over a wide range, simulation results show that for non-minimum phase processes $g(\delta)$ lies in the region of monotonic divergence while for minimum phase processes it lies in the monotonically convergence region. A number of simulations shows the same trend.

How to make the divergent scheme converging ?

As mentioned in the previous section, if expressing an equation in the form $\delta = g_1(\delta)$ does not lead to convergence, alternative expression as $\delta = g_2(\delta)$ can be tried. Here, we had $\delta = g_1(\delta) = \theta(n+m+2)/\theta(n+m+1)$ where, θ is given by eqn(2.48) and eqn.(2.50) with $\Phi = \Phi(\delta)$ and $\Psi = \Psi(\delta)$. and it was found that only for non-minimum phase processes this does not converge. But from simulation we found that if we express the function as $\delta = g_2(\delta) = \delta + \frac{1}{r}[\delta - g_1(\delta)]$ the iteration converges for $r = 1$. So, if the diverging scheme gives

$$\delta_{i+1}^d = g_1(\delta_i) \tag{A.4}$$

To make the scheme converging, we choose

$$\delta_{i+1}^c = \delta_i + [\delta_i - g_1(\delta_i)] = \delta_i + [\delta_i - \delta_{i+1}^d] \tag{A.5}$$

We define $\Delta\delta = \delta_i - \delta_{i+1}^d$ and for successive iteration for a value of δ_i, δ_{i+1} is computed as

$$\delta_{i+1} = \delta_i + \Delta\delta \tag{A.6}$$

Note that eqn(A.6) is applied only for the non-minimum phase processes. For the minimum phase processes, the original fixed point scheme, where $\delta_{i+1} = g_1(\delta_i)$, leads to convergence.

Appendix B

SISO equivalents of MIMO variables

B.1 Input and outputs expressions

Given below are the single input single output (SISO) equivalent expressions of input and output signals for different order multiple input multiple output (MIMO) models.

Two input three output (2×3) **process**

$$\begin{aligned} u &= u_1^1 * u_2^2 - u_1^2 * u_2^1 \\ y_{i1} &= y_i^1 * u_2^2 - y_i^2 * u_2^1 \\ y_{i2} &= y_i^2 * u_1^1 - y_i^1 * u_1^2 \\ i &= 1, 2, 3. \end{aligned}$$

Three input two output (3×2) process

$$
\begin{aligned}
u &= u_1^1 * u_2^2 * u_3^3 - u_1^1 * u_2^3 * u_3^2 - u_2^1 * u_1^2 * u_3^3 + u_2^1 * u_1^3 * u_3^2 + u_3^1 * u_1^2 * u_2^3 - u_3^1 * u_1^3 * u_2^2 \\
y_{i1} &= y_i^1 * (u_2^2 * u_3^3 - u_2^3 * u_3^2) - y_i^2 * (u_2^1 * u_3^3 - u_2^3 * u_3^1) + y_i^3 * (u_2^1 * u_3^2 - u_2^2 * u_3^1) \\
y_{i2} &= y_i^1 * (u_1^2 * u_2^3 - u_1^3 * u_2^2) - y_i^2 * (u_1^1 * u_2^3 - u_1^3 * u_2^1) + y_i^3 * (u_1^1 * u_2^2 - u_1^2 * u_2^1) \\
y_{i3} &= y_i^1 * (u_1^2 * u_2^3 - u_1^3 * u_2^2) - y_i^2 * (u_1^1 * u_2^3 - u_1^3 * u_2^1) + y_i^3 * (u_1^1 * u_2^2 - u_1^2 * u_2^1) \\
i &= 1, 2.
\end{aligned}
$$

