University of Alberta

Identification of Switching ARX Models for Hybrid Systems

by

Sohail Nazari

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
Controls

Department of Electrical and Computer Engineering

Fall 2013
Edmonton, Alberta

This dissertation is gladly and lovingly dedicated to my mother, Dr. Fatemeh Rajabi Eslami, M. D. She is a true angel who devoted her life to raising me and my sisters, and to whom I owe my existence, my life, my personality and my education.

Abstract

This dissertation explores the development of a system identification method for Switching ARX (SARX) models for Off-line and Online applications. The switching sequence in SARX models converts the model parameter's estimation into a mix-integer optimization problem. To cope with complexity of the problem, an existing approach that provides an alternative formulation for multi-mode switching models is adopted. The Algebraic Geometric approach addresses the aforementioned problem by executing the identification procedure via two steps. The first step estimates the parameters of the linear ARX model, which is constructed through embedding all the sub-models. The second step retrieves parameters of sub-models from the estimated model obtained in the first step. Although the AG method delivers exact estimation in the deterministic situation, it suffers from a lack of accuracy in the presence of noise.

This dissertation investigates the root cause of the mentioned drawback in the AG method and provides a systematic approach to deal with the measurement noise so the identification performance is improved. The proposed Stochastic Algebraic Geometric (SAG) approach reformulates the SARX parameters estimation problem into a "lifted" error-in-variable (EIV) model. Moreover, the characteristics of the proposed EIV model along with the estimation of its parameters are closely investigated. The requirements of a consistent estimation are derived through statistical analysis. In order to calculate the parameters of the sub-models improved retrieving procedures are proposed.

In order to extend the application of the SAG into the online parameter estimation, a recursive version of the SAG approach is developed. To achieve this goal, a recursive algorithm for a class of EIV models is derived. Also, a parameter retrieving procedure independent of the data points is developed to determine parameters of the sub-models.

To demonstrate a potential application of the proposed approach, a novel fault detection method is developed for linear switching systems. This approach is independent of estimating the sub-model's parameters. By using a residual evaluation method, the incipient changes of the sub-models' parameters can be detected and isolated. The applicability of this approach is demonstrated via simulation examples.

Acknowledgments

I would like to express my sincere gratitude to my supervisors, Dr. Qing Zhao and Prof. Biao Huang, for their encouragement and support over the past few years while I pursued my Ph. D. degree at the University of Alberta. Their devotion to science and careful attention to detail are highly appreciated. Without their guidance and patient help, this dissertation would not have been possible.

I would also like to sincerely thank Dr. Mahdi Tavakoli, a member of my internal examining committee. His comments and advice throughout my studies greatly aided me in the progression of my thesis. In addition, a thank you to Dr. A. Lj. Juloski for providing the data for the placement component experimental setup. Comments and considerations of the external examiners, Prof. Qing-Guo Wang and Dr. Jinfeng Liu, are highly appreciated.

I am forever indebted to my Mother, Dr. Fatemeh Rajabi Eslami, for all the patience, encouragement and love she has given to me throughout my life, and for endless sacrifices she has made for me. She developed the love of science and the passion for learning in me. I am also grateful to my sisters, Mahtab and Setareh, who have always loved and supported me.

This thesis was supported financially by my supervisors and the ECE department, which granted me several terms as an R.A. and T.A. These supports have eased the financial strains of my student life and helped me to focus on the research.

Last but not least, I offer my regards to all those who have supported me, guided me and advised me in my academic life.

Contents

List of Tables

List of Figures

List of Abbreviations

AG	Algebraic Geometric
ARX	AutoRegressive eXogenous
EIV	Error-In-Variable
ELC	Extended Linear Complementary
EW	Element-Wise
EW_TLS	Element Wise Total Least Square
FAR	False Alarm Ratio
FDI	Fault Detection and Isolation
FDR	Fault Detection Ratio
HDP	Hybrid Decoupling Polynomial
IV	Instrumental Variable
LC	Linear Complimentary
LM	Levenberg - Marquardt
LS	Least Square
MLD	Mixed Logical Dynamical
MMB	Max-Min Plus scaling
MPC	Model Predictive Control
MW	Matrix Wise
ODE	Ordinary Differential Equation
OEW	One-point Element Wise
OMW	One-point Matrix Wise

PWARX	PieceWise AutoRegressive eXogenous
PEM	Prediction Error Method
PWA	PieceWise Affine
RAG	Recursive Algebraic Geometric
REW_TLS	Recursive Element Wise Total Least Square
RSAG	Recursive Stochastic Algebraic Geometric
SAG	Stochastic Algebraic Geometric
SARX	Switch AutoRegressive eXogenous
SHDP	Stochastic Hybrid Decoupling Polynomial
STLS	Structured Total Least Square
SVM	Support Vector Machines
SWA	SWitch Affine
TLS	Total Least Square
WLS	Weighted Least Square
WTLS	Weighted Total Least Square

List of Symbols

n	The number of modes in the SARX model and its EIV model.
m	The number of inputs in an EIV model.
N	The number of data points in the SARX model and its EIV model.
K	The number of regressors in the SARX model and its EIV model.
X_1^0	The matrix of true outputs in an EIV model for all data points.
X_2^0	The matrix of true inputs in an EIV model for all data points.
X_{1t}^0	The vector of true outputs in an EIV model for the data point at time t. In the univariate case $X_{1t}^0 = [x_{1t}^0]$.
X_{2t}^0	The vector of true inputs in an EIV model for the data point at time t ($X_{2t}^0 = [x_{2_1t}^0 \;\; x_{2_2t}^0 \;\; \cdots \;\; x_{2_mt}^0]^T$).
x_{it}^0	The i^{th} true regressor value in an EIV model at time t.
B^0	The true parameters vector in an EIV model.
X^0	The matrix of true inputs and outputs in an EIV model for all data points as $X^0 = [X_1^0 \;\; X_2^0]$.
X_t^0	The vector of true inputs and outputs in an EIV model for the data point at time t as $X_t^0 = \begin{bmatrix} X_{1t}^0 \\ X_{2t}^0 \end{bmatrix}$.
X_1	The matrix of measured outputs in an EIV model for all data points.
X_2	The matrix of measured inputs in an EIV model for all data points.
X_{1t}	The vector of measured outputs in an EIV model for the data point at time t. In the univariate case $X_{1t}^0 = [x_{1t}]$.
X_{2t}	The vector of measured inputs in an EIV model for the data point at time t ($X_{2t}^0 = [x_{2_1t} \;\; x_{2_2t} \;\; \cdots \;\; x_{2_mt}]^T$).
x_{it}	The i^{th} measured regressor value in an EIV model at time t.
B	The estimated parameters vector in an EIV model.
X	The matrix of measured inputs and outputs in an EIV model for all data points as $X = [X_1 \;\; X_2]$.

X_t	The vector of measured inputs and outputs in an EIV model for the data point at time t as $X_t = \begin{bmatrix} X_{1t} \\ X_{2t} \end{bmatrix}$.
ΔX_1	The matrix of additive error terms in outputs for all data points in an EIV model.
ΔX_2	The matrix of additive error terms in inputs for all data points in an EIV model.
ΔX	The matrix of additive error terms in inputs and outputs for all data points in an EIV model, $\Delta X = [\Delta X_1 \;\; \Delta X_2]$.
ΔX_{1t}	The vector of additive error terms in outputs for the data point at time t in an EIV model. In univariate case $\Delta X_{1t} = [\Delta x_{1t}]$.
ΔX_{2t}	The vector of additive error terms in inputs for the data point at time t in an EIV model ($\Delta X_{2t} = [\Delta x_{2_1 t} \;\; \Delta x_{2_2 t} \;\; \cdots \;\; \Delta x_{2_m t}]^T$).
Δx_{it}	The i^{th} additive error term for regressor in an EIV model at time t.
ΔX_t	The vector of additive error terms in inputs and outputs for the data point at time t in an EIV model, $\Delta X_t = \begin{bmatrix} \Delta X_{1t} \\ \Delta X_{2t} \end{bmatrix}$.
V_i	The covariance matrix of the additive error vector for i^{th} data point.
V_t	The covariance matrix of the additive error vector for the data point at time t.
$V_{\Delta X_t}$	The covariance matrix of the additive error vector for the data point at time t ($V_t = V_{\Delta X_t}$).
$\otimes$	The Kronecker product.
λ_t	The discrete switching sequence at time t ($\lambda_t = i, \;\; i = 1 \cdots n$).
b_i	The vector of parameters for i^{th} sub-model in SARX model.
P_n	The HDP of order n.
ϑ_n	The Veronese map of order n.
$M_n(K)$	$M_n(K) = \binom{n+K-1}{n}$.
H	The parameter vector of the HDP or SHDP model for the SARX model.
$\tilde{H}$	A partition of H vector so that $H = [1 \;\; -\tilde{H}]$.
$D^{(n)}$	The n^{th} derivative with respect to the data point.
Z^0	The matrix of mapped regressors of the true data points so that $Z^0 = \vartheta_n(X^0) = [Z_1^0 \;\; Z_2^0 \;\; \cdots \;\; Z_{M_n(K)}^0]$.
Z_t^0	The vector of mapped regressors of the true data point at time t so that $Z_t^0 = \vartheta_n(X_t^0) = [z_{1t}^0 \;\; z_{2t}^0 \;\; \cdots \;\; z_{M_n(K)t}^0]$.

Z	The matrix of mapped regressors of the measured data points so that $Z = \vartheta_n(X) = [Z_1 \;\; Z_2 \;\; \cdots \;\; Z_{M_n(K)}]$.
Z_t	The vector of mapped regressors of the measured data point at time t so that $Z_t = \vartheta_n(X_t) = [z_{1t} \;\; z_{2t} \;\; \cdots \;\; z_{M_n(K)t}]$.
ΔZ_t	The vector of additive errors for the mapped regressors at time t in the EIV model of SHDP.
$\check{Z}_t$	The centered mapped measured data point at time t ($\check{Z}_t = Z_t - E[\Delta Z_t]$).

Chapter 1

Introduction

Nowadays, digital controllers and devices are widely adopted to control and monitor the continuous dynamical systems. The modern process is highly sophisticated, driven by the advanced digital technology. The combination of discrete control signals and continuous process data has caused an increase in the complexity of the problems facing the control community. A hybrid system is defined as a system with an interaction of continuous-time and discrete-valued dynamics. It is well-known that modeling is one of the most important steps in controlling any system, and hybrid systems are no exceptions. The modeling of hybrid systems has been studied in the last two decades. Due to the combination of discrete and continuous dynamics, the models representing these systems are much more complex than the models for continuous dynamical systems. Even linear hybrid systems are mathematically more complicated than ordinary linear systems. This complexity makes it more challenging to find a general model for all hybrid systems. Therefore researchers have made efforts to simplify the modeling of hybrid systems by imposing certain assumptions, which result in different sub-classes of hybrid systems. Some well-known models defined for hybrid systems are linear complementarity (LC) models, extended linear complementarity (ELC) models, mixed logical dynamical (MLD) models, piecewise affine (PWA) models and max-min-plus-scaling (MMPS) models, which will be reviewed in the next section. These models have been adopted in real experiments and shown to be useful. Among the aforementioned models, the PWA models are the most common ones in the literature.

1.1 Hybrid system identification and applications

Prior to research studies in the area of dynamical systems and control, hybrid automata was developed by computer scientists [1,2]. Recently, in the control field, research on hybrid dynamical systems has gained increased attention due to the application of digital technologies. In [3], a framework based on hybrid automata for modeling hybrid systems was introduced. Lygeroos et al. [4] have used the principles of optimal control to extend the concepts developed from automata theories. The technique developed determines a class of least restrictive controllers that satisfy the most important objectives, and finds a controller to optimize the system performance with respect to lower-priority objectives. The resultant controller was tested on a hybrid steam boiler benchmark problem [5]. In the control community, the switching between different dynamics is indicated by a finite set of numbers [6].

The modeling of hybrid systems depends on the adopted mathematical description. Several different approaches have been proposed. These models are categorized in [7]. Since no proper tools have been developed to analyze general cases in hybrid systems, the researchers focus more on the special subclasses of hybrid systems for which analysis technique tools are more available. Some of the examples for these subclasses are as follows: linear complementarity (LC) models, extended linear complementarity (ELC) models, mixed logical dynamical (MLD) models, piecewise affine (PWA) models and max-min-plus-scaling (MMPS) models. A brief description for each of these models is provided below:

- **Linear complementarity (LC) models:**

  A linear complementarity model is a model for hybrid dynamical system defined by a combination of the linear time-invariant ordinary differential equation (ODE) and the linear complementarity problem. In the discrete time case, LC models are given by the following equations [8,9]:

$$x(k+1) = Ax(k) + B_1u(k) + B_2w(k), \tag{1.1a}$$

$$y(k) = Cx(k) + D_1u(k) + D_2w(k), \tag{1.1b}$$

$$v(k) = E_1x(k) + E_2u(k) + E_3w(k) + g_4, \tag{1.1c}$$

$$0 \leq v(k) \perp w(k) \geq 0 \tag{1.1d}$$

where $v(k),\ w(k) \in \mathbb{R}^s,\ x(t) \in \mathbb{R}^n,\ u(t) \in \mathbb{R}^k,\ y(t) \in \mathbb{R}^l$,
$A,\ B_{1,2},\ C,\ D_{1,2},\ E_{1,2,3}$ are matrices with appropriate dimensions, g_4 is a constant vector with appropriate size, and $\perp$ shows the orthogonality of two vectors (i.e. if $a \perp b$ then $a^T b = 0$). $w(k)$ and $v(k)$ are called complementarity variables. Shen and Pang [10] investigated the Zeno states in LC systems. When a hybrid system undergoes an unbounded number of discrete transitions in a finite and bounded length of time, the Zenoness phenomenon, which is unique to hybrid systems, occurs. For example, a discrete controller that unsuccessfully attempts to satisfy an invariance specification by switching the system faster and faster among different configurations is called to be in the Zeno state [11].

- **Extended linear complementarity (ELC) models:**

  The extended version of LC changes the equality condition in (1.1c) to an inequality [12–14]. More hybrid system can be modeled by ELC models. The following equations show the discrete ELC models.

$$x(k+1) = Ax(k) + B_1 u(k) + B_2 d(k), \tag{1.2a}$$
$$y(k) = Cx(k) + D_1 u(k) + D_2 d(k), \tag{1.2b}$$
$$E_1 x(k) + E_2 u(k) + E_3 d(k) \leq g_4 \tag{1.2c}$$
$$\sum_{i=1}^{p} \prod_{j \in \phi_i} (g_4 - E_1 x(k) - E_2 u(k) - E_3 d(k))_j = 0, \tag{1.2d}$$

  where $d(k) \in \mathbb{R}^r$ is an auxiliary variable. Condition (1.2d) is equivalent to $\prod_{j \in \phi_i} (g_4 - E_1 x(k) - E_2 u(k) - E_3 d(k))_j = 0$ for each $i \in \{1, 2, \cdots, p\}$ due to the inequality condition (1.2c). This means that (1.2c) and (1.2d) can be considered as a system with p groups of linear inequalities (for every index set ϕ_i) such that, in every group, at least one inequality should coexist with equality.

- **Mixed logical dynamical (MLD) models:**

  This model describes the systems by interdependent physical laws, logic rules, and operating constraints. This model consist of linear dynamic equations subject to linear inequalities involving real and integer variables. Bemporad and Morari [15] introduced the MLD models as follow:

$$x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k), \tag{1.3a}$$
$$y(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k), \tag{1.3b}$$
$$E_1x(k) + E_2u(k) + E_3\delta(k) + E_4x(k) \leq g_5, \tag{1.3c}$$

where $x(k) = [x_r^T(k)\ x_b^T(k)]^T$ with $x_r(k) \in \mathbb{R}^{n_r}$ and $x_b(k) \in \{0,1\}^{n_b}$, and x is the state of the system that is split into two parts containing the continuous and logical components respectively. $y(k)$ and $u(k)$ have similar structures; also, $z(k) \in \mathbb{R}^{r_r}$ and $\delta(k) \in \{0,1\}^{r_b}$ are auxiliary variables. The inequality (1.3c) has to be interpreted componentwise, and the logical facts involving continuous variables can be translated to those componentwise inequalities. Recently, MLD models have been used in different applications. Andres et al. [16] have applied MLD to model $\Sigma-\Delta$ modulators along with stability analysis. In another article, MLD is used to model a front-wheel-drive passenger vehicle with manual transmission [17], and the model predictive controller (MPC) is adopted for this hybrid system. Busch et al. [18] used this model to optimize the predictive scheduling of operational strategies in the continuous process.

- **Piecewise affine(PWA) models:**

  The discrete piecewise affine model [19] is described as follows:

$$\begin{aligned} x(k+1) &= A_{\sigma(k)}x(k) + B_{\sigma(k)}u(k) + f_{\sigma(k)} \\ y(k) &= C_{\sigma(k)}x(k) + D_{\sigma(k)}u(k) + g_{\sigma(k)} \end{aligned} \tag{1.4}$$

  where $\sigma(k)$ is the discrete state indicating affine dynamics of the system at time k. For PWA models $\sigma(k)$ is defined as $\sigma(k) = i$, if and only if $[x(k)\ u(k)]^T \in \Omega_i$, and Ω_i are convex polyhedra constructed by a finite number of linear inequalities in the input/state space. It is assumed that the number of sub-models is finite, i.e. $\sigma(k) \in \{1, \cdots, n\}$, where n is a number of affine sub-models. The PWA model is one of the most commonly used models among all hybrid system models, and has been studied extensively by several researchers.

- **Max-min-plus-scaling (MMPS) models:**

  Another class for hybrid system introduced by [13] is called the max-

min-plus-scaling model, and consists of operation maximization, minimization, addition and scalar multiplication. Expressions using those operations are called MMPS expressions. A MMPS expression f of variables $x_1, \cdots, x_n$ is defined by the following syntax:

$$f := x_i | \alpha | \max(f_k, f_l) | \min(f_k, f_l) | f_k + f_l | \beta f_k \tag{1.5}$$

with $i \in \{1, 2, \cdots, n\}, \alpha, \beta \in \mathbb{R}$, and where f_k, f_l are again MMPS expressions. Operator "|" is "or" operator. Using that definition, the MMPS model can be described as follows:

$$x(k+1) = \mathcal{M}_x(x(k), u(k), d(k)), \tag{1.6a}$$
$$y(k) = \mathcal{M}_y(x(k), u(k), d(k)), \tag{1.6b}$$
$$\mathcal{M}_c(x(k), u(k), d(k)) \leq c, \tag{1.6c}$$

where $\mathcal{M}_x$, $\mathcal{M}_y$ and $\mathcal{M}_c$ are also MMPS expressions, and $d(k)$ is an auxiliary variable.

Heemels et al. [20] carried out a comprehensive review of the above models and attempted to establish equivalence relationship among them. Some of the equivalences are established under mild additional assumptions. Establishing these equivalences is useful in transferring theoretical properties and tools from one model to another. Figure 1.1 illustrates the relationship among these five models. Discussions are also available in [21] on the equivalency of PWA and MLD models, and their observability and controllability.

1.2 Piecewise affine and Switch ARX models

In switched affine (SWA) models, the switching sequence is indicated by finite linear input-state-output relations. In contrast, piecewise affine (PWA) models are switched affine models equipped with state-dependent switching law [22]. The Switched autoregressive exogenous (SARX) model, also known as Jumped autoregressive exogenous (JARX) model, is defined using the following regression vector [23]:

$$X_t = [y_{t-1}^T \ \cdots \ y_{t-n_\alpha}^T \ u_t^T \ u_{t-1}^T \ \cdots \ u_{t-n_\beta}^T]^T, \tag{1.7}$$

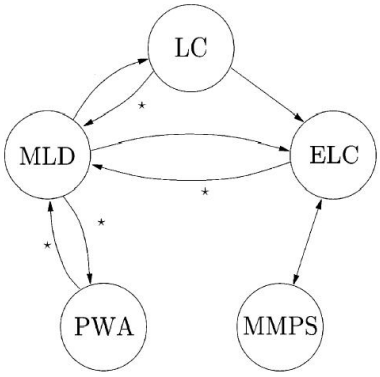

Figure 1.1: Graphical representation of the relationships between PWA, LC, ELC, MMPS and MLD models. Each arrow shows that each class is a subset of another class and $*$ represents that through the relation some additional assumption applies.

and by introducing y_k as a function of λ_k, we have:

$$y_k = b_{\lambda(t)}^T X_t \tag{1.8}$$

where $b_{\lambda(t)}$ are parameter vectors for each sub-model, and λ_t is the switching sequence. SARX models can be transformed into state space model by defining the continuous state as

$$x_t = [y_{t-1}^T \ \cdots \ y_{t-n_\alpha}^T \ u_t^T \ u_{t-1}^T \ \cdots \ u_{t-n_\beta}^T]^T. \tag{1.9}$$

Wieland et.al. [22] proved that switched affine models are equivalent to SARX models. They have shown that an observable SWA model with s different sub-models can be converted to an equivalent SARX model with s^p modes, where $p \geq n$ stands for the observability order of the original SWA model, and n is the state dimension of the system. It is also possible to find an equivalent SWA model with s modes from a SARX model with s sub-models. SWA models have been applied in pick and place machines [24], current transformers [25], tractional control [17] and motion segmentation in computer vision [26].

The identification of the SARX models has been studied extensively in recent years. Several contributions and approaches are proposed in the literature, and are summarized in [23, 27, 28]. The discrete switching sequence in the SARX models results in a mixed-integer minimization problem that makes the identification procedure computationally impractical; therefore, two main

categories of approaches try to tackle this problem. The Optimization based methods developed a relaxed version of the mix-integer optimization problem. As in [29], a convex minimization problem is solved using sparse optimization methods. The particle swarm optimization method is used to minimize the relaxed cost function in [30]. A convex relaxation, based on the solution of a constrained polynomial optimization problem, and using moments-based techniques, is developed in [31]. All these methods result in an iterative optimization problem which can be time-consuming. The second approach tries to prevent the mixed-integer optimization by reformulating the identification problem. The algebraic geometric (AG) approach [32] introduces a "lifted" one-mode ARX model from the multi-mode ARX model, called the hybrid decoupling polynomial (HDP).