Three input three output (3×3) process

$$
\begin{aligned}
u &= u_1^1 * u_2^2 * u_3^3 - u_1^1 * u_2^3 * u_3^2 - u_2^1 * u_1^2 * u_3^3 + u_2^1 * u_1^3 * u_3^2 + u_3^1 * u_1^2 * u_2^3 - u_3^1 * u_1^3 * u_2^2 \\
y_{i1} &= y_i^1 * (u_2^2 * u_3^3 - u_2^3 * u_3^2) - y_i^2 * (u_2^1 * u_3^3 - u_2^3 * u_3^1) + y_i^3 * (u_2^1 * u_3^2 - u_2^2 * u_3^1) \\
y_{i2} &= -y_i^1 * (u_1^2 * u_3^3 - u_1^3 * u_3^2) + y_i^2 * (u_1^1 * u_3^3 - u_1^3 * u_3^1) - y_i^3 * (u_1^1 * u_3^2 - u_1^2 * u_3^1) \\
y_{i3} &= y_i^1 * (u_1^2 * u_2^3 - u_1^3 * u_2^2) - y_i^2 * (u_1^1 * u_2^3 - u_1^3 * u_2^1) + y_i^3 * (u_1^1 * u_2^2 - u_1^2 * u_2^1) \\
i &= 1, 2, 3.
\end{aligned}
$$