The PWARX model as a subset of the SARX model has also been studied in the literature, with the following pioneering approaches: the clustering-based approach [25], the Bayesian approach [33], and the bounded error approach [24]. The clustering-based method has two major steps: first, the data are clustered into s pre-defined groups using future vectors constructed from local regressions. Second, the clustered data are used to estimate the parameters of each sub-model. In this method, the order of each sub-model and the number of sub-model(s) should be known. In the Bayesian approach, through an iterative procedure, the probability density functions of system parameters are derived by using the prior knowledge of the PDFs of the data. In this approach, the model order and the number of sub-models are also fixed. In the bounded error method, the number of sub-models is derived using an iterative procedure regarding the absolute error between the actual output and the estimated one, which should be less than tolerance parameter δ.

Other than the above major categories, a few other approaches have been used in modeling SARX systems. Lauer and Bloch used kernel regression and support vector machines (SVM) to model PWA and SARX systems [34]. However, their work has not been implemented in any experimental setup. Mixed-integer programming (MIP) [35] is also another method to identify a sub-class of PWA models called hinging hyperplane ARX (HHARX) models. The MIP method requires an extensive optimization procedure that can become very complex if the number of data points is high.

1.3 Problem formulation

The Switched ARX (SARX) model has the following representation:

$$y_t = -\alpha_1^{\lambda_t} y_{t-1} \cdots - \alpha_{n_\alpha(\lambda_t)}^{\lambda_t} y_{t-n_\alpha(\lambda_t)} - \beta_1^{\lambda_t} u_{t-1} \cdots - \beta_{n_\beta(\lambda_t)}^{\lambda_t} u_{t-n_\beta(\lambda_t)} - c^{\lambda_t} + \varepsilon_t \tag{1.10}$$

where $u_t \in \mathbb{R}$ is the input, and $y_t \in \mathbb{R}$ is the output of the system. λ_t is the discrete event state, also known as the mode of operation. In this chapter, λ_t is considered as an unknown arbitrary sequence from the integer set $\lambda_t \in \mathbb{Z} : 1, 2, \cdots, n$, where n is the total number of modes. Equation (1.10) describes a set of underlying sub-models for a SARX system when $\lambda_t = i,\ i = 1,\ 2,\ \cdots,\ n$. The ε_t is additive measurement noise, that has Gaussian distributions with zero mean and variance assumed to be δ^2.

Accordingly, $b_i = [1\ \alpha_1^i\ \alpha_2^i\ \cdots\ \alpha_{n_\alpha(i)}^i\ \beta_1^i\ \beta_{n_\beta(i)}^i\ c^i]^T$ is the parameter-set for each sub-model, and correspondingly,
$X_t = [y_t\ \cdots\ y_{t-n_\alpha(i)}\ u_{t-1}\ \cdots\ u_{t-n_\beta(i)}\ 1]^T \in \mathbb{R}^K$ is the regressor vector. The integer i is the value of the discrete state, and $K = \max n_\alpha(i) + \max n_\beta(i) + 1$.

The general identification problem for the above model is defined as follows:

General identification problem: Given the input/output data $\{u_t, y_t\}_{t=0}^T$ from the SARX system in (1.10), identify the number of sub-models, the maximum order of the ARX sub-models ($\max_i n_\alpha(i)$, $\max_i n_\beta(i)$), the model parameters $\{\alpha_j^i\}_{j=1}^{n_\alpha(i)}, \{\beta_j^i\}_{j=1}^{n_\beta(i)}, c_i$ and the unknown discrete-valued states $\{\lambda_t\}$.

As mentioned in Section 1.2, one of the methods that tackles this identification problem is the Algebraic Geometric (AG) approach [32]. The AG approach shows several advantages over other approaches. In this approach, the switching sequence does not play a role. Therefore, it is not necessary to pre-cluster all the data. Linear algebra and matrix calculus are the main mathematical tools for this approach. There is no need to know the order or the number of sub-models beforehand. Only an upper bound for the maximum order of sub-models is needed, and the rest can be estimated. In [36], these estimations are formulated and improved. This thesis has not focused on estimating the number of sub-models, or on the maximum order of sub-models;

hence they have been treated as known. Therefore, the following problem definition is provided:

The SARX parameter estimation problem: Given the input/output data $\{u_t, y_t\}_{t=0}^{T}$ from the SARX system in (1.10), estimate the model parameters $\{\alpha_j^i\}_{j=1}^{n_\alpha(i)}, \{\beta_j^i\}_{j=1}^{n_\beta(i)}, c_i$ and the unknown discrete-valued states $\{\lambda_t\}$.

The AG approach delivers a solution to *the SARX parameter estimation problem* via two unique and appealing steps. In the first step, all sub-models are embedded into one discrete-time model using geometric mapping methods. This new model is called the hybrid decoupling polynomial (HDP). The HDP is independent of the discrete switching sequence. The parameters of HDP can be estimated using linear regression . In the second step, the parameters of the original SARX model need to be retrieved from the identified HDP. In this step, a data point needs to be selected and matched to each sub-model. In the deterministic (noise-free) situation, the AG approach can deliver exact estimates for the parameters of SARX models. However, when the data is corrupted with noise, the errors caused by the noise in the parameter estimation stage become significant, which is the main drawback of this approach.

1.4 Thesis scope and outline

The overall aim of this research is to develop a batch and recursive solution to *the SARX parameter estimation problem*, using the advantages of the AG approach when the data is corrupted with measurement noise.

To reach this goal, a new approach is introduced that considers the measurement noise in the inputs/output data by reformulating the HDP into an Error-In-Variable (EIV) representation . This new formulation is called the stochastic HDP (SHDP), and the proposed approach is called the stochastic algebraic geometric (SAG) approach. It is shown that the SAG method results in major improvements in the estimation of SARX parameters under noisy conditions, benefiting from the original AG method properties and the extensive available literature in the EIV models parameter estimation [37]. The element-wise total least square (EW-TLS) method, which has been developed in the literature, is used for estimating the parameters of the SHDP. The SAG not only estimates the parameters for the SARX models, but can also be extended and applied to multi-mode EIV models when both inputs and

output are corrupted with measurement noise (with zero mean and possibly different variances).

The recursive SAG (RSAG) method is also proposed in this thesis. The covariance matrix of the errors in the EIV model of the SHDP changes by each data point. Therefore, a recursive parameter estimation method for the EIV models, when the additive errors' profile are changing, is developed. The developed method is the recursive version of the EW-TLS method. The proposed REW-TLS solution is then used to estimate the parameters of the SHDP.

Improved procedures for retrieving the sub-models' parameters in the SAG and the RSAG approaches are developed. Discussions on the abilities, advantages and drawbacks of each proposed retrieving procedure are delivered throughout the thesis.

The SAG method is then used practically in the fault detection and isolation (FDI) application, to show its potential capabilities. A novel FDI method for detecting and isolating small changes in the sub-models' parameters in the SARX models is proposed. The proposed method, detects abrupt changes in the sub-models' parameters without estimating the values of them.

Figure 1.2 illustrates the overall scope of the thesis and the relationship between chapters.

In summary, this thesis is organized as follows:

- Some mathematical tools and concepts are reviewed in chapter 2. The definition of the error-in-variable model and a related literature review are also provided in this chapter. The lemmas that are used in this thesis are also proved. Note that this chapter includes the preliminaries required in chapters 4 - 6. Therefore, the mentioned chapters reference the material discussed in chapter 2.

- Chapter 3 reviews the original AG approach and investigates the potential solutions for improving the performance of the AG approach in the presence of noisy data. Two improved retrieving procedures are also developed in this chapter.

- In chapter 4, the proposed SAG method is developed. An algorithm for estimating the parameters of the proposed SHDP is delivered, and the requirements for consistent estimation are delivered using statistical analysis. The performance comparisons between the proposed method

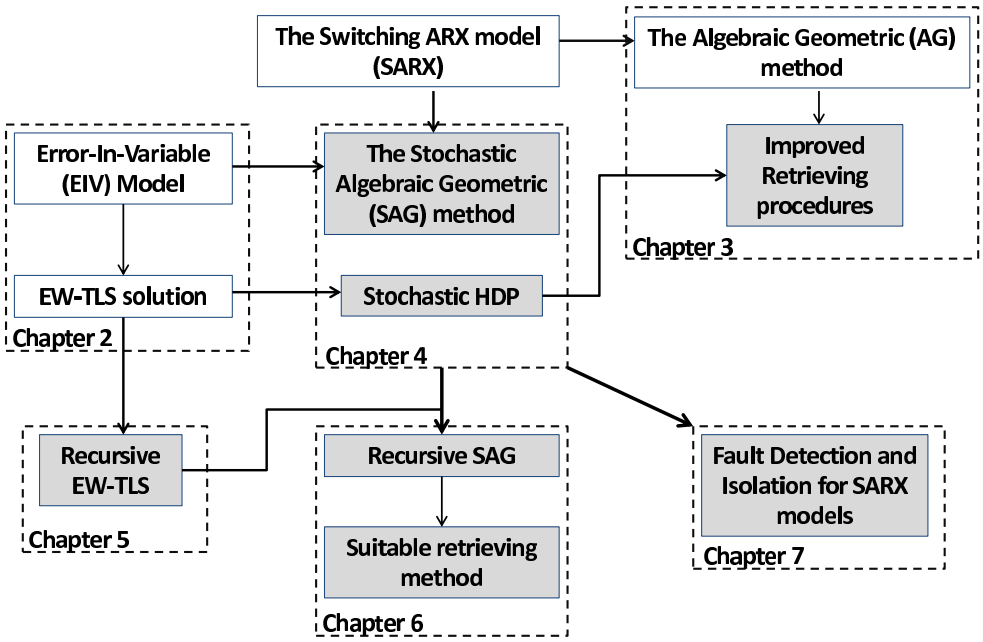

Figure 1.2: The overall outline of the thesis and contributions (The gray boxes are the developed methods in this thesis and contributions).

and existing methods are illustrated with several different simulation experiments.

- A recursive EW-TLS method is developed in chapter 5. The performance of the recursive EW-TLS method is demonstrated by several simulations.
- In chapter 6, the recursive version of the SAG approach is presented. Also, another retrieving procedure suitable for online applications is developed.
- Chapter 7 introduces the potential application of the proposed SAG method in fault detection and isolation. A novel FDI method for SARX models is presented in this chapter.
- Conclusions and a summary of the contributions are provided in chapter 9.

Chapter 2

The Preliminaries

In this chapter the error-in-variable (EIV) model and some of the existing approaches for estimating it's parameters are reviewed. In addition, some mathematical tools used in the rest of chapters are introduced. Also, some lemmas used in the rest of this thesis are defined and proved.

2.1 The error-in-variable model

The main objective of the system identification is to generate an accurate parametric model for a dynamic system, when controlling or predicting the behavior of that system is needed. The more accurate the estimated model parameters are, the more reliable the predicted output will be. The problem of system identification is more difficult if both input and output measurement of the system are corrupted with noises. In this case, the "Error-in-variables" (EIV) models usually refer to the representations of systems when the outputs and inputs are affected by noise [38]. This kind of representation is also useful when there is not enough information to distinguish the output signals from the input signals in the regressor.

The literature on EIV model based system identification is extensive. Several different methods have been developed and published which can be categorized in three classes [38]: the methods using covariance matrix, the methods using input-output spectrum and the methods using the original time series data. The instrumental variables (IV), the total least square (TLS) and the Frisch scheme belong to the first class. In [39] the so-called generalized instrumental variable estimator (GIVE) was presented. Furthermore, a thorough survey on TLS methods can be found in [40]. The Frisch scheme method

has first been used in identification of dynamic models in [41], and several improvements have been introduced since then. A good survey on EIV methods can be found in [37] and [38].

Consider the following linear systems of equations:

$$X_1^{0T} = B^{0T} X_2^{0T}, \quad X_1^0 \in \mathbb{R}^{N\times l}, \;\; X_2^0 \in \mathbb{R}^{N\times m} \tag{2.1}$$

where B^0 is the solution and $X_1^0,\ X_2^0$ are true data. The noise free (nominal) data is called the true data. N is the number of data points, m is the number of the inputs and l is the number of the outputs. In this thesis the univariate case is considered, therefore $l = 1$. Equation (2.1) can also be written as:

$$X_1^0 = X_2^0 B^0. \tag{2.2}$$

The matrix X^0 is the true regressor matrix for all the data points and is defined as:

$$X^0 = [X_1^0 \;\; X_2^0] = \begin{bmatrix} x_{11}^0 & x_{2_1 1}^0 & x_{2_2 1}^0 & \cdots & x_{2_m 1}^0 \\ x_{12}^0 & x_{2_1 2}^0 & x_{2_2 2}^0 & \cdots & x_{2_m 2}^0 \\ \vdots & \vdots & \vdots & & \vdots \\ x_{1t}^0 & x_{2_1 t}^0 & x_{2_2 t}^0 & \cdots & x_{2_m t}^0 \\ \vdots & \vdots & \vdots & & \vdots \\ x_{1N}^0 & x_{2_1 N}^0 & x_{2_2 N}^0 & \cdots & x_{2_m N}^0 \end{bmatrix}, \tag{2.3}$$

where,

$$X_1^0 = \begin{bmatrix} x_{11}^0 \\ x_{12}^0 \\ \vdots \\ x_{1t}^0 \\ \vdots \\ x_{1N}^0 \end{bmatrix}, \qquad X_2^0 = \begin{bmatrix} x_{2_1 1}^0 & x_{2_2 1}^0 & \cdots & x_{2_m 1}^0 \\ x_{2_1 2}^0 & x_{2_2 2}^0 & \cdots & x_{2_m 2}^0 \\ \vdots & \vdots & & \vdots \\ x_{2_1 t}^0 & x_{2_2 t}^0 & \cdots & x_{2_m t}^0 \\ \vdots & \vdots & & \vdots \\ x_{2_1 N}^0 & x_{2_2 N}^0 & \cdots & x_{2_m N}^0 \end{bmatrix}, \tag{2.4}$$

and $x_{jt}^0, \;\; j \in \{1 \;\; , 2_1 \;\; 2_2 \;\; \cdots \;\; 2_m\}$, are true regressors at time t. Therefore,

the true data point at time t is described by the following vector:

$$X_t^0 = \begin{bmatrix} x_{1t}^0 \\ x_{2_1t}^0 \\ \vdots \\ x_{jt}^0 \\ \vdots \\ x_{2_mt}^0 \end{bmatrix} = \begin{bmatrix} x_{1t}^0 \\ X_{2t}^0 \end{bmatrix} = \begin{bmatrix} X_{1t}^0 \\ X_{2t}^0 \end{bmatrix}. \tag{2.5}$$

Note that in the univariate case $X_{1t}^0 = x_{1t}^0$. Equation (2.1) is also valid for each data point, therefore the following equation is also used in this thesis:

$$X^0{}_{1t} = B^{0T} X_{2t}^0 \Rightarrow [1 \quad -B^{0T}] X_t^0 = 0. \tag{2.6}$$

The total least square (TLS) aims to approximate this solution when the data measurements $X_1 = X_1^0 + \Delta X_1$ and $X_2 = X_2^0 + \Delta X_2$ are perturbed by $\Delta X = [\Delta X_1 \quad \Delta X_2]$. The ΔX is the measurement noise matrix. Therefore, the error-in-variable model for the (2.1) is:

$$X_1{}^T \approx B^T X_2{}^T \tag{2.7}$$

Consequently, the EIV model for each data point is:

$$X_{1t} \approx B^T X_{2t}. \tag{2.8}$$

The noise profile in the inputs and outputs and their correlations among each other have an important impact on the estimation accuracy. The original TLS solution is restricted to the parameter estimation in static models when there is no correlation between inputs and output measurement noises [38] . In the dynamic models the data matrix has the block-Hankel structure. This is the motivation for developing a solution called Structured TLS (STLS) for identifying dynamic models [42]. Several improvements have been developed for STLS for decreasing its computation time and complexity [43] and to cover data matrices with more variety of structural forms [44].

2.1.1 The total least square solution

If the error in the ΔX matrix have uncorrelated Gaussian distribution with mean zero and constant variance σ^2, then the conventional TLS method provides a consistent estimation to the following minimization [45]:

$$B_{TLS} = \arg\min_{B} \|[\Delta X_1 \ \Delta X_2]\|_F^2 \qquad \text{subject to } (X_1 - \Delta X_1) = (X_2 - \Delta X_2)B, \tag{2.9}$$

where $||\cdot||_F$ is the Frobenius norm. The solution to this problem using singular value decomposition (SVD) is given as follow: Let $X = [X_1 \ X_2] = U\Sigma W^T$ where $\Sigma = \text{diag}(\sigma_1, \cdots, \sigma_{m+1})$ is a singular value decomposition of X. Denote the following partitioning: $W = \left[\begin{smallmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{smallmatrix}\right]$ and $\Sigma = \left[\begin{smallmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{smallmatrix}\right]$, in which W_{22} and Σ_2 have the dimension of $m \times m$. The TLS solution exists if and only if W_{22} is non-singular and it is unique if and only if $\sigma_i \neq \sigma_{i+1}$. With these conditions, the solution will be $B_{TLS} = -W_{12} \times W_{22}^{-1}$.

2.1.2 The element-wise total least square solution

Different approaches and algorithms have been developed in the literature to provide a consistent parameter estimation B for B^0 under different noise conditions. One of the important indicators for choosing the appropriate TLS method is the form of the covariance matrix of ΔX, V, which indicates the correlation between the measurements noise and their variances. These different methods are reviewed and categorized in [40] and [46].

The total least square (TLS) solution to the regression problem in Section 2.1.1 provides consistent estimation for B^0 if V has a constant format of $\sigma^2 I$ for all the data points [45]. The structured total least square (STLS) method, estimates the parameters in (2.1) when the regressor matrix X has a structural form (block-Hankel form in dynamic models) [42, 43]. When the measurement noise matrix is row wise independent and column wise correlated with equal row covariance matrix, the weighted total least square provides consistent estimation [47, 48]. The element wise weighted total least square method is developed for a special case when the measurements noise row covariance matrices are not equal and vary among the rows (changes among the data points) [49].

The EW-TLS minimization problem is defined as:

$$\min_{B,\Delta X} \sum_{t=1}^{N} ||V_t^{-1/2}\Delta X_t||_2^2 \quad s.t.(X+\Delta X)\begin{bmatrix} I \\ -B \end{bmatrix} = 0. \tag{2.10}$$

where V_t is the covariance matrix for the ΔX_t for each data point t and N is the number of the data points. This minimizations provides a consistent parameter estimation for B^0 [49] if the regressor error vector ΔX has zero mean. Equation (2.10) can be reformed into the following unconstrained minimization problem:

$$\begin{aligned} &\min_{B} \sum_{t=1}^{N} f_t(B), \\ &f_t(B) = r_t^T(B) Q_t^{-1} r_t(B) \end{aligned} \tag{2.11}$$

where $Q_t(B) = [I \quad -B^T] \times V_t \times \begin{bmatrix} I \\ -B \end{bmatrix}$ and $r_t(B) = X_{1_t} - B^T X_{2_t}$. V_t is partitioned as

$$V_t = \begin{bmatrix} var(X_{1_t}) & cov(X_{1_t}, X_{2_t}) \\ cov(X_{2_t}, X_{1_t}) & var(X_{2_t}) \end{bmatrix} = \begin{bmatrix} V_{i_t} & V_{t_{12}} \\ V_{t_{21}} & V_{t_2} \end{bmatrix}. \tag{2.12}$$

Standard local optimization methods are preferable choices for solving the above non-convex minimization problem. In [50] the Levenberg-Marguardt algorithm (Matlab's lsqnonlin) is considered. Another algorithm based on the classical optimization methods is developed in [51]. [49] introduced an interactive algorithm in the univariate case ($l = 1$) using a procedure developed in [52].

The EW-TLS method's consistent estimation requires the covariance matrices values to be known up to a scalar coefficient. Therefore any matrix cV_t can be used instead of V_t, where c is a constant. This relaxes the condition of knowing the true variance values of the measurements noise. Handling the measured data with combination of noisy and deterministic values is another advantage of this method. The deterministic measurements result in the near zero rows and columns in V_ts, which makes them near singular. The EW-TLS method is able to tolerate this singularity problem so that it does not have adverse impact on the estimation results.

2.2 The Kronecker product

The Kronecker product, also known as the direct product or the Tensor product, is defined by a partitioned matrix whose (i,j)'s partition is $A_{ij}B$ [53]:

$$A \otimes B \equiv \left[\begin{pmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{pmatrix} \right] = \text{matrix}\left[A_{ij}B\right]. \quad \begin{matrix} A \in \mathbb{R}^{m\times n} \\ B \in \mathbb{R}^{r\times s} \\ A\otimes B \in \mathbb{R}^{mr\times ns} \end{matrix} \tag{2.13}$$

Some of the properties of this product are as follows.

$$A \otimes (B + C) = A \otimes B + A \otimes C \quad \text{Distributivity} \tag{2.14a}$$

$$(A \otimes B) \otimes C = A \otimes (B \otimes C) \quad \text{Associativity} \tag{2.14b}$$

$$(A \otimes B)(C \otimes D) = AC \otimes BD \quad \text{Mixed product rule} \tag{2.14c}$$

$$(A \otimes B)^T = (A^T \otimes B^T), \quad (A \otimes B)^{-1} = (A^{-1} \otimes B^{-1}) \tag{2.14d}$$

Also the following derivatives are defined:

$$\frac{\partial A(M)B(M)}{\partial M^{r\times s}} = \frac{\partial A}{\partial M}(I_s \otimes B) + (I_r \otimes A)\frac{\partial A}{\partial M} \tag{2.15a}$$

$$\frac{\partial a(M)B(M)}{\partial M^{r\times s}} = \frac{\partial a}{\partial M} \otimes B + a(M)\frac{\partial A}{\partial M} \tag{2.15b}$$

$$\frac{\partial A^{n\times m}(M) \otimes B^{k\times l}(M)}{\partial M^{r\times s}} = \frac{\partial A}{\partial M} \otimes B + (I_r \otimes U_{nk})(\frac{\partial B}{\partial M} \otimes A)(I_s \otimes U_{lm}) \tag{2.15c}$$

$$\frac{\partial A^{-1}(M)}{\partial M^{r\times s}} = -(I_r \otimes A^{-1})\frac{\partial A}{\partial M}(I_s \otimes A^{-1}) \tag{2.15d}$$

where $A(M)$ and $B(M)$ are matrix functions of matrix M and $a(M)$ is a scalar function of matrix M.

Also, let us prove the following Lemma which will be used later.

Lemma 2.1: If A_l is a square matrix, and B is a matrix with appropriate

dimensions, the following equation holds :

$$\coprod_{l=1}^{n}(BA_lB^T) = \left(\coprod_{l=1}^{n} B\right) \times \left(\coprod_{l=1}^{n} A_l\right) \times \left(\coprod_{l=1}^{n} B^T\right), \tag{2.16}$$

where $\coprod_{l=1}^{n} A_l = A_1 \otimes A_2 \otimes A_3 \otimes \cdots \otimes A_n$.

Proof. According to associativity property and the mixed product rule of the Kronecker product we have:

$$\begin{aligned}
\coprod_{l=1}^{n} BA_lB^T &= (BA_1B^T) \otimes (BA_2B^T) \otimes \cdots \otimes (BA_nB^T) \\
&= \big((BA_1B^T) \otimes (BA_2B^T)\big) \otimes \cdots \otimes (BA_nB^T) \\
&= \big((B \otimes B) \times (A_1B^T \otimes A_2B^T)\big) \otimes (BA_3B^T) \otimes \cdots \otimes (BA_n^TB^T) \\
&= \big((B \otimes B) \times (A_1 \otimes A_2) \times (B^T \otimes B^T)\big) \\
&\qquad \otimes (BA_3B^T) \otimes \cdots \otimes (BA_n^TB^T) \\
&= \big(\big((B \otimes B) \times (A_1 \otimes A_2) \times (B^T \otimes B^T)\big) \otimes (BA_3B^T)\big) \\
&\quad \otimes (BA_4B^T) \otimes \cdots \otimes (BA_n^TB^T) \\
&\vdots \\
&= (B \otimes B \otimes \cdots \otimes B) \times (A_1 \otimes A_2 \otimes \cdots \otimes A_n) \\
&\qquad \times (B^T \otimes B^T \otimes \cdots \otimes B^T) \\
&= \left(\coprod_{l=1}^{n} B\right) \times \left(\coprod_{l=1}^{n} A_l\right) \times \left(\coprod_{l=1}^{n} B^T\right)
\end{aligned}$$

□

2.3 Moments of the normal distribution and some of its properties

The moment generating function of a random variable ΔX with normal distribution $\mathcal{N}(\mu, \sigma^2)$ is defined as [54]:

$$M_{\Delta X}(t) = E[e^{t\Delta X}] = e^{t\Delta X + \frac{1}{2}\sigma^2 t^2} \tag{2.17}$$

Also the following property holds:

$$E[\Delta X^k] = M^{(k)}_{\Delta X}(0). \tag{2.18}$$

By using the above, if the random variable ΔX has $\mu = 0$ then its moments of degree $k(E[(\Delta X)^k])$ can be calculated as:

$$E[\Delta X^k] = \begin{cases} 0 & k = 2r-1 \\ \frac{(\sigma^2/2)^r (2r)!}{(r)!} = (k-1)!!\ \sigma^k & k = 2r \end{cases} \tag{2.19}$$

where $k!!$ is the factorial of the odd numbers for k. The following two lemma will be used in the rest of the thesis:

Lemma 2.2: Assume that X is a deterministic variable, therefore The following equation holds:

$$E[(X - \Delta X)^j] = \begin{cases} \sum_{k=0}^{r-1} (B_k X^{j-2k} \sigma^{2k}) & j = 2r-1 \\ \sum_{k=0}^{r} (B_k X^{j-2k} \sigma^{2k}) & j = 2r \end{cases} \tag{2.20}$$

where $B_k = \begin{pmatrix} j \\ 2k \end{pmatrix} (2k-1)!!$.

Proof. Using (2.19) the following proves the lemma:

$$\begin{aligned} E[(X - \Delta X)^j] &= E[\sum_{i=0}^{j} ((-1)^i \begin{pmatrix} j \\ i \end{pmatrix} X^{j-i} (\Delta X)^i)] \\ &= \sum_{i=0}^{j} ((-1)^i \begin{pmatrix} j \\ i \end{pmatrix} B_k \Delta X^{j-i} E[\Delta X^i]) \\ &= \begin{cases} \sum_{k=0}^{r-1} (B_k X^{j-2k} \sigma^{2k}) & j = 2r-1 \\ \sum_{k=0}^{r} (B_k X^{j-2k} \sigma^{2k}) & j = 2r \end{cases} \end{aligned}$$

□

Lemma 2.3: The additional deterministic values does not have any effect on the covariance of random variables. Thus, for any two random variables $\Delta X_1, \Delta X_2$ and deterministic variables X_1, X_2 the following holds:

$$Cov(X_1 + \Delta X_1, X_2 + \Delta X_2) = Cov(\Delta X_1, \Delta X_2) \tag{2.21}$$

Proof.

$$
\begin{aligned}
&Cov(X_1 + \Delta X_1, X_2 + \Delta X_2) \\
&= E[(X_1 + \Delta X_1 - E[X_1 + \Delta X_1])(X_2 + \Delta X_2 - E[X_2 + \Delta X_2])] \\
&\qquad - E[(X_1 + \Delta X_1 - E[X_1 + \Delta X_1])]E[(X_2 + \Delta X_2 - E[X_2 + \Delta X_2])] \\
&= E[(X_1 + \Delta X_1 - X_1 - E[\Delta X_1])(X_2 + \Delta X_2 - X_2 - E[\Delta X_2])] \\
&\qquad - E[(X_1 + \Delta X_1 - X_1 - E[\Delta X_1])]E[(X_2 + \Delta X_2 - X_2 - E[\Delta X_2])] \\
&= E[(\Delta X_1 - E[\Delta X_1])(\Delta X_2 - E[\Delta X_2])] \\
&\qquad\qquad - E[(\Delta X_1 - E[\Delta X_1])]E[(\Delta X_2 - E[\Delta X_2])] \\
&= Cov(\Delta X_1, \Delta X_2)
\end{aligned}
$$

□

Chapter 3

The AG Noise Problem and New Retrieving Procedures

[1] In this section, an overview on the Algebraic Geometric approach is presented, and the roots of noise handling problem in this approach are investigated. As discussed, in the AG method all the ARX models are embedded in one linear model called hybrid decoupling polynomial (HDP). The parameters of HDP are then estimated using linear regression. In the second stage, the parameters of SARX model need to be retrieved from the identified HDP. In this section two new approaches are presented that improve the performance of retrieving stage under noisy conditions significantly.

3.1 The AG approach

The Switched ARX (SARX) model has the following representation:

$$y_t = -\alpha_1^{\lambda_t} y_{t-1} \cdots - \alpha_{n_\alpha(\lambda_t)}^{\lambda_t} y_{t-n_\alpha(\lambda_t)} - \beta_1^{\lambda_t} u_{t-1} \cdots - \beta_{n_\beta(\lambda_t)}^{\lambda_t} u_{t-n_\beta(\lambda_t)} - c^{\lambda_t} + \varepsilon_t \tag{3.1}$$

[1] A version of this chapter has been published in following papers:

Nazari, Sohail, Qing Zhao, and Biao Huang. "An improved algebraic geometric solution to the identification of switched ARX models with noise." American Control Conference (ACC), 2011. IEEE, 2011.

Nazari, Sohail, Qing Zhao, and Biao Huang. "Matrix-wise approach for identification of multi-mode Switched ARX models with noise." American Control Conference (ACC), 2012. IEEE, 2012.

where $u_t \in \mathbb{R}$ is the input, and $y_t \in \mathbb{R}$ is the output of the system. λ_t is the discrete event state, also known as the mode of operation. In this chapter, λ_t is considered as an unknown arbitrary sequence from the integer set $\lambda_t \in \mathbb{Z} : 1, 2, \cdots, n$, where n is the total number of modes. Equation (3.2) describes a set of underlying sub-models for SARX system when $\lambda_t = i,\ i = 1,\ 2,\ \cdots,\ n$. The ε_t is additive measurement noise that has Gaussian distributions with zero mean and variance assumed to be δ^2.

Another representation of (3.1) is as follows:

$$\begin{cases} b_1^T X_t = \varepsilon_t \\ b_2^T X_t = \varepsilon_t \\ \vdots \\ b_n^T X_t = \varepsilon_t \end{cases}, \tag{3.2}$$

where $b_i = [1\ \alpha_1^i\ \alpha_2^i\ \cdots\ \alpha_{n_\alpha(i)}^i\ \beta_1^i\ \beta_{n_\beta(i)}^i\ c^i]^T$ is the parameter-set for each sub-model and correspondingly, $X_t = [y_t\ \cdots\ y_{t-n_\alpha(i)}\ u_{t-1}\ \cdots\ u_{t-n_\beta(i)}\ 1]^T \in \mathbb{R}^K$ is the regressor vector. The integer i is the value of the discrete state at time t, and $K = \max n_\alpha(i) + \max n_\beta(i) + 1$. Note that only one of the sub-models in (3.2) is valid at each data point. In other words, if X_t belongs to the sub-model i at time t ($\lambda_t = i$) then the linear regression model $b_i^T X_t = \varepsilon_t$ represents the system at that time.

3.1.1 The HDP formulation

The main difference between the SARX model and the ARX model is the discrete switching sequence (λ_t), which dramatically increases the complexity of the identification procedure. Since each data point belongs to a different mode, the linear regression cannot be used to estimate the parameters. To circumvent this difficulty, a model that is independent from switching sequence should be constructed. Notice that for any one data point, one of the sub-models in (3.2) should hold. Taking the product of all equations of all sub-models in (3.2) and embedding them into one higher order equation ensures that all the data points satisfy the following polynomial equation:

$$P_n(X_t) = \prod_{i=1}^{n} b_i^T X_t = \epsilon_t. \tag{3.3}$$

In this section, the original AG approach [32] is reviewed. This methods is based on the assumption that the error (ϵ_t) in (3.3) is zero (when the sub-models are noise free, $\varepsilon_t = 0$) or negligible. However we will show that this error can be significant in the presence of the noise and is not simply the multiplication of all white noises ($\epsilon_t \neq \prod_{i=1}^{n} \varepsilon_t$).

P_n is called the hybrid decoupling polynomial (HDP) [32]. Although taking the product is not the only way to eliminate the switching sequence, the use of the HDP leads to certain advantages in its algebraic structure. The HDP is a multivariate polynomial of degree n with K variables, which can be written linearly in terms of its coefficients as

$$P_n(X_t) = \prod_{i=1}^{n} b_i^T X_t = \sum_{I=1}^{M_n(K)} h_I z_{It} = H_n^T \vartheta_n(X_t) = \epsilon_t \tag{3.4}$$

where, $h_I \in \mathbb{R}$ is the coefficient of the monomial

$$z_{It} = y_t^{n_1}\ y_{t-1}^{n_2}\ \cdots\ y_{t-n_{\max(n_\alpha(i))}}^{\max(n_\alpha(i))}\ u_{t-1}^{\max(n_\alpha(i))+1}\ \cdots\ u_{t-\max(n_\beta(i))}^{n_{K-1}}\ 1^{n_K},$$

where $0 \leq n_j \leq n,\ j = 1 \cdots K,\ n_1 + \cdots + n_K = n.\ \vartheta_n : \mathbb{R}^K \to \mathbb{R}^{M_n(K)}$ is a Veronese map of degree n [55] , which is defined as $\vartheta_n : [x_{1t}\ \cdots\ x_{Kt}]^T \to [\cdots\ z_{It}\ \cdots]^T$ with I chosen in the degree-lexicographic order, e.g.

$$\begin{aligned} \vartheta_2([x_{1t}\ x_{2t}\ x_{3t}]^T) &= \begin{bmatrix} x_{1t}^2 & x_{1t}x_{2t} & x_{1t}x_{3t} & x_{2t}^2 & x_{2t}x_{3t} & x_{3t}^2 \end{bmatrix}^T \\ &= [z_{1y}\ \ z_{2t}\ \ z_{3t}\ \ z_{4t}\ \ z_{5t}\ \ z_{6t}]^T; \end{aligned} \tag{3.5}$$

$M_n(K) = \binom{n+K-1}{n}$ is the total number of independent monomials in (3.4), and $H = [h_1\ \ h_2\ \ \cdots\ \ h_{M_n(K)}]^T$ is the parameters vector of the HDP. In order to have unique parameter vector H, h_1 is assumed to be equal to one.

Equations (3.3) and (3.4) should hold for all data points. Therefore, the following equation represents a linear ARX model with the parameter vector H:

$$L_n(n_\alpha, n_\beta)H = \left[\vartheta_n\left(X_{\max(n_\alpha,n_\beta)}\right)\ \cdots\ \vartheta_n\left(X_{\max(n_\alpha,n_\beta)+N-1}\right)\right]^T H = \epsilon_{N\times 1}, \tag{3.6}$$

where N is the number of data points, and $L_n(max(n_\alpha(i), n_\beta(i))) \in \mathbb{R}^{T\times M_n(K)}$ is the matrix of the embedded and mapped input/output data via the Veronese variety. Note that in this linear model, the switching sequence is eliminated

completely. Estimation of the SARX parameters can now be achieved in two steps. In the first step the HDP is constructed by (3.4) and its parameters are estimated through (3.6). In the second step the SARX parameters for each sub-model are recovered from the estimated HDP.

3.1.2 The HDP's parameters estimation

As discussed above, equation (3.6) represents a linear ARX model. In this ARX model the elements of $\vartheta_n(X_t)$ are regressors and H_n is the parameter vector. Estimating the coefficients of HDP can be done through regression. In [56] a least square solution is suggested for this regression problem, but due to the multiplication nature of $\vartheta_n(X_t)$, the regressors in P_n carry measurement errors. Therefore in this regression problem the regressors have errors in themselves. As it is discussed in [45] the least square solution to the regression problem with error in variable will result in a biased estimation. Therefore, a better candidate for solving this regression problem is total least square (TLS) method. The main goal in the identification of SARX models is to find the parameters of each sub-model. Estimating these parameters can be done using the estimated HDP. Ma and Vidal [32] suggested a method for solving this problem by using the derivatives of HDP in the deterministic (noise-free) situation:

$$DP_n(X_t) = \frac{\partial P_n(X_t)}{\partial X_t} = \frac{\partial(\prod_{i=1}^{n} b_i^T X_t)}{\partial X_t} = \sum_{i=1}^{n} b_i \prod_{l \neq i}^{n} b_l^T X_t. \tag{3.7}$$

If ${X_t}^{\lambda_t}$ belongs to the i^{th}sub-model so that $\lambda_t = i$, then $b_i^T {X_t}^{\lambda_t} = 0$, and since the first element of each b_i is equal to 1, each parameter set can be found uniquely through the following equation:

$$b_i = \frac{DP_n({X_t}^{\lambda_t})}{(e_1^T DP_n({X_t}^{\lambda_t}))}. \tag{3.8}$$

where ${X_t}^{\lambda_t}$ is the data point belonging to the i^{th} sub-model. In this method, at least one data-point is needed for identifying each sub-model's parameter. In order to have an unsupervised procedure for estimating the parameters, one point belonging to each sub-model should be found. But when there is noise, the data points in the space of the embedded mapped model do not lie exactly on the hyperplanes constructed by (3.4). Therefore, the nearest point

to each hyperplane should be found. This becomes the problem of calculating the distance of each data point to its closest hyperplane without knowing the normals of the hyperplanes. In [56], the first-order approximation to such distance is derived. Using that approximation an algorithm is developed to find the nearest point to each hyperplanes. By using those founded data points, each b_i can be recovered through (3.8). This algorithm works well for data with small noise variances, but when the variance of the noise increases, this algorithm does not provide accurate results and can end up into the local minimum. The situation is worse when the system is more complex and has higher orders. Even when the appropriate data point is found, equation (3.8) holds only in the absence of the noise. Also in order to recover the parameter vectors, one data point should be clustered for each sub-model. The above issues are motivations for developing new methods to recover the parameter vectors from HDP without clustering any data points and as a result having more accuracy in presence of noise.

3.2 Improved retrieving procedures

3.2.1 The element-wise approach

In order to overcome the problems addressed in the previous section, we proposed an approach that avoids finding the data point for each sub-model. In fact, in this approach, all data points are utilized without being clustered. This approach results in a closed-form solution for the two-mode models. We name this as element-wise approach, in the sense that the parameters of the original sub-models are solved one by one in an analytical form.

In this approach, after reconstructing (3.8) to make it suitable for a linear regression solution, all the data points are used to build a system of equations. The derivation is as follows. Consider the two-mode model as:

$$\begin{cases} b_1^T X_t = \varepsilon_t \\ b_2^T X_t = \varepsilon_t \end{cases}, \tag{3.9}$$

and its corresponding HDP as: $P_n(X_t) = (b_1^T X_t) \times (b_2^T X_t)$. The derivative

with respect to X_t is:

$$DP_n(X_t) = \frac{\partial P_n(X_t)}{\partial X_t} = b_1(b_2^T X_t) + b_2(b_1^T X_t) \tag{3.10}$$

$DP_n(X_t)$ is a vector since X_t is a vector. Therefore, (3.10) is an equation in the vector form. In order to fit the problem into a linear regression problem, one can consider each element of the above equation:

$$D_i P_n(X_t) = b_{1i}(b_2^T X_t) + b_{2i}(b_1^T X_t), \tag{3.11}$$

where the i^{th} element of $DP_n(X_t)$ is $D_i P_n(X_t)$, and the j^{th} element of b_i is b_{ij}. Now, starting with the first element, we have

$$D_1 P_n(X_t) = b_{11}(b_2^T X_t) + b_{21}(b_1^T X_t),$$

As assumed previously, that b_{11} and b_{21} are equal to one. Therefore,

$$D_1 P_n(X_t) = (b_2^T X_t) + (b_1^T X_t) = (b_1^T + b_2^T)X_t = X_t^T(b_1 + b_2).$$

Now, using all the data points in the above equation results in

$$\begin{bmatrix} D_1 P_n(X_1) \\ \vdots \\ D_1 P_n(X_N) \end{bmatrix} = \begin{bmatrix} X_1^T \\ \vdots \\ X_N^T \end{bmatrix} (b_1 + b_2) = L\,(b_1 + b_2)\,, \tag{3.12}$$

It is clear that above equation is a linear regression problem with regression vector (L), and the parameter vector to be estimated is $(b_1 + b_2)$. Since $D_1 P_n(X_t)$ is noisy, and some elements of X_t also have in-variable noise, the TLS method will be used to find the parameter estimation.

Again, as discussed in Section 2.1.1, the X matrix is equal to $\begin{bmatrix} \begin{bmatrix} X_1^T \\ \vdots \\ X_N^T \end{bmatrix} & \begin{bmatrix} D_1 P_n(X_1) \\ \vdots \\ D_1 P_n(X_N) \end{bmatrix} \end{bmatrix}$; and with the SVD method, $(b_1 + b_2)$ can be estimated. Let us assume

$$(b_1 + b_2) = A^1. \tag{3.13}$$

A^1 is a vector with K elements, containing the estimated parameters from the

first element of the HDP derivative. Using (3.10) in (3.13), we have:

$$DP_n(X_t) = b_1\left((A^1 - b_1)^T\right) + (A^1 - b_1)(b_1^T X_t) \tag{3.14}$$

This equation is a vector-wise equation. Using the second element of the above equation, we obtain

$$\begin{aligned} &D_2P_n\left(X_t\right) = b_{12}\left((A^1 - b_1)^T X_t\right) + \left(A_2^1 - b_{12}\right)\left(b_1^T X_t\right) \\ &= \left(b_{12}\left(A^1 - b_1\right)^T + \left(A_2^1 - b_{12}\right) b_1^T\right) X_t = X_t^T\left(b_{12}\left(A^1 - b_1\right) + \left(A_2^1 - b_{12}\right) b_1\right). \end{aligned}$$

Similar to (3.12), all the data points should be used in the above equation, resulting in the following equation:

$$\begin{aligned} \begin{bmatrix} D_2P_n\left(X_1\right) \\ \vdots \\ D_2P_n\left(X_T\right) \end{bmatrix} &= \begin{bmatrix} X_1^T \\ \vdots \\ X_N^T \end{bmatrix} \left(b_{12}\left(A^1 - b_1\right) + \left(A_2^1 - b_{12}\right) b_1\right) \\ &= L\left(b_{12}\left(A^1 - b_1\right) + \left(A_2^1 - b_{12}\right) b_1\right). \end{aligned}$$

Again, using the SVD and TLS, we have

$$\left(b_{12}\left(A^1 - b_1\right) + \left(A_2^1 - b_{12}\right) b_1\right) = A^2, \tag{3.15}$$

where A^2 contains the parameters estimated from the second element of the HDP derivative. The goal is to find b_1. Based on the second element of each vector in (3.15), we have

$$A_2^2 = \left(b_{12}\left(A_2^1 - b_{12}\right) + \left(A_2^1 - b_{12}\right) b_{12}\right) = 2b_{12}A_2^1 - 2b_{12}^2.$$

This ends in $b_{12}^2 - A_2^1 b_{12} + \frac{1}{2}A_2^2 = 0$, which is a second-order equation with respect to b_{12} . By using solution formula for the quadratic equations, we have

$$b_{12} = \frac{A_2^1 \pm \sqrt{(A_2^1)^2 - 2A_2^2}}{2}. \tag{3.16}$$

Now that b_{12} is found, using the third element of each vector in (3.15), we will have:

$$A_3^2 = \left(b_{12}\left(A_3^1 - b_{13}\right) + \left(A_2^1 - b_{12}\right) b_{13}\right).$$

Then, by solving for b_{13}, the above equation gives $b_{13} = \frac{A_3^2 - b_{12}A_3^1}{A_2^1 - 2b_{12}}$. Similarly all the other parameters can be obtained.

Remark 3.1. From the above procedure, b_1 is found, and b_2 can be found from (3.13). This indicates that for a two-mode model, the parameters can be estimated without clustering any data point.

Remark 3.2. In the original AG method, for a two-mode model, two minimization problems needed to be solved, similarly, in the proposed EW method, two TLS regression problems should be solved.Therefore, the proposed method does not increase the complexity with respect to the original AG method.