Four input four output (4×4) process

$$
\begin{aligned}
u &= u_1^1 * (u_2^2 * u_3^3 * u_4^4 - u_2^2 * u_3^4 * u_4^3 - u_3^2 * u_2^3 * u_4^4 + u_3^2 * u_2^4 * u_4^3 + u_4^2 * u_2^3 * u_3^4 - u_4^2 * u_2^4 * u_3^3) \\
&\quad -u_1^2 * (u_2^1 * u_3^3 * u_4^4 + u_2^1 * u_3^4 * u_4^3 + u_3^1 * u_2^3 * u_4^4 - u_3^1 * u_2^4 * u_4^3 - u_4^1 * u_2^3 * u_3^4 + u_4^1 * u_2^4 * u_3^3) \\
&\quad +u_1^3 * (u_2^1 * u_3^2 * u_4^4 - u_2^1 * u_3^4 * u_4^2 - u_3^1 * u_2^2 * u_4^4 + u_3^1 * u_2^4 * u_4^2 + u_4^1 * u_2^2 * u_3^4 - u_4^1 * u_2^4 * u_3^2) \\
&\quad -u_1^4 * (u_2^1 * u_3^2 * u_4^3 + u_2^1 * u_3^3 * u_4^2 + u_3^1 * u_2^2 * u_4^3 - u_3^1 * u_2^3 * u_4^2 - u_4^1 * u_2^2 * u_3^3 + u_4^1 * u_2^3 * u_3^2) \\
y_{i1} &= y_i^1 * (u_2^2 * u_3^3 * u_4^4 - u_2^2 * u_3^4 * u_4^3 - u_3^2 * u_2^3 * u_4^4 + u_3^2 * u_2^4 * u_4^3 + u_4^2 * u_2^3 * u_3^4 - u_4^2 * u_2^4 * u_3^3) \\
&\quad -y_i^2 * (u_1^1 * u_3^3 * u_4^4 + u_1^1 * u_3^4 * u_4^3 + u_3^1 * u_1^3 * u_4^4 - u_3^1 * u_1^4 * u_4^3 - u_4^1 * u_1^3 * u_3^4 + u_4^1 * u_1^4 * u_3^3) \\
&\quad +y_i^3 * (u_2^1 * u_3^2 * u_4^4 - u_2^1 * u_3^4 * u_4^2 - u_3^1 * u_2^2 * u_4^4 + u_3^1 * u_2^4 * u_4^2 + u_4^1 * u_2^2 * u_3^4 - u_4^1 * u_2^4 * u_3^2) \\
&\quad -y_i^4 * (u_2^1 * u_3^2 * u_4^3 + u_2^1 * u_3^3 * u_4^2 + u_3^1 * u_2^2 * u_4^3 - u_3^1 * u_2^3 * u_4^2 - u_4^1 * u_2^2 * u_3^3 + u_4^1 * u_2^3 * u_3^2) \\
y_{i2} &= -y_i^1 * (u_1^2 * u_3^3 * u_4^4 + u_1^2 * u_3^4 * u_4^3 + u_3^2 * u_1^3 * u_4^4 - u_3^2 * u_1^4 * u_4^3 - u_4^2 * u_1^3 * u_3^4 + u_4^2 * u_1^4 * u_3^3) \\
&\quad +y_i^2 * (u_1^1 * u_3^3 * u_4^4 - u_1^1 * u_3^4 * u_4^3 - u_3^1 * u_1^3 * u_4^4 + u_3^1 * u_1^4 * u_4^3 + u_4^1 * u_1^3 * u_3^4 - u_4^1 * u_1^4 * u_3^3) \\
&\quad -y_i^3 * (u_1^1 * u_3^2 * u_4^4 + u_1^1 * u_3^4 * u_4^2 + u_3^1 * u_1^2 * u_4^4 - u_3^1 * u_1^4 * u_4^2 - u_4^1 * u_1^2 * u_3^4 + u_4^1 * u_1^4 * u_3^2) \\
&\quad +y_i^4 * (u_1^1 * u_3^2 * u_4^3 - u_1^1 * u_3^3 * u_4^2 - u_3^1 * u_1^2 * u_4^3 + u_3^1 * u_1^3 * u_4^2 + u_4^1 * u_1^2 * u_3^3 - u_4^1 * u_1^3 * u_3^2) \\
y_{i3} &= y_i^1 * (u_1^2 * u_2^3 * u_4^4 - u_1^2 * u_2^4 * u_4^3 - u_2^2 * u_1^3 * u_4^4 + u_2^2 * u_1^4 * u_4^3 + u_4^2 * u_1^3 * u_2^4 - u_4^2 * u_1^4 * u_2^3 \\
&\quad -y_i^2 * (u_1^1 * u_2^3 * u_4^4 + u_1^1 * u_2^4 * u_4^3 + u_2^1 * u_1^3 * u_4^4 - u_2^1 * u_1^4 * u_4^3 - u_4^1 * u_1^3 * u_2^4 + u_4^1 * u_1^4 * u_2^3 \\
&\quad +y_i^3 * (u_1^1 * u_2^2 * u_4^4 - u_1^1 * u_2^4 * u_4^2 - u_2^1 * u_1^2 * u_4^4 + u_2^1 * u_1^4 * u_4^2 + u_4^1 * u_1^2 * u_2^4 - u_4^1 * u_1^4 * u_2^2 \\
&\quad -y_i^4 * (u_1^1 * u_2^2 * u_4^3 + u_1^1 * u_2^3 * u_4^2 + u_2^1 * u_1^2 * u_4^3 - u_2^1 * u_1^3 * u_4^2 - u_4^1 * u_1^2 * u_2^3 + u_4^1 * u_1^3 * u_2^2) \\
y_{i4} &= -y_i^1 * (u_1^2 * u_2^3 * u_3^4 + u_1^2 * u_2^4 * u_3^3 + u_2^2 * u_1^3 * u_3^4 - u_2^2 * u_1^4 * u_3^3 - u_3^2 * u_1^3 * u_2^4 + u_3^2 * u_1^4 * u_2^3) \\
&\quad +y_i^2 * (u_1^1 * u_2^3 * u_3^4 - u_1^1 * u_2^4 * u_3^3 - u_2^1 * u_1^3 * u_3^4 + u_2^1 * u_1^4 * u_3^3 + u_3^1 * u_1^3 * u_2^4 - u_3^1 * u_1^4 * u_2^3) \\
&\quad -y_i^3 * (u_1^1 * u_2^2 * u_3^4 + u_1^1 * u_2^4 * u_3^2 + u_2^1 * u_1^2 * u_3^4 - u_2^1 * u_1^4 * u_3^2 - u_3^1 * u_1^2 * u_2^4 + u_3^1 * u_1^4 * u_2^2) \\
&\quad +y_i^4 * (u_1^1 * u_2^2 * u_3^3 - u_1^1 * u_2^3 * u_3^2 - u_2^1 * u_1^2 * u_3^3 + u_2^1 * u_1^3 * u_3^2 + u_3^1 * u_1^2 * u_2^3 - u_3^1 * u_1^3 * u_2^2) \\
i &= 1, 2, 3, 4.
\end{aligned}
$$