Remark 3.3. Equation (3.16) has two solutions. Selecting each of them and continuing the procedure will lead to either one of the parameter sets. Since only one quadratic equation needs to be solved, this method results in a unique solution for the two-mode models. However, in the three-mode models, using the same approach and with some straightforward mathematical manipulations, one ends up solving K different polynomial equations with order three. Since one cannot determine which solution belongs to which parameter vector, the unique solution cannot be found. Therefore, further studies are needed for generalizing this new method.

Remark 3.4. After estimating the parameters, the data points can be clustered via the following minimization. Clustering the data points will reconstruct the switching sequence.

$$\lambda_t = \arg \min_{i=1,\dots,n} (b_i^T X_t)^2 \tag{3.17}$$

3.2.2 The matrix-wise approach

In this section a general solution for recovering SARX parameters from the estimated SHDP is developed. The approach given in this section is called matrix-wise (MW) approach, which show improved results compared to the original AG algorithm (the SARX parameters recovering step). Since all the data points are used in this approach, the need for matching one data point to each sub-model prior to recovering parameters is eliminated. In the derivation of the matrix-wise approach, the Kronecker product ($\otimes$) and some of its properties as well as matrix differential calculus are used.

Consider a SARX model with n sub-models in (3.2), and its corresponding SHDP as $P_n(X_t) = \prod_{i=1}^{n} (b_i^T X_t)$. The $n-1$ derivatives with respect to X_t

should be taken from (3.4). Consider the first four derivatives:

$$
\begin{aligned}
D^1P_n(X_t) &= \frac{\partial P_n(X_t)}{\partial X_t} = \sum_{i_1=1}^{n}\left(b_{i_1}\times\prod_{i_2\neq i_1}^{n}(b_{i_2}^TX_t)\right)\\
D^2P_n(X_t) &= \frac{\partial^2 P_n(X_t)}{\partial X_t\partial x_t^T} = \sum_{i_1,i_2\in M={}_nP_2}\left(b_{i_1}\times b_{i_2}^T\times\prod_{i_3\in M^c}(b_{i_3}^TX_t)\right)\\
D^3P_n(X_t) &= \frac{\partial^3 P_n(X_t)}{\partial X_t\partial x_t^T\partial X_t} = \sum_{i_1,i_2,i_3\in M={}_nP_3}\left((b_{i_1}\times b_{i_2}^T)\otimes b_{i_3}\times\prod_{i_4\in M^c}(b_{i_4}^TX_t)\right)\\
D^4P_n(X_t) &= \frac{\partial^4 P_n(X_t)}{\partial X_t\partial x_t^T\partial X_t\partial x_t^T}\\
&= \sum_{i_1,i_2,i_3,i_4\in M={}_nP_4}\left((b_{i_1}\times b_{i_2}^T)\otimes(b_{i_3}\times b_{i_4}^T)\times\prod_{i_5\in M^c}(b_{i_5}^TX_t)\right)
\end{aligned}
\tag{3.18}
$$

where $M = {}_nP_k$ is a set of all possible combinations in selecting k numbers from n numbers where the order is important and M^c is the complement set of M. Since b_{i_j} is a vector, $b_{i_j}\otimes b_{i_k}^T = b_{i_j}\times b_{i_k}^T$. Therefore, the $(n-1)^{th}$ derivative of $P_n(X_t)$ when n is an odd number is as follows:

$$
\begin{aligned}
D^{n-1}P_n(X_t) &= \frac{\partial^{n-1}P_n(X_t)}{\partial X_t\partial x_t^T\partial X_t\partial x_t^T\cdots\partial X_t\partial x_t^T}\\
&= \sum_{i_1\cdots i_{n-1}\in M={}_nP_{n-1}}\left(\coprod_{k\in\{1,\,3,\,\cdots,\,n-2\}}(b_{i_k}b_{i_{k+1}}^T)\times(b_{i_n\in M^c}^TX_t)\right)
\end{aligned}
\tag{3.19}
$$

If n is an even number we have:

$$
\begin{aligned}
D^{n-1}P_n(X_t) &= \frac{\partial^{n-1}P_n(X_t)}{\partial X_t\partial x_t^T\partial X_t\partial x_t^T\cdots\partial X_t}\\
&= \sum_{i_1\cdots i_{n-1}\in M={}_nP_{n-1}}\left(\coprod_{k\in\{1,\,3,\,\cdots,\,n-3\}}(b_{i_k}b_{i_{k+1}}^T)\otimes b_{i_{n-1}}\times(b_{i_n\in M^c}^TX_t)\right),
\end{aligned}
\tag{3.20}
$$

where $\coprod$ is defined in Section 2.2. It can be seen that each term in (3.19) and (3.20) is linear with respect to (X_t). Both derivatives are matrix-wise equations. First, assume that the number of modes is odd. By considering each element of (3.19), the following element-wise equations for the odd n is

obtained:

$$\begin{aligned} D_{lm}^{n-1}P_n(X_t) &= \sum_{i_1\cdots i_{n-1}\in M=_nP_{n-1}} \left[\coprod_{k\in\mathcal{G}}(b_{i_k}b_{i_{k+1}}^T)\times(b_{i_n\in M^c}^T X_t)\right]_{lm} \\ &= \sum_{i_1\cdots i_{n-1}\in M=_nP_{n-1}} \left[\coprod_{k\in\mathcal{G}}(b_{i_k}b_{i_{k+1}}^T)\times(b_{i_n\in M^c}^T X_t) - \coprod_{k\in\mathcal{G}}(b_{i_k}b_{i_{k+1}}^T)\right]_{lm} \\ &= \sum_{i_1\cdots i_{n-1}\in M=_nP_{n-1}} \left[\coprod_{k\in\mathcal{G}}(b_{i_k}b_{i_{k+1}}^T)\times(b_{i_n\in M^c}^T X_t)\right]_{lm} \end{aligned} \tag{3.21}$$

where $[.]_{lm}$ is the element of the matrix at row l and column m, and $\mathcal{G} \equiv \{1,\ 3,\ \cdots,\ n-2\}$. The above scalar equation should hold for all data points. Therefore, by taking the transpose of (3.21) and using all data points, we have:

$$\begin{aligned} &\begin{bmatrix} D_{lm}^{n-1}P_n(X_1) \\ \vdots \\ D_{lm}^{n-1}P_n(X_N) \end{bmatrix} = \\ &\begin{bmatrix} X_1^T \\ \vdots \\ X_N^T \end{bmatrix} \sum_{i_1\cdots i_{n-1}\in M=_nP_{n-1}} \left[b_{i_n\in M^c}\times\coprod_{k\in\mathcal{G}}(b_{i_k}b_{i_{k+1}}^T)\right]_{lm}. \end{aligned} \tag{3.22}$$

The above is linear regression problem with the regressors matrix as $\left[\begin{bmatrix} X_1^T \\ \vdots \\ X_N^T \end{bmatrix} \begin{bmatrix} D_{lm}^{n-1}P_n(X_1) \\ \vdots \\ D_{lm}^{n-1}P_n(X_N) \end{bmatrix}\right]$. Therefore by using TLS solution the following parameter vector can be estimated:

$$\sum_{i_1\cdots i_{n-1}\in M=_nP_{n-1}} \left[b_{i_n\in M^c}\times\coprod_{k\in\mathcal{G}}(b_{i_k}b_{i_{k+1}}^T)\right]_{lm} = A_{lm}. \tag{3.23}$$

The above equation is a vector-wise equation. Repeating the same procedure for all the elements in (3.19) and combining all the resultant vector-wise equations into one matrix-wise equation results in the following equation:

$$\sum_{i_1\cdots i_{n-1}\in M=_nP_{n-1}} \left(\coprod_{k\in\mathcal{G}}(b_{i_k}b_{i_{k+1}}^T)\otimes b_{i_n\in M^c}^T\right) = \begin{bmatrix} A_{11}^T & \cdots & A_{1n}^T \\ A_{21}^T & \cdots & A_{2n}^T \\ \vdots & \ddots & \vdots \\ A_{n1}^T & \cdots & A_{nn}^T \end{bmatrix} = \Lambda. \tag{3.24}$$

Also consider the following equation:

$$\sum_{i_1\cdots i_{n-1}\in M={}_nP_{n-1}} \left(\coprod_{k\in\mathcal{G}} (b_{i_k} b_{i_{k+1}}^T) \otimes b_{i_{n\in M^c}}^T \right) = \sum_{i_1\cdots i_{n-1}\in M={}_n\acute{P}_{n-1}} \left(\coprod_{k\in\mathcal{G}} (b_{i_k} b_{i_{k+1}}^T + b_{i_{k+1}} b_{i_k}^T) \otimes b_{i_{n\in M^c}}^T \right) \tag{3.25}$$

where $M = {}_n\acute{P}_k$ is the set of all possible combinations of selecting k numbers from n numbers when the order is important between each pair of numbers but not between two numbers in the pairs. Equation (3.24) can be rewritten as:

$$\sum_{i_1\cdots i_{n-1}\in M={}_n\acute{P}_{n-1}} \left(\coprod_{k\in\mathcal{G}} (BC_{i_k i_{k+1}} B^T) \otimes e_{i_{n\in M^c}}^T B^T \right) = \Lambda \tag{3.26}$$

where e_{i_j} is a unit vector with length n and its i_j^{th} element is equal to one and other elements equal to zero, $B = [b_1, \cdots, b_n]$, and $\begin{bmatrix} e_{i_k} & e_{i_{k+1}} \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} e_{i_k}^T \\ e_{i_{k+1}}^T \end{bmatrix} = C_{i_k i_{k+1}}$.

Using *Lemma 2.1* with the above equation results in:

$$\sum_{i_1\cdots i_{n-1}\in M={}_n\acute{P}_{n-1}} \left((\coprod_{l=1}^{(n-1)/2} B) \times (\coprod_{k\in\mathcal{G}} C_{i_k i_{k+1}}) \times (\coprod_{l=1}^{(n-1)/2} B^T) \otimes e_{i_{n\in M^c}}^T B^T \right) = \Lambda \tag{3.27}$$

By using the mixed product property of Kronecker product and equation (3.27) the generalized matrix-wise equation is obtained (when n is odd).

$$(\coprod_{l=1}^{(n-1)/2} B) \times \left(\sum_{i_1\cdots i_{n-1}\in M={}_n\acute{P}_{n-1}} (\coprod_{k\in\mathcal{G}} C_{i_k i_{k+1}}) \otimes e_{i_{n\in M^c}}^T \right) \times (\coprod_{l=1}^{(n+1)/2} B^T) = \Lambda$$
$$\text{S.T.:} \quad B_{1i} = 1 \quad i \in 1, \cdots, n \tag{3.28}$$

Following the same approach when n is an even number, (3.20) will result

in:

$$\sum_{i_1\cdots i_{n-1}\in M=_n\acute{P}_{n-1}}\left(\coprod_{k\in\{1,\,3,\,\cdots,\,n-3\}}(BC_{i_ki_{k+1}}B^T)\otimes(Be_{i_{n-1}}\otimes e^T_{i_n\in M^c}B^T)\right)=\Lambda \tag{3.29}$$

Above equation can be rewritten as:

$$\sum_{i_1\cdots i_n\in M=_n\acute{P}_n}\left(\coprod_{k\in\{1,\,3,\,\cdots,\,n-1\}}(BC_{i_ki_{k+1}}B^T)\right)=\Lambda \tag{3.30}$$

Finally the generalized matrix-wise equation when n is even can be found as:

$$\begin{aligned}&(\coprod_{l=1}^{n/2}B)\times\left(\sum_{i_1\cdots i_n\in M=_n\acute{P}_n}(\coprod_{k\in\{1,\,3,\,\cdots,\,n-1\}}C_{i_ki_{k+1}})\right)\times(\coprod_{l=1}^{(n)/2}B^T)=\Lambda\\&\text{S.T.:}\quad B_{1i}=1\quad i\in 1,\cdots,n\end{aligned} \tag{3.31}$$

The solutions to equations (3.28) and (3.31) are the parameter vectors.

Remark 3.5: For the two-mode models, (3.31) is simplified to:

$$\begin{aligned}&B\times\begin{bmatrix}0&1\\1&0\end{bmatrix}\times B^T=\Lambda,\\&\text{s.t.: } B_{11}=B_{12}=1\end{aligned} \tag{3.32}$$

for which K TLS problems need to be solved.

Remark 3.6: Analytical solutions for equations (3.28) and (3.31) are generally difficult to find if not impossible, due to the complexity created by the product of the matrices. In this case, iterative numerical methods can be used. In the simulation section, two numeric methods including the Newton method and the particle swarm optimization (PSO) method are utilized. The PSO method [57], which is an evolutionary optimization method, is used to find an appropriate initial point. Then the Newton method is applied based on this initial point. Combination of these two ensures faster convergence to the solution. The PSO's parameters are set by the guidelines discussed in

[58]. A toolbox for MATLAB has been developed for the PSO algorithm [59]. This toolbox is used for solving the generalized Matrix-wise equations in our simulations.

Remark 3.7: In three-mode models, (3.28) can be simplified to:

$$\begin{aligned} & B \times [C_{23}\ C_{13}\ C_{12}] \times \left(B^T \otimes B^T\right) = \Lambda, \\ & \text{s.t.: } B_{11} = B_{12} = B_{13} = 1 \end{aligned} \tag{3.33}$$

Remark 3.8: The EW or MW approaches can be used independently in the second step of the original AG algorithm. Hence, instead of recovering the SARX parameters from SHDP, MW or EW approaches can be applied to recover these parameters from HDP used in original AG algorithm.

3.2.3 Simulation results

In this section, some of the simulation results are discussed. Firstly, the identification results obtained by original AG approach, EW approach and MW approach for two two-mode systems will be presented. The experimental results obtained from EW approach will also be shown.

Simulation results for a two-mode model

Two 2-mode systems are used in the simulations. The first system is from [56]. This system has been used in several references such as [60],

$$\text{system 1:} \begin{cases} y_i = 2u_{i-1} + 10 + w(t) & u_{i-1} < 0 \\ y_i = -1.5u_{i-1} + 10 + w(t) & u_{i-1} \geq 0 \end{cases}$$

The second system, which is more complex is:

$$\text{system 2:} \begin{cases} y_i = -0.5y_{i-1} + 0.3y_{i-2} + u_{i-1} + 5 + w(t) & y_{i-1} \geq 0 \\ y_i = 0.7y_{i-1} - 0.4y_{i-2} + 2u_{i-1} - 3 + w(t) & y_{i-1} < 0 \end{cases}$$

In both systems, u has uniform distribution in the range of $[-10\ 10]$, and $y(0)$ is -10. $\omega(t)$ is Gaussian noise with zero mean and varying variances. The noise variances used in the simulations are $\sigma^2 = [0.01\ 0.04\ 0.1\ 0.25]$. Simulations have been run 100 times for each noise variance. In each run, the generated noise is initialized with a random number in order to prevent any replication

in MATLAB. In the MW approach, the initialization of the PSO algorithm is a random number in the range of $[-20\ 20]$.Also, the number of particles and the number of iterations are assumed to be 50 and 1000 respectively. As mentioned before, the Newton method uses the value found by PSO as the starting point. The error between the estimated parameter $\hat{b}$ and the actual parameters b is computed as:

$$\text{error} = \max_{i=1,\cdots,n} \min_{j=1,\cdots,n} \frac{||\hat{b_i} - b_j||}{||b_j||}.$$

The accuracy of the clustering is computed by using (3.17). Simulation results are presented in Fig. 3.1. This figure reveals that the overall performances of the new approaches are better than the original method. The most important observation from Fig. 3.1 is that the original approach fails to have satisfactory results when the system becomes more complex. In system 2, the difference between the performances of the new approaches and the original method is significant. Another important point is that the variance of the estimation error decreases dramatically for both systems 1 and 2 when applying the proposed methods. This advantage is mainly due to the utilization of the TLS regression. It is also noticeable that EW approach gives better results than MW approach. This is due to optimization error using PSO and Newton method.

Experimental results

In this section, an experimental setup introduced in [61] is used to show the effectiveness of proposed method on real systems. The pilot plant is a tank system with two different level controllers. Two controllers have different transient responses in set-point tracking. One of them results in a fast response with high overshoot and the other one provides slower responses with much less overshoots. Switching between two controllers is performed following a random sequence. In this experimental setup the goal is to identify the closed-loop model for the two mode hybrid system.

The orders of the sub-models are selected as $n_\alpha = 2, n_\beta = 2$, which is the same as the ones used in [61]. By applying the proposed EW approach to the de-trended data collected from the plant, the following parameters are

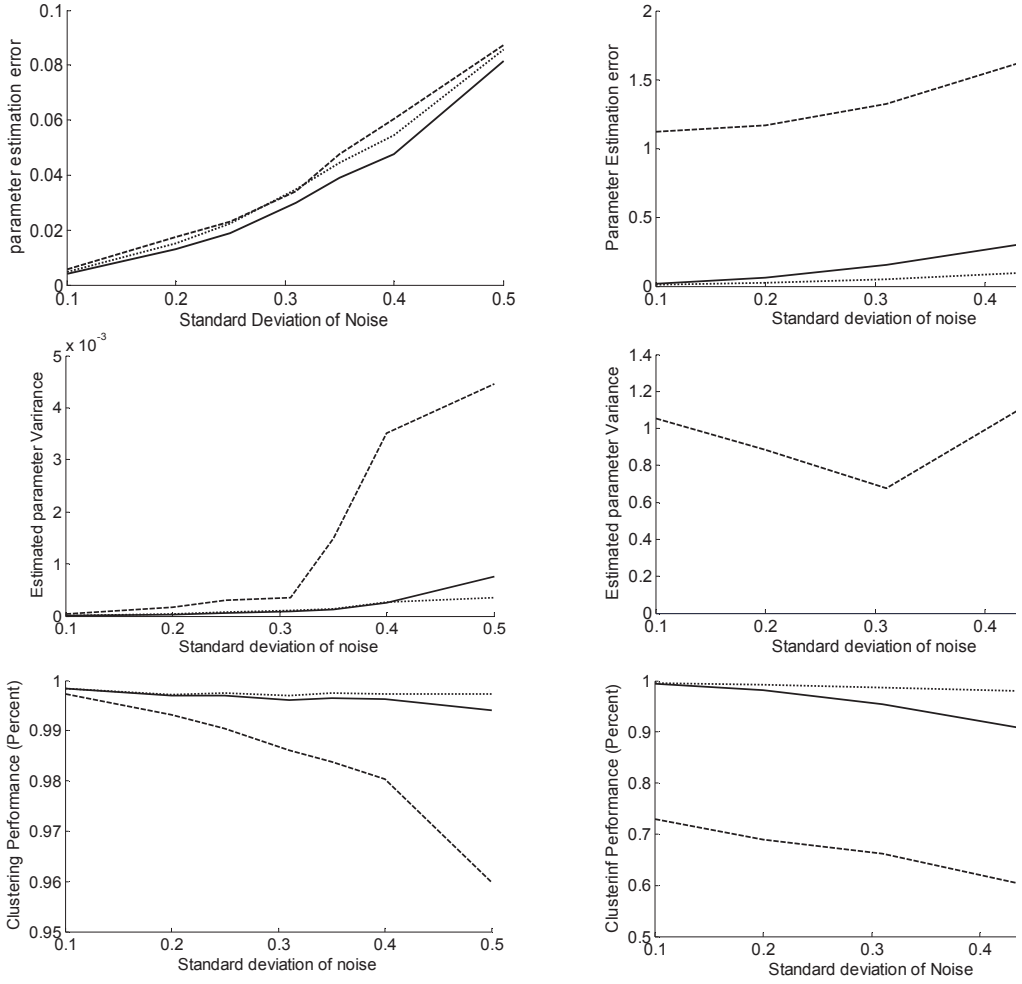

Figure 3.1: Simulation results for two systems with two modes. The left is System 1 and the right is System 2. EW (Doted line), MW (Solid line) and Original approach (Dashed Line) for both systems.

estimated for the two mode system:

$$\begin{cases} y_k = 1.3547y_{k-1} - 0.4304y_{k-2} + 0.0406u_{k-1} - 0.0344u_{k-2} \\ y_k = 2.4864y_{k-1} - 1.7158y_{k-2} + 0.1422u_{k-1} - 0.1552u_{k-2} \end{cases} \tag{3.34}$$

The data set available from this experiment consists of 1000 data points. The first 700 data points are used for training and estimating purposes and the rest are used for model validation. The sampling time is 3 second. The self-validation and cross-validation results of the model are shown in Fig. 3.2. As it can be seen in this figure, the model estimated by the proposed method shows a good performance in both self-validation and cross-validation tests. The results obtained by this method are improved significantly over the results reported in [61], which are obtained by using expectation maximization (EM) algorithm. The MSE of the self-validation results is 9.8×10^{-6} and the MSE for cross-validation results is 1.04×10^{-5}. Unfortunately these MSEs for the illustrated algorithm in [61] are not reported. However, for comparison the cross validation and self validation of these two algorithms are reported in figures 3.2 and 3.3. Overall this experiment shows the applicability of our proposed methods.

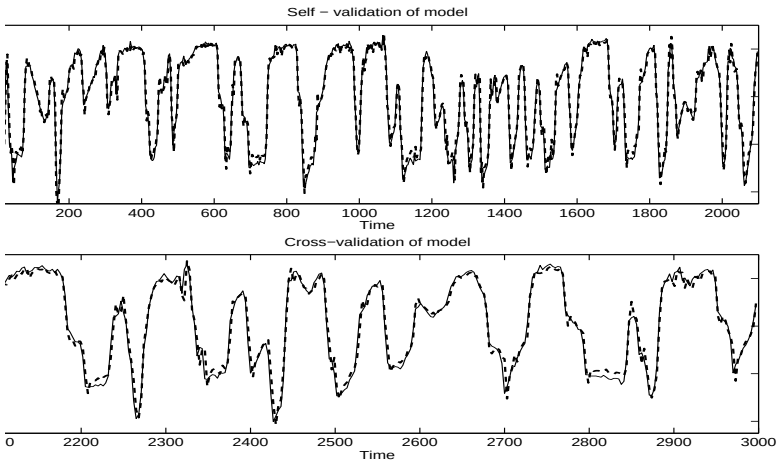

Figure 3.2: Validation results: '–' the actual output; '- -' the estimated output based on the estimated parameters

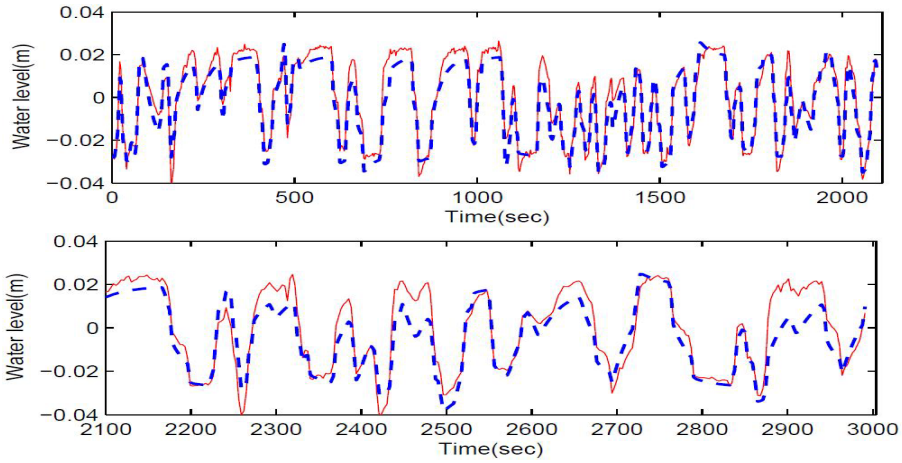

Figure 3.3: Validation results: '–' the actual output; '- -' the estimated output based on the estimated parameters from [61]

3.3 The AG approach's noise problem

The results provided in the previous section are satisfactory. The mean error in estimated parameters were less than the original AG method. The models discussed in the simulation section were two-mode models. Estimating the parameters for three-mode models leads to a larger error. Although the results obtained from MW approach (EW approach cannot be implemented on three-mode models) shows improvement from the original method, the mean error is far from acceptable range. This was a motivation to have a further study on (3.3) and the way the noise is embedded in that equation.

From (3.3), one can notice that the right hand side is the noise with unknown variance and zero mean. In the previous section, this noise was considered to be small and additive. However, this assumption has been contradicted by our simulations results. Table 3.1 shows the mean and variance of ϵ in (3.3) for system 2, described in the previous section, and Table 3.2 shows results for a three mode system with $n_\alpha = 2, n_\beta = 1$. As seen in these tables, the variances are very high especially in the three-mode model (table 3.2).

For further investigations, consider the HDP for the three-mode model:

$$P_3(X_t) = (b_1^T X_t) \times (b_2^T X_t) \times (b_3^T X_t)$$

For one data point (X_t), one of the brackets will have a small value equal to ε_i and the other two will have a possibly large values (the distance of that

Table 3.1: Mean and variance of the noise ϵ in the two-mode systems

δ_ε^2	**0.01**	**0.05**	**0.1**	**0.5**	**1**
Mean ϵ	0.004	0.046	0.17	0.54	1.22
Variance ϵ	3.093	15.83	32.15	158.29	312.91

Table 3.2: Mean and variance of the noise ϵ in the three-mode system

δ_ε^2	**0.01**	**0.05**	**0.1**	**0.5**
Mean ϵ	.94	1.19	5.54	1.67
Variance ϵ	1.53×10^4	7.79×10^4	1.54×10^5	7.93×10^5

data point from the two other hyperplanes). This means that for all the data points, the HDP will not equal to a small noise. Therefore, this problem is addressed fundamentally in the next chapter with reformulating the HDP.

Chapter 4

The Stochastic Algebraic Geometric Approach

[1] In this chapter, a new approach is introduced that considers the measurement noise in the inputs/output data, by revisiting the formulation of the hybrid decoupling polynomial (HDP). An error-in-variable (EIV) representation is adopted. This new formulation is called the stochastic HDP (SHDP) and the proposed approach is called the stochastic algebraic geometric (SAG) approach.

4.1 The problem formulation

Consider the following true switching linear dynamic model:

$$\begin{cases} X_{1t}^0 = {b_1^0}^T X_{2t}^0 \\ X_{1t}^0 = {b_2^0}^T X_{2t}^0 \\ \vdots \\ X_{1t}^0 = {b_n^0}^T X_{2t}^0 \end{cases}, \quad X_{1t}^0 = x_{1t}^0 \in \mathbb{R}^{1\times 1}, \;\; X_{2t}^0 = \begin{bmatrix} x_{2_1t}^0 \\ x_{2_2t}^0 \\ \vdots \\ x_{2_mt}^0 \end{bmatrix} \in \mathbb{R}^{m\times 1} \;\; b_i^0 \in \mathbb{R}^{m\times 1} \tag{4.1}$$

where b_i^0s are the true parameter vectors for each sub-model at the mode i, n is the number of the sub-models in this multi-mode model and m is the maximum number of the parameters among all the parameter sets. Also X_{1t}^0, X_{2t}^0 are the output and inputs of a physical model before corrupting with measurement

[1] A version of this chapter is under preparation to be submitted to a journal.

noise. $X_t^0 = \begin{bmatrix} X_{1t}^0 \\ X_{2t}^0 \end{bmatrix}$, and X^0 are the noise-free data matrix defined as:

$$X^0 = [X_1^0 \;\; X_2^0] = \begin{bmatrix} x_{11}^0 & x_{2_1 1}^0 & x_{2_2 1}^0 & \cdots & x_{2_m 1}^0 \\ x_{12}^0 & x_{2_1 2}^0 & x_{2_2 2}^0 & \cdots & x_{2_m 2}^0 \\ \vdots & \vdots & \vdots & & \vdots \\ x_{1t}^0 & x_{2_1 t}^0 & x_{2_2 t}^0 & \cdots & x_{2_m t}^0 \\ \vdots & \vdots & \vdots & & \vdots \\ x_{1N}^0 & x_{2_1 N}^0 & x_{2_2 N}^0 & \cdots & x_{2_m N}^0 \end{bmatrix}, \tag{4.2}$$

where,

$$X_1^0 = \begin{bmatrix} x_{11}^0 \\ x_{12}^0 \\ \vdots \\ x_{1t}^0 \\ \vdots \\ x_{1N}^0 \end{bmatrix}, \qquad X_2^0 = \begin{bmatrix} x_{2_1 1}^0 & x_{2_2 1}^0 & \cdots & x_{2_m 1}^0 \\ x_{2_1 2}^0 & x_{2_2 2}^0 & \cdots & x_{2_m 2}^0 \\ \vdots & \vdots & & \vdots \\ x_{2_1 t}^0 & x_{2_2 t}^0 & \cdots & x_{2_m t}^0 \\ \vdots & \vdots & & \vdots \\ x_{2_1 N}^0 & x_{2_2 N}^0 & \cdots & x_{2_m N}^0 \end{bmatrix}, \tag{4.3}$$

and $x_{jt}^0, \;\; j \in \{1 \;\; , 2_1 \;\; 2_2 \;\; \cdots \;\; 2_m\}$, are noise-free regressors at time t. Due to the fact that the system is switching among n modes, only one of the above sub-models is the true representation of the system at the time t, or only one of the equations above is satisfied at the time t. Alternative representation of the above multi-model is as follows.

$$X_{1t}^0 = {b_{\lambda_t}^0}^T X_{2t}^0 \tag{4.4}$$

where $\lambda_t = 1, 2, \cdots, n$ is the indicator for the sub-model at time t that is active in this multi-mode model. The λ_t is also called the discrete switching sequence among the sub-models. The inputs, outputs or any arbitrary exogenous cause from outside of the system(e.g. time, random sequence) can change the value of λ_t at time t. Therefore the switching sequence is not necessarily dependent on the regressors.

In a SISO dynamic system, the X_{1t}^0 contains the true output value of the

system and X_{2t}^0 vector contains the true back-shifted outputs and inputs:

$$X_{1t}^0 = y_t^0, \quad X_{2t}^0 = \begin{bmatrix} x_{2_1t}^0 \\ x_{2_2t}^0 \\ \vdots \\ x_{2_mt}^0 \end{bmatrix} = \begin{bmatrix} y_{t-1}^0 \\ \vdots \\ y_{t-n_\alpha(i)}^0 \\ u_{t-1}^0 \\ \vdots \\ u_{t-n_\beta(i)}^0 \\ 1 \end{bmatrix}, \tag{4.5}$$

where y_{t-i}^0 and u_{t-j}^0 are the true value regressors before being corrupted with noise. The identification problem for (4.1) is defined as follow:

Identification Problem: Consider the available inputs and outputs measurements $X = [X_1 \quad X_2] = [X_1^0 + \Delta X_1 \quad X_2^0 + \Delta X_2]$. Assuming that X is persistent enough and the modes are visited frequently enough, the identification problem is to find b_i as an estimation to b_i^0. In addition, the switching sequence λ_t needs to be identified.

The matrix $\Delta X = [\Delta X_1 \quad \Delta X_2]$ is the measurement noise matrix. The following assumptions are used through the rest of the thesis:

i) The number of modes n is previously known.

ii) The univariate case is considered ($l = 1$) although the developed results can be easily extended to multivariate case. Therefore $X_{1t} = x_{1t}$ and $X_{2t} = [x_{2_1t} \;\; x_{2_2t} \;\; \cdots \;\; x_{2_mt}]^T$ respectively. Hence, $X_t = \begin{bmatrix} X_{1t} \\ X_{2t} \end{bmatrix}$.

iii) The inputs and output are perturbed by independent additive measurement gaussian noises with zero mean and constant variances. The variance of each noise source can be different. Therefore the covariance matrix of
$\Delta X_t = [\Delta x_{1t} \quad \Delta x_{2_1t} \quad \Delta x_{2_2t} \quad \cdots \quad \Delta x_{2_mt}]^T$ is the following diagonal matrix:

$$Cov(\Delta X_t) = \text{Diag}(\sigma_{x_1}^2, \; \sigma_{x_{2_1}}^2, \; \cdots, \; \sigma_{x_{2_m}}^2) \tag{4.6}$$

Finally, the error-in-variable (EIV) representation of the model (4.1) is:

$$\begin{cases} X_{1t} \approx {b_1}^T X_{2t} \\ X_{1t} \approx {b_2}^T X_{2t} \\ \vdots \\ X_{1t} \approx {b_n}^T X_{2t} \end{cases} \tag{4.7}$$

All the sub-models in (4.7) are linear regression models and parameter estimation for each of them individually is possible using the total least square methods discussed in Section 2.1.2. However, the identification is dependent on the switching sequence as there is no information about which data belongs to which sub-model. The lack of this information increases the identification complexity dramatically. Therefore, constructing a model that is independent of switching sequence λ_t is a promising way to circumvent this complexity.

Assuming that $\lambda_t = k,\ k \in \{1, \cdots, n\}$ for X_t^0 at time t, therefore in (4.1) $X_{1t}^0 - {b_k^0}^T X_{2t}^0 = 0$. Taking the product of all the equations of all the sub-models in (4.1) and embedding them into one higher order equation ensures that all the data points satisfy the following polynomial equation [32]:

$$P_n(X_t^0) = \prod_{i=1}^{n} (X_{1t}^0 - b_i^T X_{2t}^0) = 0. \tag{4.8}$$

The right hand side of the above multiplication is zero, since one of the brackets is equal to zero. P_n is called the hybrid decoupling polynomial (HDP). As discussed in Section 3.1.1, the HDP is a multivariate polynomial of degree n with $K = m + 1$ variables, which can be written linearly in terms of its coefficients as:

$$P_n(X_t^0) = \sum_{I=1}^{M_n(K)} h_I^0 z_{It}^0 = {H^0}^T \vartheta_n(X_t^0) = 0. \tag{4.9}$$

where, $\vartheta_n : \mathbb{R}^K \rightarrow \mathbb{R}^{M_n(K)}$ is a Veronese map of degree n [55] , which is defined as $\vartheta_n : [x_{1t} \ \ \cdots \ \ x_{Kt}]^T \rightarrow [\cdots \ z_{It} \ \cdots]^T$ with I chosen in the degree-lexicographic order. Also, $h_I \in \mathbb{R}$ is the coefficient of the monomial

$$z_{It}^0 = (x_{1t}^0)^{n_{1I}} (x_{2_1 t}^0)^{n_{2_1 I}} \cdots (x_{2_m t}^0)^{n_{2_m I}},$$

where $0 \leq n_{jI} \leq n,\ j \in \{1, 2_1 \cdots, 2_m\}$, and $n_{1I} + n_{2_1 I} + \cdots + n_{2_m I} = n,\ \ I = 1,\ \cdots,\ M_n(K)$, and $M_n(K) = \binom{n+K-1}{n}$ is the total number of independent monomials in (4.9). e.g.

$$\begin{aligned}\vartheta_2([x_{1t}^0 \ \ x_{2_1 t}^0 \ \ x_{2_2 t}^0]^T) &= \left[{x_{1t}^0}^2 \ \ x_{1t}^0 x_{2_1 t}^0 \ \ x_{1t}^0 x_{2_2 t}^0 \ \ {x_{2_1 t}^0}^2 \ \ x_{2_1 t}^0 x_{2_2 t}^0 \ \ {x_{2_2 t}^0}^2\right]^T \\ &= [z_{1t}^0 \ \ z_{2t}^0 \ \ z_{3t}^0 \ \ z_{4t}^0 \ \ z_{5t}^0 \ \ z_{6t}^0]^T = Z_t^0.\end{aligned}$$

Equations (4.8) and (4.9) hold for all the data points, therefore the following equation represents a linear regression model with the true parameter vector H^0:

$$H^{0^T}\left[\vartheta_n(X_1^0) \ \ \cdots \ \ \vartheta_n(X_t^0) \ \ \cdots \ \ \vartheta_n(X_N^0)\right] = H^{0^T} Z^{0^T} = \mathbf{0}_{1\times N}, \tag{4.10}$$

where N is the number of data points, and $Z^0 \in \mathbb{R}^{N \times M_n(K)}$ is the matrix of the embedded and mapped true input/output data via the Veronese variety

$$Z^0 = [Z_1^0 \ \ Z_2^0 \ \ \cdots \ \ Z_{M_n(K)}^0] = \begin{bmatrix} z_{11}^0 & z_{12}^0 & \cdots & z_{I1}^0 & \cdots & z_{M_n(K)1}^0 \\ z_{12}^0 & z_{22}^0 & \cdots & z_{I2}^0 & \cdots & z_{M_n(K)2}^0 \\ \vdots & \vdots & & \vdots & & \vdots \\ z_{1t}^0 & z_{2t}^0 & \cdots & z_{It}^0 & \cdots & z_{M_n(K)t}^0 \\ \vdots & \vdots & & \vdots & & \vdots \\ z_{1N}^0 & z_{2N}^0 & \cdots & z_{IN}^0 & \cdots & z_{M_n(K)N}^0 \end{bmatrix}, \tag{4.11}$$

where $Z_I^0 = \begin{bmatrix} z_{I1}^0 \\ z_{I2}^0 \\ \vdots \\ z_{It}^0 \\ \vdots \\ z_{IN}^0 \end{bmatrix}$, and $Z_t^0 = \begin{bmatrix} z_{1t}^0 \\ z_{2t}^0 \\ \vdots \\ z_{It}^0 \\ \vdots \\ z_{M_n(K)t}^0 \end{bmatrix}$.

In (4.10) the switching sequence is eliminated completely and the true parameter vector H^0 can be estimated using regression methods without knowing the λ_t. In order to have a unique parameter vector, the first element of parameter vector H^{0^T} is always assumed to be one. By defining $\tilde{H}^{0^T}$ so that

$H^{0^T} = [1 \quad -\tilde{H}^{0^T}]$, (4.10) is changed to:

$$H^{0^T} Z^{0^T} = [1 \quad -\tilde{H}^{0^T}] \begin{bmatrix} Z_1^{0^T} \\ Z_2^{0^T} \\ \vdots \\ Z_I^{0^T} \\ \vdots \\ Z^{0^T}_{M_n(K)} \end{bmatrix} = \mathbf{0} \tag{4.12}$$

Therefore, the true SARX model can be represented by the following system of linear equations similar to (2.1):

$$Z_1^{0^T} = \tilde{H}^{0^T} \begin{bmatrix} Z_2^{0^T} \\ \vdots \\ Z_I^{0^T} \\ \vdots \\ Z^{0^T}_{M_n(K)} \end{bmatrix} \tag{4.13}$$

Linear regression methods can be used for estimating the parameter vector $\tilde{H}^0$ in the above equation. Hence, the identification of the multi-mode model (4.1) can be achieved in two consecutive steps. In the first step, the embedded model (HDP) is constructed and its parameter vector H^0 is estimated. In the second step, the sub-model parameters (b_i^0) are retrieved from the estimated HDP.

Let us introduce an illustrative example that will be used through the rest of this chapter.

Illustrative Example 4.1: Consider the following two-mode model:

$$\begin{cases} x_{1t}^0 = \alpha_1^0 x_{2_1 t}^0 + \beta_1^0 x_{2_2 t}^0 \\ x_{1t}^0 = \alpha_2^0 x_{2_1 t}^0 + \beta_2^0 x_{2_2 t}^0 \end{cases} \tag{4.14}$$

The HDP for the above example will be:

$$\begin{aligned}&(x_{1t}^0 - \alpha_1^0 x_{2_1t}^0 - \beta_1^0 x_{2_2t}^0) \times (x_{1t}^0 - \alpha_2^0 x_{2_1t}^0 - \beta_2^0 x_{2_2t}^0)\\ &= {x_{1t}^0}^2 + (-\alpha_1^0 - \alpha_2^0) x_{1t}^0 x_{2_1t}^0 + (-\beta_1^0 - \beta_2^0) x_{1t}^0 x_{2_2t}^0 + (\alpha_1^0 \alpha_2^0) {x_{2_1t}^0}^2\\ &+ (\alpha_1^0 \beta_2^0 + \alpha_2^0 \beta_1^0) x_{2_1t}^0 x_{2_2t}^0 + (\beta_1^0 \beta_2^0) {x_{2_2t}^0}^2\\ &= 0\end{aligned} \tag{4.15}$$

Therefore

$$z_t^0 = [z_{1t}^0 \;\; z_{2t}^0 \;\; z_{3t}^0 \;\; z_{4t}^0 \;\; z_{5t}^0 \;\; z_{6t}^0]^T = [{x_{1t}^0}^2 \;\; x_{1t}^0 x_{2_1t}^0 \;\; x_{1t}^0 x_{2_2t}^0 \;\; {x_{2_1t}^0}^2 \;\; x_{2_1t}^0 x_{2_2t}^0 \;\; {x_{2_2t}^0}^2]^T$$

and correspondingly,

$${H^0}^T = [1 \;\; (-\alpha_1^0 - \alpha_2^0) \;\; (-\beta_1^0 - \beta_2^0) \;\; (\alpha_1^0 \alpha_2^0) \;\; (\alpha_1^0 \beta_2^0 + \alpha_2^0 \beta_1^0) \;\; (\beta_1^0 \beta_2^0)].$$

Also note that $H^0(1) = 1$.

In this chapter we focus on estimating the parameters for HDP when regressors are corrupted with the measurements noise. The proposed approach in this chapter complies with the parameter recovering procedures developed in the Section 2.3. They can be readily applied to retrieve the SARX sub-models' parameters.

4.2 The stochastic hybrid decoupling polynomial

When the data is corrupted with measurement noise, the equation (4.8) is not valid anymore. In order to make the right hand side of (4.8) equal to zero, the influence of measurement noise has to be taken into account [31]. In this case, the regressors containing noises are represented by $X = X^0 + \Delta X$. Denote $X^0 = X - \Delta X$, the equation (4.8) becomes:

$$P_n(X_t^0) = P_n(X_t - \Delta X_t) = \prod_{i=1}^{n}((X_{1t} - \Delta X_{1t}) - b_i^T(X_{2t} - \Delta X_{2t})) = 0. \tag{4.16}$$

Taking X_t as the main argument, the HDP is denoted by $SP_n(X_t)$, i.e.:

$$\begin{aligned} SP_n(X_t) &= \prod_{i=1}^{n}((X_{1t} - \Delta X_{1t}) - b_i^T(X_{2t} - \Delta X_{2t})) \\ &= \sum_{I=1}^{M_n(K)} h_I^0 z_{It}^0 = H^{0^T}\vartheta_n(X_t - \Delta X_t) = 0. \end{aligned} \tag{4.17}$$

Since $SP_n(X_t)$ accounts for the measurements noise inside the HDP, for further referrals, it is called stochastic hybrid decoupling polynomial (SHDP).

The SHDP can also be written as a multivariate polynomial of degree n and $K = m + 1$ variables. The monomials in SHDP have the form:

$$z_{It}^0 = (x_{1t} - \Delta x_{1t})^{n_{1I}}(x_{2_1t} - \Delta x_{2_1t})^{n_{2_1I}} \cdots (x_{2_mt} - \Delta x_{2_mt})^{n_{2_mI}}, \tag{4.18}$$

where $n_{1I} + n_{2_1I} + \cdots + n_{2_mI} = n, \quad I = 1, \cdots, M_n(K)$. The structure of the coefficients of the monomials in the SHDP is as same as the monomials in the HDP. Therefore the parameter vector for SHDP is also the true parameter vector H^0. Multiplying the brackets in (4.18) results in:

$$\begin{aligned} z_{It}^0 &= (x_{1t} - \Delta x_{1t})^{n_{1I}}(x_{2_1t} - \Delta x_{2_1t})^{n_{2_1I}} \cdots (x_{2_mt} - \Delta x_{2_mt})^{n_{2_mI}} \\ &= ((x_{1t})^{n_{1I}}(x_{2_1t})^{n_{2_1I}} \cdots (x_{2_mt})^{n_{2_mI}}) - \psi_I(X_t, \Delta X_t) \\ &= z_{It} - \psi_I(X_t, \Delta X_t) = z_{It} - \Delta z_{It}, \end{aligned} \tag{4.19}$$

where $\psi_I(X_t, \Delta X_t)$ is a multivariate polynomial of variables $x_{1t}, \ x_{2_1t}, \ \cdots \ x_{2_mt}$ and $\Delta x_{1t}, \ \Delta x_{2_1t}, \ \cdots \ \Delta x_{2_mt}$. Hence, $z_{It} = z_{It}^0 + \Delta z_{It}$ and its vector representation is:

$$Z_t = Z_t^0 + \Delta Z_t. \tag{4.20}$$

where, $Z_t = [z_{1t} \ z_{2t} \ \cdots \ z_{It} \ \cdots \ z_{M_n(K)t}]^T = \vartheta_n(X_t)$. Using (4.20) in (4.10)

results in:

$$H^{0^T} Z^{0^T} = H^{0^T} \begin{bmatrix} {Z_1}^T - \Delta {Z_1}^T \\ {Z_2}^T - \Delta {Z_2}^T \\ \vdots \\ {Z_I}^T - \Delta {Z_I}^T \\ \vdots \\ Z^T_{M_n(K)} - \Delta Z^T_{M_n(K)} \end{bmatrix} = H^{0^T}(Z^T - \Delta Z^T) = \mathbf{0}_{1\times N}. \quad (4.21)$$

By comparing (2.1) with (4.10), and by following the EIV model definition (2.7) of (2.1), the EIV model of (4.10) is:

$$H^T Z^T \approx 0. \quad (4.22)$$

Hence, by using $H^T = [1 \quad -\tilde{H}^T]$, the EIV model of (4.13) becomes:

$${Z_1}^T \approx \tilde{H}^T \begin{bmatrix} {Z_2}^T \\ \vdots \\ {Z_I}^T \\ \vdots \\ Z^T_{M_n(K)} \end{bmatrix} \quad (4.23)$$

where, $\tilde{H}$ is the estimated model parameter vector. It is best to illustrate the derivation of SHDP by continuing the *Illustrative example 4.1* from the Section 4.1.

Illustrative example 4.2: The SHDP of the two-mode model (4.14) can be

derived as follow:

$$
\begin{aligned}
SP_2(X_t) &= \Big((x_{1t} - \Delta x_{1t}) - \alpha_1^0(x_{2_1t} - \Delta x_{2_1t}) - \beta_1^0(x_{2_2t} - \Delta x_{2_2t})\Big) \\
&\qquad \times \Big((x_{1t} - \Delta x_{1t}) - \alpha_2^0(x_{2_1t} - \Delta x_{2_1t}) - \beta_2^0(x_{2_2t} - \Delta x_{2_2t})\Big) \\
&= (x_{1t} - \Delta x_{1t})^2 + (-\alpha_1^0 - \alpha_2^0)(x_{1t} - \Delta x_{1t})(x_{2_1t} - \Delta x_{2_1t}) \\
&\quad + (-\beta_1^0 - \beta_2^0)(x_{1t} - \Delta x_{1t})(x_{2_2t} - \Delta x_{2_2t}) + (\alpha_1^0\alpha_2^0)(x_{2_1t} - \Delta x_{2_1t})^2 \\
&\quad + (\alpha_1^0\beta_2^0 + \alpha_2^0\beta_1^0)(x_{2_1t} - \Delta x_{2_1t})(x_{2_2t} - \Delta x_{2_2t}) + (\beta_1^0\beta_2^0)(x_{2_2t} - \Delta x_{2_2t})^2 \\
&= \Big(x_{1t}^2 - 2x_{1t}\Delta x_{1t} + (\Delta x_{1t})^2\Big) \\
&\quad + (-\alpha_1^0 - \alpha_2^0)\Big(x_{1t}x_{2_1t} - (x_{1t}\Delta x_{2_1t} + \Delta x_{1t}x_{2_1t}) + \Delta x_{1t}\Delta x_{2_1t}\Big) \\
&\quad + (-\beta_1^0 - \beta_2^0)\Big(x_{1t}x_{2_2t} - (x_{1t}\Delta x_{2_2t} + \Delta x_{1t}x_{2_2t}) + \Delta x_{1t}\Delta x_{2_2t}\Big) \\
&\quad + (\alpha_1^0\alpha_2^0)\Big(x_{2_1t}^2 - 2x_{2_1t}\Delta x_{2_1t} + (\Delta x_{2_1t})^2\Big) \\
&\quad + (\alpha_1^0\beta_2^0 + \alpha_2^0\beta_1^0)\Big(x_{2_1t}x_{2_2t} - (x_{2_1t}\Delta x_{2_2t} + \Delta x_{2_1t}x_{2_2t}) + \Delta x_{2_1t}\Delta x_{2_2t}\Big) \\
&\quad + (\beta_1^0\beta_2^0)\Big(x_{2_2t}^2 - 2x_{2_2t}\Delta x_{2_2t} + (\Delta x_{2_2t})^2\Big)
\end{aligned}
\tag{4.24}
$$

Therefore, the EIV model of this SHDP is:

$$
\begin{aligned}
&z_{1t} \approx \tilde{H}^T[z_{2t} \quad \cdots \quad z_{6t}]^T, \\
&Z_t = [z_{1t} \;\; z_{2t} \;\; z_{3t} \;\; z_{4t} \;\; z_{5t} \;\; z_{6t}]^T = [{x_{1t}}^2 \;\; x_{1t}x_{2_1t} \;\; x_{1t}x_{2_2t} \;\; {x_{2_1t}}^2 \;\; x_{2_1t}x_{2_2t} \;\; {x_{2_2t}}^2]^T \\
&\tilde{H}^T = [(\alpha_1 + \alpha_2) \;\; (\beta_1 + \beta_2) \;\; -(\alpha_1\alpha_2) \;\; -(\alpha_1\beta_2 + \alpha_2\beta_1) \;\; -(\beta_1\beta_2)],
\end{aligned}
\tag{4.25}
$$

and the ΔZ_t can be written as:

$$
\begin{aligned}
\Delta Z_t = [&(2x_{1t}\Delta x_{1t} - (\Delta x_{1t})^2) \;\; (x_{1t}\Delta x_{2_1t} + \Delta x_{1t}x_{2_1t} - \Delta x_{1t}\Delta x_{2_1t}) \\
&(x_{1t}\Delta x_{2_2t} + \Delta x_{1t}x_{2_2t} - \Delta x_{1t}\Delta x_{2_2t}) \;\; (2x_{2_1t}\Delta x_{2_1t} - (\Delta x_{2_1t})^2) \\
&(x_{2_1t}\Delta x_{2_2t} + \Delta x_{2_1t}x_{2_2t} - \Delta x_{2_1t}\Delta x_{2_2t}) \;\; (2x_{2_2t}\Delta x_{2_2t} - (\Delta x_{2_2t})^2)]^T
\end{aligned}
\tag{4.26}
$$

It is understood from the definition of ΔZ_t in (4.19) that the regressors error vector of the EIV model (4.23) is under the influence of the regressors X_t. This means that the covariance matrix of ΔZ_t at time t depends on the measured data points at that time. Hence, a suitable TLS method for a

consistent estimation of parameter vector $\tilde{H}$ is applied that can handle this varying regressors error covariance matrix. As discussed in the Section 2.1.2 the element-wise total least square (EW-TLS) method provides a consistent estimation in such cases. This is true, when the regressor error vector at each data point has zero mean.

Unfortunately, this is not the case for the ΔZ_t in (4.23). To overcome this problem, one can define $\check{Z}_t = Z_t - E[\Delta Z_t]$, where $E[\cdot]$ is the expected value. This change of variable ensures that $\Delta\check{Z}_t = \check{Z}_t - Z_t^0$ has zero mean:

Theorem 4.1: Assuming that Z_t is a random variable vector, then the vector $\check{Z}_t = Z_t - E[\Delta Z_t]$ has zero mean and $V_{\Delta\check{Z}_t} = V_{\Delta Z_t}$.

Proof.

$$\begin{aligned} &\check{Z}_t = Z_t - E[\Delta Z_t] \\ &\check{Z}_t = Z_t^0 + \Delta Z_t - E[\Delta Z_t] \;\; \Rightarrow \;\; \Delta\check{Z}_t = \Delta Z_t - E[\Delta Z_t] \\ &E[\Delta\check{Z}_t] = E[\Delta Z_t - E[\Delta Z_t]] = E[\Delta Z_t] - E[\Delta Z_t] = 0. \end{aligned} \tag{4.27}$$

$$\begin{aligned} V_{\Delta\check{Z}_t} &= E[(\Delta\check{Z}_t - E[\Delta\check{Z}_t])(\Delta\check{Z}_t - E[\Delta\check{Z}_t])^T] \\ &= E[(\Delta\check{Z}_t)(\Delta\check{Z}_t)^T] = E[(\Delta Z_t - E[\Delta Z_t])(\Delta Z_t - E[\Delta Z_t])^T] \\ &= V_{\Delta Z_t}. \end{aligned} \tag{4.28}$$

where, $V_{\Delta Z_t}$ is partitioned as :

$$V_{\Delta Z_t} = \begin{bmatrix} var(z_{1t}) & cov(z_{1t}, Z_t^c) \\ cov(Z_t^c, z_{1t}) & var(Z_t^c) \end{bmatrix} = \begin{bmatrix} V_{\Delta Z_t 1} & V_{\Delta Z_t 12} \\ V_{\Delta Z_t 21} & V_{\Delta Z_t 2} \end{bmatrix}, \tag{4.29}$$

and $Z_t^c = [z_{2t} \;\; \cdots \;\; z_{M_n(K)t}]^T$.

□

Therefore, the following theorem delivers a consistent estimation for $\tilde{H}$:

Theorem 4.2: Consider the available regressors X corrupted with additive measurement noise ΔX, and the mapped regressors $Z_t = \vartheta_n(X_t)$ defined in equations (4.18) - (4.21), where $\vartheta_(\cdot)$ is the Veronese map of degree n, the following minimization problem can generate a consistent estimation

of the parameters $\tilde{H}$ in the EIV model (4.23).

$$\begin{aligned} & f_t(\tilde{H}) = r_t^T(\tilde{H}) Q_t^{-1} r_t(\tilde{H}) \\ & \min_{\tilde{H}} \sum_{t=1}^{N} f_t(\tilde{H}) \end{aligned} \tag{4.30}$$

where

$$Q_t(\tilde{H}) = [1 \quad -\tilde{H}^T] \times V_{\Delta Z_t} \times \begin{bmatrix} 1 \\ -\tilde{H} \end{bmatrix}, \; r_t(\tilde{H}) = \check{z}_{1t} - \tilde{H}^T [\check{z}_{2t} \quad \cdots \quad \check{z}_{M_n(K)t}]^T.$$

Proof. It is shown in this section that in the EIV model (4.23) the additive errors are dependent on the regressors so that they have time-variant covariance matrix. Also, It is proved in **Theorem 4.1** that the $E[\check{Z}_t]$ is zero. As discussed in Section 2.1.2, when the errors' profile are variant by time (the covariance matrix of error is changing at each data point), the parameters of an EIV model (2.7) can be estimated consistently using (2.11) as the cost function [49]. This is true if the expected value of error matrix is zero and the covariance matrix of error is known up to a scalar at each data point. Therefore by using **Theorem 4.1** and also by calculating the covariance matrix of $V_{\Delta \check{Z}_t}$ in each data point the parameters $\tilde{H}$ in EIV model (4.23) can be estimated consistently. It is also important to note that $V_{\Delta \check{Z}_t} = V_{\Delta Z_t}$ as shown in **Theorem 4.1**

□

Thus, the calculation of expected value of the mapped (using Veronese variety) regressors' error $E[\Delta Z_t]$ and the covariance matrix of the mapped (using Veronese variety) regressors' error $Cov(\Delta Z_t)$ is essential for consistent estimation. Next section is dedicated to finding a general formula for calculating the mentioned expected and covariance values at each time t.

Remark 4.1: It is important to emphasize that the formulation of the SHDP does not increase the complexity or order of the existing HDP. The length of the parameter vector H is not changed. The use of the EW-TLS for finding the solution to the EIV problem (4.23) is very straightforward and is not iterative. As it is illustrated in the simulation results, using the calculated covariance matrix for the mapped regressors' errors together with the EW-TLS solution increases the accuracy of the estimated parameters significantly but without

increasing complexity.

Remark 4.2: In the section 2.1.2 some optimization methods are discussed for solving (2.11). The "fminunc" function in MATLAB can handle this optimization if the gradient of the cost function is supplied. The calculation of such a gradient is included in the Section 5.2, which is:

$$f'(\tilde{H}) = \frac{\partial f(\tilde{H})}{\partial \tilde{H}} = \sum_{t=1}^{N} (2\check{Z}_{1t} r_t^T(\tilde{H}) Q_t^{-1}(\tilde{H}) - r_t^T(\tilde{H}) Q_t^{-1}(\tilde{H})\ G_t\ Q_t^{-1}(\tilde{H}) r_t(\tilde{H})) \tag{4.31}$$

where $G_t = 2(V_{\Delta Z_t 1} - V_{\Delta Z_t 21}\tilde{H})$ and $V_{\Delta Z_t}$ is partitioned as (2.12). The initial point can be put either zero vector, or the estimated H parameter for HDP using the basic total least square method.

For the summary, the Stochastic Algebraic Geometric method for multi-mode EIV models is as follow.

Algorithm 1: *Stochastic algebraic geometric (SAG) approach for estimating SARX model parameters:*

Initialization:

1. Construct the HDP.
2. Find an initial approximation for H using TLS and estimate the parameters of the HDP.

Main Algorithm:

1. Calculate the $E[\Delta Z_t]$ using **Theorem 4.3** in the Section 4.3.1 for all the data points.
2. Calculate the $V_{\Delta Z_t}$ using **Theorem 4.4** in the Section 4.3.2 for all the data points.
3. Calculate $\check{Z}_t = Z_t - E[\Delta Z_t]$.
4. Solve the minimization (4.30) using $\check{Z}$ and $V_{\Delta Z_t}$.
5. Recover the parameters of the sub-models using one of the recovering algorithms discussed in the Section 3.2.

4.3 Calculation of expected value and covariance matrix of ΔZ_t

From the definition of the ΔZ_t in (4.19) we have:

$$\Delta z_{It} = z_{It} - (x_{1t} - \Delta x_{1t})^{n_{1I}} \cdots (x_{2_m t} - \Delta x_{2_m t})^{n_{2_m I}}, \tag{4.32}$$

where $[Z_{1t} \quad \cdots \quad Z_{M_n(K)t}]^T = \vartheta_n(X_t)$. In the EIV model (4.23), the values Z_{It} and x_{it} are the available data and considered to be deterministic at time t. These are a posterior data after the measurement happens, therefore at time t they are known constants. The measurement error Δx_{it} describes the error in the regression. Therefore at each data point at time t, $\Delta z^r_{It} = (x_{1t} - \Delta x_{1t})^{n_{1I}}(x_{2_1 t} - \Delta x_{2_1 t})^{n_{2_1 I}} \cdots (x_{2_m t} - \Delta x_{2_m t})^{n_{2_m I}}$ has a random distribution.

4.3.1 The expected value

Theorem 4.3: The expected value of the Δz_{It} having the available data point X_t and Z_t at time t is:

$$\begin{aligned} E[\Delta z_{It}] = z_{It} - \Big(& E[(x_{1t} - \Delta x_{1t})^{n_{1I}}] \\ & E[(x_{2_1 t} - \Delta x_{2_1 t})^{n_{2_1 I}}] \\ & \cdots E[(x_{2_m t} - \Delta x_{2_m t})^{n_{2_m I}}] \Big) \end{aligned} \tag{4.33}$$

where, $n_{1I} + n_{2_1 I} + \cdots + n_{2_m I} = n, \;\; I = 1, \; \cdots, \; M_n(K)$.

Proof. Using the definition of ΔZ_t in (4.32) we have:

$$E[\Delta z_{It}] = E[z_{It} - (x_{1t} - \Delta x_{1t})^{n_{1I}}(x_{2_1 t} - \Delta x_{2_1 t})^{n_{2_1 I}} \cdots (x_{2_m t} - \Delta x_{2_m t})^{n_{2_m I}}]. \tag{4.34}$$

As discussed earlier $E[z_{It}] = z_{It}$, and from the assumption *iii* in the Section 4.1 that ensures the independency of any Δx_{it} and Δx_{jt} the proof follows:

$$E[\psi_i(x_{it}, \Delta x_{it})\psi_j(x_{jt}, \Delta x_{jt})] = E[\psi_i(x_{it}, \Delta x_{it})]E[\psi_j(x_{jt}, \Delta x_{jt})]. \tag{4.35}$$

□

It is important to note that calculating the $E[(x_{it} - \Delta x_{it})^{n_{iI}}]$ is a very straightforward procedure and does not increase any complexity. They can be calculated readily by using **Lemma 2.2** in Section 2.3. This will be shown in a illustrative example at the end of this section.

4.3.2 The covariance matrix

Theorem 4.4: The ij^{th} element of the covariance matrix ΔZ_t with the available data X_t and Z_t at time t is calculated as:

$$\begin{aligned}
&V_{ij_{\Delta Z_t}} \\
&= \Big(E[(x_{1t} - \Delta x_{1t})^{(n_{1i}+n_{1j})}]E[(x_{2_1t} - \Delta x_{2_1t})^{(n_{2_1i}+n_{2_1j})}] \\
&\qquad \cdots E[(x_{2_mt} - \Delta x_{2_mt})^{(n_{2_mi}+n_{2_mj})}]\Big) \\
&\quad - \Bigg(E[(x_{1t} - \Delta x_{1t})^{n_{1i}}]E[(x_{2_1t} - \Delta x_{2_1t})^{n_{2_1i}}] \\
&\qquad \cdots E[(x_{2_mt} - \Delta x_{2_mt})^{n_{2_mi}}]E[(x_{1t} - \Delta x_{1t})^{n_{1j}}] \\
&\qquad E[(x_{2_1t} - \Delta x_{2_1t})^{n_{2_1j}}] \cdots E[(x_{2_mt} - \Delta x_{2_mt})^{n_{2_mj}}]\Bigg)
\end{aligned} \tag{4.36}$$

where, $n_{1I} + n_{2_1I} + \cdots + n_{2_mI} = n, \;\; I = 1, \; \cdots, \; M_n(K)$.

Proof. Let us first show the following for any X and Y random variable,

$$\begin{aligned}
Cov(X, Y) &= E[(X - E[X])(Y - E[Y])] \\
&= E[XY - E[X]Y - E[Y]X + E[Y]E[X]] \\
&= E[XY] - 2E[X]E[Y] + E[X]E[Y] \\
&= E[XY] - E[X]E[Y].
\end{aligned} \tag{4.37}$$

Since Z_{It} is deterministic at time t, using **Lemma 2.3** in the Section 2.3, it does not have any effect on the covariance of ΔZ_t.

Therefore,

$$\begin{aligned}
V_{ij_{\Delta Z_t}} &= Cov(\Delta Z_{it}, \Delta Z_{jt}) = Cov(\Delta Z^r_{it}, \Delta Z^r_{jt}) \\
&= E[\Delta Z^r_{it} \Delta Z^r_{jt}] - E[\Delta Z^r_{it}] E[\Delta Z^r_{jt}] \\
&= E[(x_{1t} - \Delta x_{1t})^{n_{1_i}} (x_{2_1 t} - \Delta x_{2_1 t})^{n_{2_{1_i}}} \cdots (x_{2_m t} - \Delta x_{2_m t})^{n_{2_{m_i}}} \\
&\quad (x_{1t} - \Delta x_{1t})^{n_{1_j}} (x_{2_1 t} - \Delta x_{2_1 t})^{n_{2_{1_j}}} \cdots (x_{2_m t} - \Delta x_{2_m t})^{n_{2_{m_j}}}] \\
&\quad - E[(x_{1t} - \Delta x_{1t})^{n_{1_i}} (x_{2_1 t} - \Delta x_{2_1 t})^{n_{2_{1_i}}} \cdots (x_{2_m t} - \Delta x_{2_m t})^{n_{2_{m_i}}}] \\
&\quad \times E[(x_{1t} - \Delta x_{1t})^{n_{1_j}} (x_{2_1 t} - \Delta x_{2_1 t})^{n_{2_{1_j}}} \cdots (x_{2_m t} - \Delta x_{2_m t})^{n_{2_{m_j}}}]
\end{aligned} \tag{4.38}$$

With the assumption (*iii*) in the Section 4.1 and using (4.35) the rest of the proof is trivial. □

Remark 4.3: The calculation of $E[(x_{it} - \Delta x_{it})^{n_{iI}}]$ is very straight forward with the formula provided in **Lemma 2.2** in the Section 2.3. For a model with n modes, general formulas for $E[(Y - \Delta Y)^i],\ i = 1 \cdots 2n$, where ΔY is a Gaussian random variable, need to be calculated using (2.20). Thereafter, the computation of the the covariance and expected value of ΔZ_t is only multiplication of polynomials with substitution of the values inside the calculated general formulas. MATLAB and most other programming languages can handle this multiplication very efficiently. By continuing our illustrative example 4.2 we show this simplicity.

Illustrative example 4.3: Since the model (4.14) has two modes, $E[(Y - \Delta Y)^i],\ i = 1 \cdots 2 \times 2 = 4$ has to be derived. These can be calculated using (2.20):

$$\begin{aligned}
E[(Y - \Delta Y)] &= Y \\
E[(Y - \Delta Y)^2] &= Y^2 + \sigma^2_{\Delta Y} \\
E[(Y - \Delta Y)^3] &= Y^3 + 3Y\sigma^2_{\Delta Y} \\
E[(Y - \Delta Y)^4] &= Y^4 + 6Y^2\sigma^2_{\Delta Y} + 3\sigma^4_{\Delta Y}
\end{aligned} \tag{4.39}$$

The ΔZ_t vector for the two-mode model (4.14) and its corresponding expected value, using **Theorem 4.3** and (4.39), will be:

$$\begin{aligned}
\Delta Z_t &= \begin{bmatrix}
x_{1t}^2 - (x_{1t} - \Delta x_{1t})^2 \\
x_{1t}x_{2_1t} - (x_{1t} - \Delta x_{1t})(x_{2_1t} - \Delta x_{2_1t}) \\
x_{1t}x_{2_2t} - (x_{1t} - \Delta x_{1t})(x_{2_2t} - \Delta x_{2_2t}) \\
x_{2_1t}^2 - (x_{2_1t} - \Delta x_{2_1t})^2 \\
x_{2_1t}x_{2_2t} - (x_{2_1t} - \Delta x_{2_1t})(x_{2_2t} - \Delta x_{2_2t}) \\
x_{2_2t}^2 - (x_{2_2t} - \Delta x_{2_2t})^2
\end{bmatrix} \\
\Rightarrow E[\Delta Z_t] &= \begin{bmatrix}
x_{1t}^2 - (x_{1t}^2 - \sigma_{\Delta x_1}^2) \\
x_{1t}x_{2_1t} - (x_{1t}x_{2_1t}) \\
x_{1t}x_{2_2t} - (x_{1t}x_{2_2t}) \\
x_{2_1t}^2 - (x_{2_1t}^2 - \sigma_{\Delta x_{2_1}}^2) \\
x_{2_1t}x_{2_2t} - (x_{2_1t}x_{2_2t}) \\
x_{2_2t}^2 - (x_{2_2t}^2 - \sigma_{\Delta x_{2_2}}^2)
\end{bmatrix} = \begin{bmatrix}
\sigma_{\Delta x_1}^2 \\ 0 \\ 0 \\ \sigma_{\Delta x_{2_1}}^2 \\ 0 \\ \sigma_{\Delta x_{2_2}}^2
\end{bmatrix}
\end{aligned} \tag{4.40}$$

Also calculation of some of the elements in the covariance matrix by using **Theorem 4.4** and (4.39) are as follow:

$$\begin{aligned}
V_{11\Delta Z_t} &= E[(x_{1t} - \Delta x_{1t})^4] - E[(x_{1t} - \Delta x_{1t})^2]E[(x_{1t} - \Delta x_{1t})^2] \\
&= x_{1t}^4 + 6x_{1t}^2\sigma_{\Delta x_1}^2 + 3\sigma_{\Delta x_1}^4 - (x_{1t}^2 + \sigma_{\Delta x_1}^2)(x_{1t}^2 + \sigma_{\Delta x_{1t}}^4) \\
&= 4x_{1t}^2\sigma_{\Delta x_{1t}}^2 + 2\sigma_{\Delta x_{1t}}^4 \\
V_{12\Delta Z_t} &= E[(x_{1t} - \Delta x_{1t})^3]E[(x_{2_1t} - \Delta x_{2_1t})] \\
&\qquad - E[(x_{1t} - \Delta x_{1t})^2]E[(x_{1t} - \Delta x_{1t})]E[(x_{2_1t} - \Delta x_{2_1t})] \\
&= (x_{1t}^3 + 3x_{1t}\sigma_{\Delta x_1}^2)(x_{2_1t}) - (x_{1t}^2 + \sigma_{\Delta x_1}^2)(x_{1t})(x_{2_1t}) \\
&= 2x_{1t}x_{2_1t}\sigma_{\Delta x_1}^2 \\
V_{22\Delta Z_t} &= E[(x_{1t} - \Delta x_{1t})^2E[(x_{2_1t} - \Delta x_{2_1t})^2] - E[(x_{1t} - \Delta x_{1t})]^2E[(x_{2_1t} - \Delta x_{2_1t})]^2 \\
&= (x_{1t}^2 + \sigma_{\Delta x_1}^2)(x_{2_1t}^2 + \sigma_{\Delta x_{2_1}}^2) - (x_{1t}^2)(x_{2_1t}^2) \\
&= x_{1t}^2\sigma_{\Delta x_{2_1}}^2 + x_{2_1t}^2\sigma_{\Delta x_1}^2 + \sigma_{\Delta x_1}^2\sigma_{\Delta x_{2_1}}^2 \\
V_{14\Delta Z_t} &= E[(x_{1t} - \Delta x_{1t})^2]E[(x_{2_1t} - \Delta x_{2_1t})^2] - E[(x_{1t} - \Delta x_{1t})^2]E[(x_{2_1t} - \Delta x_{2_1t})^2] \\
&= 0
\end{aligned} \tag{4.41}$$

As mentioned, the procedure of calculating the covariance matrix $V_{\Delta Z_t}$ can be implemented in MATLAB or any other programming languages efficiently.

4.4 Simulation results

The simulation examples in this section are chosen to illustrate the scalability and consistency (A three-mode model numerical example), practicality (A two-mode blending process) and applicability (A component placement experimental setup data) of the proposed method. The simulation results are also compared to simulation results using Algebraic Geometric approach [32] that intends to solve the same problem.

4.4.1 A three-mode model numerical example

Consider the following three-mode ARX model:

$$\text{Simulation Example 1:} \begin{cases} y_t^0 = -0.9y_{t-1}^0 + u_{t-1}^0 + 0.3u_{t-2}^0 \\ y_t^0 = 0.7y_{t-1}^0 - u_{t-1}^0 + 0.6u_{t-2}^0 \\ y_t^0 = 0.5y_{t-1}^0 - 0.6u_{t-1}^0 - 0.5u_{t-2}^0 \end{cases} \tag{4.42}$$

In this example the output is considered to have additive white gaussian noise with normal distribution with zero mean and σ^2 variance and the inputs are noise free. Since each sub-model is an ARX model, not only the available data y_t at time t is noisy, but also y_{t-1} is having additive measurement noise. Therefore the EIV model of the above example will be:

$$\begin{aligned} X_t &= [X_{1t} \;\; X_{2t}] = [X_{1t}^0 \;\; X_{2t}^0] + [\Delta X_{1t} \;\; \Delta X_{2t}] \\ &= [y_t \;\; y_{t-1} \;\; u_{t-1} \;\; u_{t-2}] = [y_t^0 \;\; y_{t-1}^0 \;\; u_{t-1}^0 \;\; u_{t-2}^0] + [\nu_t \;\; \nu_{t-1} \;\; 0 \;\; 0] \end{aligned} \tag{4.43}$$

The ν_t is considered to be a white gaussian noise, and ν_t and ν_{t-1} are independent from each other. Therefore the ΔX in this example complies with the assumption *iii* in the Section 4.1. The input has a uniform distribution of $[-1 \;\; 1]$ and the switching sequence is a discrete random number between 1, 2 and 3. The simulation is performed under different noise conditions. In this simulation, 100 independent executions are performed for $\sigma = [0.01 \;\; 0.1 \;\; 0.2 \;\; 0.3 \;\; 0.4]$ and the number of data points for each run is $N = 200$. The estimation error for parameters vector H is computed as the following normalized error:

$$\text{error H} = \frac{||H - H^0||_2}{||H^0||_2}. \tag{4.44}$$

The mean values of estimation's errors for each noise standard deviation are illustrated in table 4.1, using the proposed Stochastic Algebraic Geometric approach and the Algebraic Geometric approach [32].

Table 4.1: The mean estimation error in 100 independent execution for parameter vector (H) for example 1

σ	0.01	0.1	0.2	0.3	0.4
The SAG approach	0.0050	0.0773	0.1832	0.3360	0.5534
The AG approach	0.0176	0.2593	0.5188	0.7248	0.9095

Table 4.1 clearly shows the improvement of estimation for the H parameter vector. The estimation error is decreased by 50% for all different noise variances scenarios. This improvement is achieved without increasing any complexity with respect to the length of the HDP parameters. In implementation of both the AG and the SAG approaches, the "fminunc" function is used in the MATLAB. Also note that the length of the H vector is $M_n(K) = M_3(4) = 20$. Hence, this example clearly shows that the proposed method can provide satisfactory estimation even in complicated scenarios.

The consistency of the proposed method is also evaluated with this simulation example. For this purpose, 10 independent runs are executed for each data point numbers:

$$N = [100 \;\; 1000 \;\; 2000 \;\; 3000 \;\; 4000 \;\; 5000].$$

The standard deviation of the measurement noise for all these executions is set to $\sigma = 0.3$. Figure 4.1 illustrates the simulation results for the mean of parameter vector H estimation error with different number of data points.

4.4.2 The blender process

Consider a blending process as figure 4.2.

In the steady state, it is assumed that the volume of liquid in the tank is constant since all the flow rates $q_1,\ q_2,\ q_3,\ q_4, q_5$ are constant, subject to a small variation. This is a revised example from [62]. The density of all streams is constant at 90 lb/ft^3. The recycling line is 68.8ft long and has an inside diameter of 4in. The tank is 6ft in diameter and is perfectly mixed. The normal steady state values are:

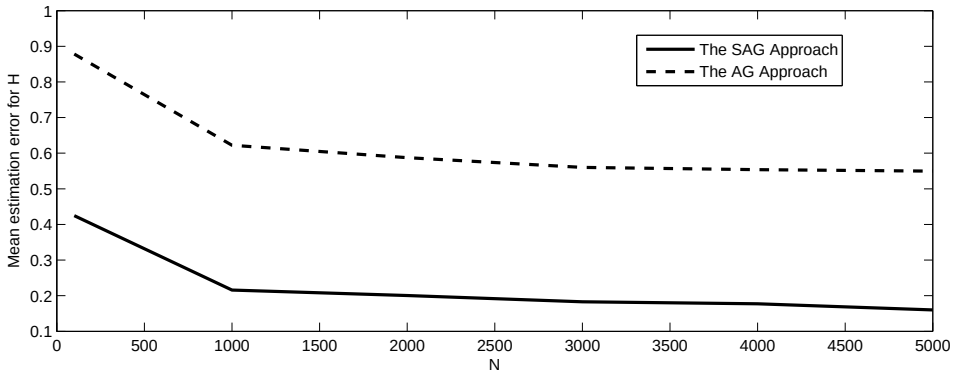

Figure 4.1: The consistency of the estimation for H in 3-mode model

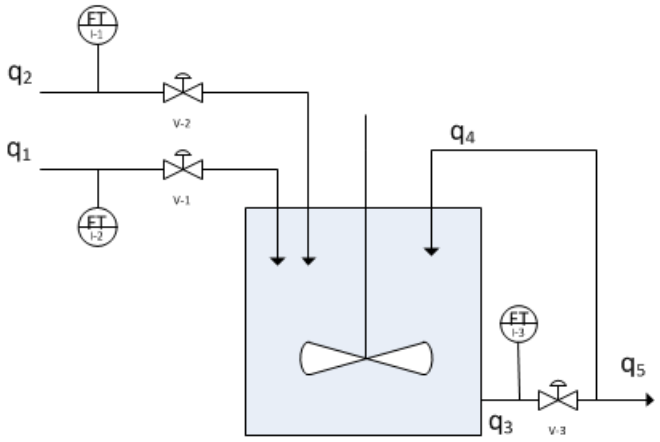

Figure 4.2: The blender process schematic

$$\bar{q_1} = 50ft^3/min \quad \bar{q_2} = 2ft^3/min \quad \bar{q_3} = 82ft^3/min \quad V = 100ft^3$$

The mass balance equation for this blender is:

$$\frac{1}{(c-1)}q_1 + \frac{1}{(c-1)}q_2 = q_3, \tag{4.45}$$

where c is the recycling rate. In this simulation example, the recycling rate is considered to switch between its nominal value $C = 0.37$ and $C = 0.407$ in a periodic manner. The flow rates $q_1,\ q_2,\ q_3$ are measured with independent additive gaussian noise with zero mean and variances σ_i^2. Therefore, in this example both input and outputs are noisy. For this example: $X_1 = q_3$, $X_2 =$

$[q_1 \;\; q_2]$ and $\Delta X = [\nu_1 \;\; \nu_2 \;\; \nu_3]$.

The switching happens every 30 samples and number of the data points in each simulation run is $N = 200$. The inputs q_1 and q_2 are considered to have uniform distribution $[-1 \;\; 1]$ and the simulation is executed under two different noise scenarios:

Noise scenario 1: 100 independent executions for each standard deviation: $\sigma_1 = \sigma_2 = \sigma_3 = [0.1 \;\; 0.2 \;\; 0.3 \;\; 0.4]$

Noise scenario 2: 100 independent executions for each standard deviation: $\sigma_1 = \sigma_2 = [0.1 \;\; 0.2 \;\; 0.3 \;\; 0.4]$ and $\sigma_3 = 2\sigma_2$.

The mean estimation error for HDP parameters vector (H) and also mean estimation error for the sub-model parameter vectors are measured. The later error is calculated by using (4.44) and the former error is calculated by using the following:

$$\text{error b} = \max_{i=1,\cdots,n} \min_{j=1,\cdots,n} \frac{||b_i - b_j^0||}{||b_j^0||} \tag{4.46}$$

Figures 4.3 and 4.4 show the performance results for the SAG and the AG approaches under two noise scenarios. It is illustrated that the proposed method outperforms the AG approach and delivers estimations with significantly lower (around 50%) estimation errors.

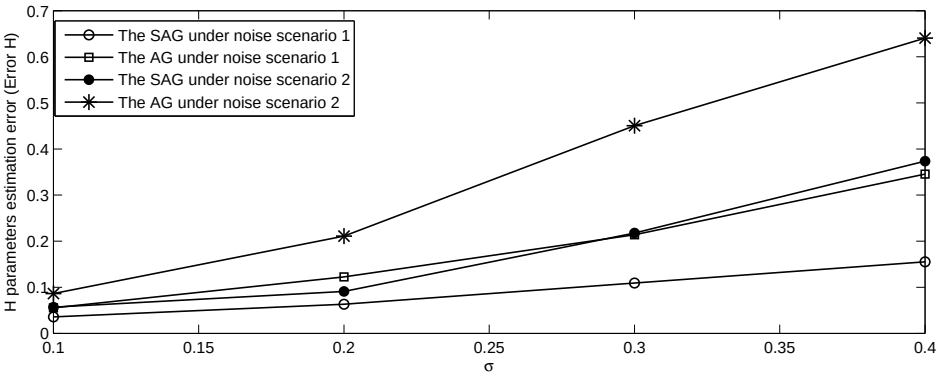

Figure 4.3: The mean estimation error of parameter vector H (error H)for the blender under both noise scenarios

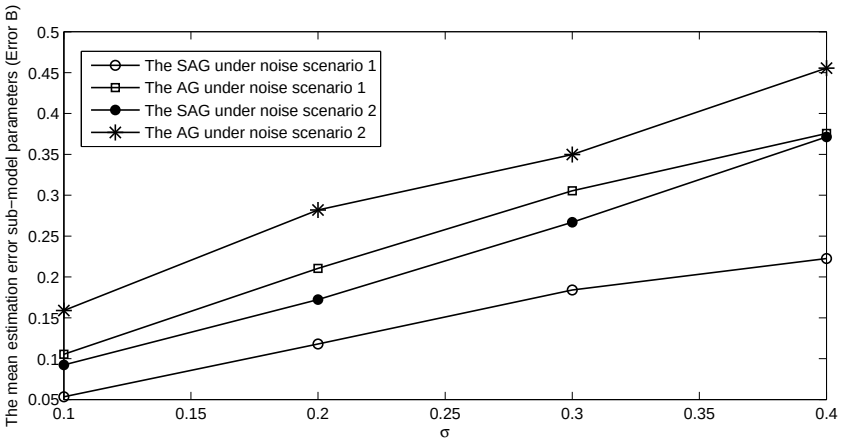

Figure 4.4: The mean estimation error of parameter vector b (error b)for the blender under both noise scenarios

4.4.3 The component placement experimental setup

In this section the proposed SAG method is applied on four experimental setup data sets. This setup is a component placement process in pick-and-place machines [63] as shown in figure 4.5.

Figure 4.5: The component placement setup [63].

The data sets are down sampled to $50Hz$ so that the results can be comparable to the results reported in [32]. The down-sampled data has 750 samples, which is divided into two set of training and testing data sets with portion of

Table 4.2: The comparison of SSR and SSE for SAG approach and AG approach in the component placement process

		Dataset 1	Dataset 2	Dataset 3	Dataset 4
SSR	The SAG approach	0.0678	0.4220	0.6132	1.1070
	The AG approach	0.0803	0.4765	0.6692	3.1004
SSE	The SAG approach	0.0816	0.3342	0.6621	1.1768
	The AG approach	0.1195	0.4678	0.7368	3.8430

$2:1$. The model is considered to have two switching mode with the regressor vectors: $X_1 = [y_t]$, $X_t = [y_{t-1} \;\; y_{t-2} \;\; u_{t-1} \;\; u_{t-2}]$. The u signal is the input to the motor that is deriving the movement in the vertical axes, and the y output is the hight. Table 4.2 reports the average square sum of one step ahead prediction error (SSR) and average sum of squared one step ahead simulation errors resulted by implementing the proposed method (SAG) on four sets of the data. The variance of the noise is assumed to be $\sigma = 0.1$. The results reported in [32] are also reported in table 4.2 for comparison.

The results in table 4.2 illustrates the improvements achieved by using the proposed method in performing the parameter estimation. Also it shows that the SAG approach is practical and can be implemented in real applications. In order to better show the promising estimation results, the cross and self validation plots in dataset 4 , which has the most SSR and SSE error, are presented in figures 4.6 and 4.7.

Remark 4.4: As discussed in the Section 2.1.2, calculation of the covariance matrix to a scalar value suffice. Therefore, although the actual value of the standard deviation of the measurement noise in these experimental data is not available but having a logical assumption delivers acceptable results. The proposed method is not highly sensitive to the true value of the standard deviation of the noise. In order to show this, the proposed method is used for estimation of the parameters in the dataset 4 using different values for the standard deviation of the measurement noise. The computed SSR and SSE for different noise variance assumption are reported in table 4.3.

As it can be seen in the table 4.3, the SSR and SSE does not change noticeably under different noise assumptions.

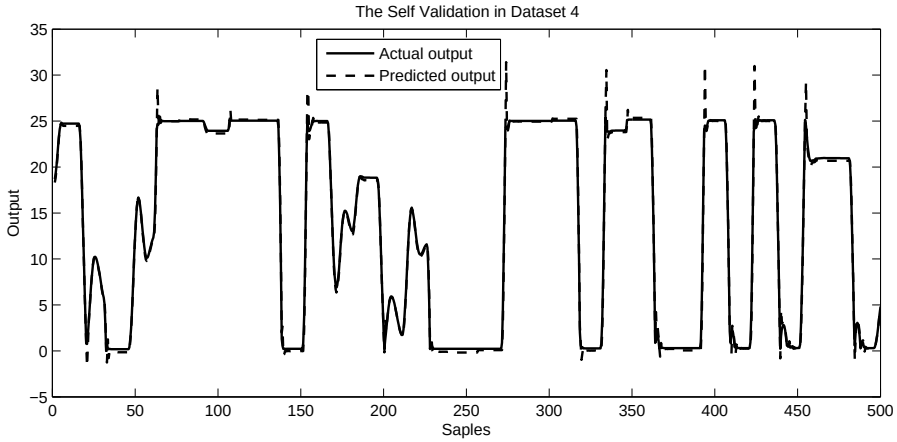

Figure 4.6: The self validation of the estimation in dataset 4 using the proposed SAG method.

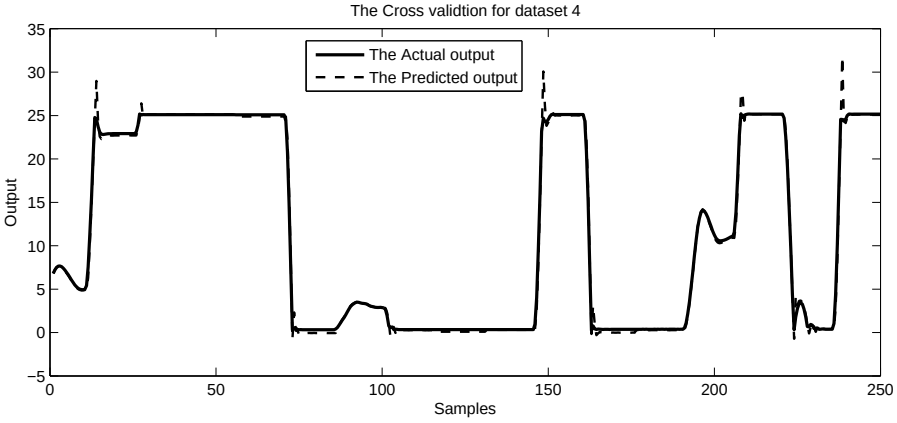

Figure 4.7: The cross validation of the estimation in dataset 4 using the proposed SAG method.

Table 4.3: The SSR and SSE for dataset 4 using the proposed method with different assumptions for standard deviation of the noise

σ	0.01	0.05	0.1	0.2
The SSR	1.1191	1.1010	1.1017	1.1996
The SSE	1.1655	1.1673	1.1768	1.2538

Chapter 5

The Recursive Element-Wise Total Least Square Method

[1] Most of the system identification methods developed based on the EIV model need to utilize all the data and are not normally suitable for on-line applications. Hence, the online recursive version for these methods are of great interests and some have been developed in the literature. In [64] and [65] a recursive solution for the EIV model based system identification was presented. However, the proposed method has some restrictions on the correlation of inputs and the output noises. In this method, inputs and the output noises are assumed to be uncorrelated, and the noise characteristics cannot change with the time, meaning that the noise covariance matrix is constant for all the data points.

This chapter introduces a new recursive solution to the EIV system identification problem that relieves most of the restrictions on the noise characteristics. The input and output noises can be correlated among themselves and to each other. The noise variance for each input or output signal can be time varying. This allows us to apply this method of identification to systems or processes where the measurement noise profile changes with time. As an example, in a petrochemical plant, starting different motors can induce different noise characteristics. Another example is that a car driving under different road conditions may experience different noise characteristics. Finally the ability to deal with some noise-free inputs or outputs among the noisy ones is one other advantage of our purposed method. In this case, the noise covariance matrix has some zero rows or columns (i.e. ill conditioned) and the parameters

[1] A version of this chapter has been submitted to control system letters journal.

can still be estimated.

5.1 Recursive prediction error method (PEM)

The proposed recursive version of the EW-TLS method for (2.10) is developed in this chapter, by following the recursive PEM algorithm discussed in [66,67]. Consider all the data points available till the sample t, using the cost function (2.11):

$$f_t(B) = \sum_{i=1}^{t} r_i^T(B) Q_i^{-1}(B) r_i(B). \tag{5.1}$$

where, $r_i^T(B)$ and $Q_i^{-1}(B)$ are defined as in (2.11). The symbols and notations in this chapter are described in the Section 2.1. Let the $B(t-1)$ be the parameter estimate at the time $t-1$. The goal is to find the $B(t)$ that minimizes the $f_t(B)$. Using the Tailor expansion of the cost function around $B(t-1)$ we have:

$$\begin{aligned} f_t(B) = f_t(B(t-1)) &+ f_t'(B(t-1))[B - B(t-1)] \\ &+ 0.5[B - B(t-1)]^T f_t''(B(t-1))[B - B(t-1)] \\ &+ o(|B(t) - B(t-1)|^2). \end{aligned} \tag{5.2}$$

Where the differentiation is made with respect to B. Minimization of the above equation with respect to B results in

$$B(t) = B(t-1) - [f_t''(B(t-1))]^{-1} f_t'[B(t-1))]^T + o(|B(t) - B(t-1)|). \tag{5.3}$$

Since $f_t(B)$ is a summation of similar functions up to sample t, it can be expanded as follow:

$$\begin{aligned} f_t(B) =& \sum_{i=1}^{t-1} \left(r_i^T(B) Q_i^{-1} r_i(B) \right) + r_t^T(B) Q_t^{-1} r_t(B) \\ =& f_{t-1}(B) + r_t^T(B) Q_t^{-1} r_t(B) \end{aligned} \tag{5.4}$$

Therefore taking the derivative with respect to B leads to:

$$\begin{aligned} f'_t(B) &= f'_{t-1}(B) + \left(r_t^T(B) Q_t^{-1} r_t(B)\right)' \\ &= f'_{t-1}(B) + g_t(B) \end{aligned} \tag{5.5}$$

By taking the second derivative, the following can be obtained,

$$f''_t(B) = f''_{t-1}(B) + g'_t(B). \tag{5.6}$$

In order to find the recursive regression, these assumptions need to be made:

- The difference of the next estimate $B(t)$ and the previous one $(B(t-1))$ is very small so that $o(|B(t) - B(t-1)|)$ is negligible and $f''_t(B(t)) = f''_t(B(t-1)$.
- $B(t-1)$ is indeed the optimal solution for f_{t-1}, therefore $f'_{t-1}(B(t-1)) = 0$.

With the above assumptions the equation (5.5) becomes

$$\begin{aligned} f'_t(B(t-1)) =& f'_{t-1}(B(t-1)) + g_t(B(t-1)) \\ =& g_t(B(t-1)) \end{aligned} \tag{5.7}$$

and equation (5.6) will change in

$$f''_t(B(t)) = f''_{t-1}(B(t-1)) + g'_t(B(t-1)). \tag{5.8}$$

Now, by substituting (5.5) and (5.6) into (5.3) we have:

$$B(t) = B(t-1) - [f''_t(B(t))]^{-1} g_t^T(B(t-1)). \tag{5.9}$$

It is clear that in the recursive PEM method, the first and second derivatives of the (5.1) are necessary. The cost function in (5.1) is not linear with respect to B, therefore calculating these derivatives are not straightforward. the Section 5.2 is dedicated to calculation of these derivatives.

5.2 The first and the second derivatives of the cost function

Without loss of generality and in order to save space only the univariate case is discussed here. The multivariate case can be readily obtained by following the same procedure. Only the mathematical forms are more complex. In this case, $Q_i(B)$ and $r_i(B)$ in (5.1) are scaler functions of vector $B \in \mathbb{R}^{m\times 1}$, where m is the number of parameters to be estimated. The first derivative of the cost function with respect to B is:

$$r\frac{\partial f_t(B)}{\partial B} = \sum_{i=1}^{t} \frac{\partial f_i(B)}{\partial B} \tag{5.10}$$

For simplicity $\frac{\partial f_i(B)}{\partial B}$ is calculated in the following.

$$\begin{aligned}\frac{\partial f_i(B)}{\partial B} &= 2X_{1i}r_i^T(B)Q_i^{-1}(B) - \\ & r_i^T(B)Q_i^{-1}(B)\left(2(V_{i1}B - V_{i12}) + (I_m \otimes [B^T \quad -1])\frac{dV_i}{dB}\begin{bmatrix} B \\ -1 \end{bmatrix}\right) Q_i^{-1}(B)r_i(B)\end{aligned} \tag{5.11}$$

For the simplicity consider $G = 2(V_{i1}B - V_{i12})$ and $V'_B = (I_m \otimes [B^T \quad -1])\frac{dV_i}{dB}\begin{bmatrix} B \\ -1 \end{bmatrix}$. Therefore,

$$\begin{aligned}\frac{\partial f_i(B)}{\partial B} &= 2X_{1i}r_i^T(B)Q_i^{-1}(B) - r_i^T(B)Q_i^{-1}(B)(G + V'_B)Q_i^{-1}(B)r_i(B) \\ &= \mathbf{f'_1} - \mathbf{f'_2}\end{aligned} \tag{5.12}$$

Also the following derivatives are important and will be used.

$$\frac{\partial r_i(B)}{\partial B^T} = \frac{\partial r_i^T(B)}{\partial B^T} = X_{1i}^T \tag{5.13}$$

and,

$$\begin{aligned}\frac{\partial B^T}{\partial B^T} &= \overline{U}_{1m} = \text{row}(I_m) \\ \frac{\partial B}{\partial B^T} &= I_m\end{aligned} \tag{5.14}$$

where *row()* is a function that puts all the rows of a matrix in a single row vector. Using above and (2.15a) we have,

$$\begin{aligned}\frac{\partial Q_i}{\partial B^T} =&\frac{[B^T-1]}{\partial B^T}(I_m \otimes V_i \begin{bmatrix} B \\ -1 \end{bmatrix}) + [B^T-1]\frac{\partial V_i \begin{bmatrix} B \\ -1 \end{bmatrix}}{\partial B^T} \\ =&[\overline{U}_{1m} \;\; \mathbf{0}_{1\times m}](I_m \otimes V_i \begin{bmatrix} B \\ -1 \end{bmatrix}) \\ &+ [B^T-1]\left(\frac{\partial V_i}{\partial B^T}(I_m \otimes \begin{bmatrix} B \\ -1 \end{bmatrix}) + V_i \begin{bmatrix} I_m \\ \mathbf{0} \end{bmatrix}\right) \\ =&2(B^T V_{1i} - V_{i12}^T) + [B^T-1]\frac{\partial V_i}{\partial B^T}(I_n \otimes \begin{bmatrix} B \\ -1 \end{bmatrix}) \\ =&G^T + {V'_B}^T\end{aligned} \tag{5.15}$$

Using (2.15d) and the above equation results in:

$$\begin{aligned}\frac{\partial Q_i^{-1}}{\partial B^T} &= -Q_i^{-1}\frac{\partial Q_i}{\partial B^T}(I_m \otimes Q_i^{-1}) \\ &= -Q_i^{-1}(G^T + {V'_B}^T)Q_i^{-1}\end{aligned} \tag{5.16}$$

Also consider

$$\begin{aligned}\frac{\partial G}{\partial B^T} &= 2\frac{\partial V_{i1}B - V_{i12}}{\partial B^T} \\ &= 2\left(\frac{\partial V_{i1}}{\partial B^T}(I_m \otimes B) + V_{i1}\frac{\partial B}{\partial B^T} - \frac{\partial V_{i12}}{\partial B^T}\right) \\ &= 2\left(\frac{\partial V_{i1}}{\partial B^T}(I_m \otimes B) + V_{i1} - \frac{\partial V_{i12}}{\partial B^T}\right),\end{aligned} \tag{5.17}$$

and by using (2.15c), we have:

$$\begin{aligned}
\frac{\partial V'_B}{\partial B^T} &= \frac{\partial (I_m \otimes [B^T \quad -1]) \frac{dV_i}{dB} \begin{bmatrix} B \\ -1 \end{bmatrix}}{\partial B^T} \\
&= \frac{\partial (I_m \otimes [B^T \quad -1])}{\partial B^T} (I_m \otimes \frac{dV_i}{dB} \begin{bmatrix} B \\ -1 \end{bmatrix}) + (I_m \otimes [B^T \quad -1]) \frac{\partial (\frac{dV_i}{dB} \begin{bmatrix} B \\ -1 \end{bmatrix})}{\partial B^T} \\
&= U_{m1}([\overline{U}_{1m} \;\; \mathbf{0}_{1\times m}] \otimes I_m)(I_m \otimes U_{(m+1)m})(I_m \otimes \frac{dV_i}{dB} \begin{bmatrix} B \\ -1 \end{bmatrix}) \\
&\qquad + (I_m \otimes [B^T \quad -1]) \left(\frac{\partial V_i}{\partial B \partial B^T} (I_m \otimes \begin{bmatrix} B \\ -1 \end{bmatrix}) + \frac{\partial V_i}{\partial B} \begin{bmatrix} I_m \\ \mathbf{0} \end{bmatrix} \right) \\
&= ([\overline{U}_{1m} \;\; \mathbf{0}_{1\times m}] \otimes I_m)(I_m \otimes U_{(m+1)m})(I_m \otimes \frac{dV_i}{dB} \begin{bmatrix} B \\ -1 \end{bmatrix}) \\
&\qquad + (I_m \otimes [B^T \quad -1]) \left(\frac{\partial V_i}{\partial B \partial B^T} (I_m \otimes \begin{bmatrix} B \\ -1 \end{bmatrix}) + \frac{\partial V_i}{\partial B} \begin{bmatrix} I_m \\ \mathbf{0} \end{bmatrix} \right)
\end{aligned} \tag{5.18}$$

since $U_{m1} = I_m$. Now that all the necessary component for deriving the second derivative of f is developed, the following can be considered: $\frac{\partial f}{\partial B \partial B^T} = \frac{\partial \mathbf{f}'_\mathbf{1}}{\partial B^T} - \frac{\partial \mathbf{f}'_\mathbf{2}}{\partial B^T}$.

$$\begin{aligned}
\frac{\partial \mathbf{f}'_\mathbf{1}}{\partial B^T} &= \frac{\partial X_{1i} Q_i^{-1} r_i}{\partial B^T} = X_{1i} \frac{\partial Q_i^{-1}}{\partial B^T} (I_m \otimes r_i) + X_{1i} Q_i^{-1} \frac{\partial r_i}{\partial B^T} \\
&= -X_{1i} Q_i^{-1} (G^T + {V'_B}^T) Q_i^{-1} r_i + X_{1i} Q_i^{-1} X_{1i}^T
\end{aligned} \tag{5.19}$$

and by using (2.15b)

$$
\begin{aligned}
\frac{\partial \mathbf{f_2'}}{\partial B^T} &= \frac{\partial r_i^T Q_i^{-1}(G+V_B')Q_i^{-1}r_i}{\partial B^T} \\
&= \underbrace{X_{1i}^T \otimes (Q_i^{-1}(G+V_B')Q_i^{-1}r_i)}_{f_{2_1}''} + r_i^T \frac{\partial Q_i^{-1}(G+V_B')Q_i^{-1}r_i}{\partial B^T} \\
&= f_{2_1}'' + r_i^T \frac{\partial Q_i^{-1}}{\partial B^T} \otimes ((G+V_B')Q_i^{-1}r_i) + r_i^T Q_i^{-1} \frac{\partial ((G+V_B')Q_i^{-1}r_i)}{\partial B^T} \\
&= f_{2_1}'' \underbrace{-r_i^T Q_i^{-1}(G^T + {V_B'}^T)Q_i^{-1} \otimes ((G+V_B')Q_i^{-1}r_i)}_{f_{2_2}''} \\
&\quad + r_i^T Q_i^{-1} \frac{\partial ((G+V_B')Q_i^{-1}r_i)}{\partial B^T} \\
&= f_{2_1}'' + f_{2_2}'' + \underbrace{r_i^T Q_i^{-1}(\frac{\partial G}{\partial B^T} + \frac{\partial V_B'}{\partial B^T})(I_m \otimes (Q_i^{-1}r_i))}_{f_{2_3}''} \\
&\quad + r_i^T Q_i^{-1}(G+V_B')\frac{\partial Q_i^{-1}r_i}{\partial B^T} \\
&= f_{2_1}'' + f_{2_2}'' + f_{2_3}'' + r_i^T Q_i^{-1}(G+V_B')\left(\frac{\partial Q_i^{-1}}{\partial B^T}r_i + Q_i^{-1}\frac{\partial r_i}{\partial B^T}\right) \\
&= f_{2_1}'' + f_{2_2}'' + f_{2_3}'' \underbrace{-r_i^T Q_i^{-1}(G+V_B')Q_i^{-1}(G^T + {V_B'}^T)Q_i^{-1}r_i}_{f_{2_4}''} \\
&\quad + \underbrace{r_i^T Q_i^{-1}(G+V_B')Q_i^{-1}X_{1i}^T}_{f_{2_5}''}
\end{aligned}
\tag{5.20}
$$

At the end we have:

$$
\begin{aligned}
\frac{\partial f_i}{\partial B \partial B^T} = 2\left(X_{1i}Q_i^{-1}X_{1i}^T - X_{1i}Q_i^{-1}(G^T + {V_B'}^T)Q_i^{-1}r_i\right) \\
- (f_{2_1}'' + f_{2_2}'' + f_{2_3}'' + f_{2_4}'' + f_{2_5}'')
\end{aligned}
\tag{5.21}
$$

Remark 5.1: In the batch version of EW-TLS it is assumed that the covariance matrices V_is are independent from the parameter vectors B therefore $\frac{dV_i}{dB} = 0$. However, in this thesis $\frac{dV_i}{dB}$ is not considered zero but can possibly have the second non-zero derivative. This shows that the covariance matrices can be functions of parameter matrix as well. Such a case is possible when the measurement noise is dependent on the characteristics of the system and the system itself is time-varying.

Remark 5.2: Since all of the terms in (5.21) are matrices and the deriva-

tives can be found by only multiplication and summation of matrices, the computation complexity and time for this calculation is low.

5.3 The Levenberg - Marquardt (LM) algorithm

The chance of falling into a wrong local minimum is always a critical issue in on-line recursive approaches. The batch version of EW-TLS method has been shown to have the local convergence property [49], while the convergency of the recursive version proposed in this work can be improved by Levenberg - Marquardt [68], [69] algorithm. This algorithm implies that in (5.9) instead of $[f_t''(B(t))]^{-1}$ one should use $[f_t''(B(t)) + \lambda \text{ diag}[f_t''(B(t))] \,]^{-1}$.

The constant λ is called the damping parameter. The damping parameter is selected by the following procedure: Select and initial value λ. After a recursion, evaluate the prediction error by using the estimated parameters; if the error is higher than the previous error then increase the damping parameter by a factor (ν). If the error is lower, then decrease the damping parameter by the same factor and continue to the next recursion.

5.4 Recursive EW-TLS algorithm

At the end, the following algorithm can be presented as the recursive element-wise total least square solution to the general error-in-variable problem.

Algorithm 2: *Recursive EW-TLS Algorithm*

1. Choose a suitable initialization values for $B(0)$, $f_0''(0)$, λ and $\nu > 1$.
2. Calculate $f_t'(B(t-1))$ from equation (5.12).
3. Calculate $f_t''(B(t))$ from (5.21).
4. Update

$$B(t) = B(t-1) - [f_t''(B(t)) + \lambda \text{ diag}\{f_t''(B(t))\} \,]^{-1} g_t^T(B(t-1)).$$

5. If $f_t(B(t)) < f_t(B(t-1))$ then $(\lambda = \lambda / \nu)$ else $(\lambda = \lambda \times \nu)$

6. repeat for the next point

Remark 5.3: Unlike the batch version of the EW-TLS algorithm, the recursive version proposed in this chapter does not need to solve any nonlinear equations. Therefore, it does not need to solve any nonconvex problems using the linearization assumption. The main approximation only appears in the Tailor expansion, in which the third and higher order terms are omitted.

Remark 5.4: The initialization step is always important for the recursive approaches. One way to choose a proper initialization values is to use a limited number of starting samples and estimate the model parameters using the TLS solution ate the beginning. The initialization of f'' is also important. The larger value of $f_0(0)''$ can increase the time of convergence but if the initial parameters are accurate enough it decreases the chance of the algorithm falling into a wrong local minimum.

Clearly, this proposed online REW-TLS method can find useful applications in monitoring and tracking the system parameter changes.

5.5 Simulation results

5.5.1 The recursive EW-TLS performance analysis

In this section the performance and accuracy of the recursive EW-TLS in different noise scenarios will be examined by using an example. In addition, we compare the estimation results of the REW-TLS to the recursive PEM method. The following system is used for the simulations:

$$y(t) = 0.3y(t-1) - 0.5u(t-1) - 0.3u(t-2) + 0.8 \tag{5.22}$$

In this system $X_1 = [y(t-1) \quad u(t-1) \quad u(t-2) \quad 1]$ and $X_2 = y(t)$, consequently the $n = 4, l = 1$ and $B^0 = [0.3 \quad -.5 \quad -0.3 \quad 0.8]$. The input (u) is a random Gaussian sequence with zero mean and variance equal to one. For each noise scenario the standard deviation (σ)of the measurement noise is changed as $\sigma = [0.1 \quad 0.2 \quad 0.3 \quad 0.4 \quad 0.5]$. Also, the number of samples is changed as $N = 500, \ 1000, \ 3000, \ 5000, \ 10000$. In each combination of the different cases 100 noise realizations are generated, therefore 100 simulations

are performed. The estimation error for each noise level and each number of data points is found as the mean of the following error in each simulation:

$$error(\sigma, m) = \frac{||B(N) - B^0||}{||B^0||}.$$

Four different noise scenarios are considered in this simulation study:

Scenario 1: The covariance matrices for each row of ΔD are random and the measurement noise for the inputs/output are correlated. In this scenario each element of V_is are uniform random numbers ($\mu = 1$ and $\sigma = .5$).

Scenario 2: The covariance matrices for each row af ΔD are random but the measurement noise for the inputs/output are uncorrelated. In this scenario V_i are diagonal matrices with uniform random numbers ($\mu = 1$ and $\sigma = .5$).

Scenario 3: The covariance matrices for each row of ΔD are an all equal to an identical matrix. In this case $V_i = diag([.1 \quad .2 \quad .15 \quad .5 \quad .3])$. This is consistent with the framework of the TLS method.

Scenario 4: There is no noise in inputs and the only noise is in the output (X_2) therefore the V_is are all the same and have all their elements equal to zero except the element in the last row and the last column. This is the LS problem.

The initial parameter set ($B(0)$) for both methods is considered as 10% higher than the actual parameters(B^0), also $P(0) = I$ in the recursive PEM method and $f_0(B(0))'' = 10 \times I$ in the recursive EW-TLS solution. The simulation results are summarized in Tables 5.1-5.4.

Several observations are made from the tables 5.1 - 5.4, which are summarized in the following remarks.

Remark 5.5: In all the scenarios the REW-TLS shows significantly improved results compared to the RPEM method, specially in more randomly changing covariance matrices. The improvement in estimation error is more significant when the noise level is higher. This suggests that, the REW-TLS method is more useful in highly noisy data. It is important to mention that in the last scenario (Table 5.4) the REW-TLS and RPEM method both have almost the same performance. This scenario is the simple Least Square estimation problem therefore the optimum solution is obtained by both methods.

Table 5.1: The comparison of estimation error for recursive EW-TLS and recursive PEM method in noise scenario 1

		$m = 500$	$m = 1000$	$m = 3000$	$m = 5000$	$m = 10000$
$\sigma = 0.1$	REW-TLS	0.0358	0.0137	0.0099	0.0083	0.0065
	RPEM	0.0286	0.0281	0.0278	0.0280	0.0285
$\sigma = 0.2$	REW-TLS	0.0785	0.0457	0.0277	0.0258	0.0316
	RPEM	0.0956	0.1004	0.0987	0.0990	0.0982
$\sigma = 0.3$	REW-TLS	0.1293	0.0925	0.0695	0.0518	0.0506
	RPEM	0.1800	0.1771	0.1792	0.1791	0.1799
$\sigma = 0.4$	REW-TLS	0.1638	0.1315	0.0929	0.0847	0.0864
	RPEM	0.2558	0.2545	0.2532	0.2530	0.2538
$\sigma = 0.5$	REW-TLS	0.2696	0.1797	0.1525	0.1467	0.1224
	RPEM	0.3159	0.3079	0.3127	0.3140	0.3135

Table 5.2: The comparison of estimation error for recursive EW-TLS and recursive PEM method in noise scenario 2

		$m = 500$	$m = 1000$	$m = 3000$	$m = 5000$	$m = 10000$
$\sigma = 0.1$	REW-TLS	0.0254	0.0187	0.0113	0.0094	0.0055
	RPEM	0.0791	0.0779	0.0785	0.0792	0.0794
$\sigma = 0.2$	REW-TLS	0.0560	0.0433	0.0244	0.0205	0.0168
	RPEM	0.2089	0.2187	0.2150	0.2160	0.2173
$\sigma = 0.3$	REW-TLS	0.0872	0.0770	0.0490	0.0373	0.0272
	RPEM	0.3216	0.3228	0.3184	0.3233	0.3191
$\sigma = 0.4$	REW-TLS	0.1749	0.1365	0.0747	0.0457	0.0427
	RPEM	0.3840	0.3902	0.3855	0.3935	0.3899
$\sigma = 0.5$	REW-TLS	0.2558	0.2152	0.0931	0.0832	0.0568
	RPEM	0.4415	0.4402	0.4392	0.4412	0.4399

Table 5.3: The comparison of estimation error for recursive EW-TLS and recursive PEM method in noise scenario 3

		$m = 500$	$m = 1000$	$m = 3000$	$m = 5000$	$m = 10000$
$\sigma = 0.1$	REW-TLS	0.0097	0.0119	0.0025	0.0022	0.0026
	RPEM	0.0083	0.0073	0.0079	0.0082	0.0082
$\sigma = 0.2$	REW-TLS	0.0171	0.0110	0.0121	0.0071	0.0074
	RPEM	0.0308	0.0301	0.0321	0.0313	0.0318
$\sigma = 0.3$	REW-TLS	0.0186	0.0143	0.0145	0.0162	0.0161
	RPEM	0.0647	0.0643	0.0662	0.0671	0.0678
$\sigma = 0.4$	REW-TLS	0.0310	0.0240	0.0284	0.0268	0.0272
	RPEM	0.1095	0.1124	0.1122	0.1122	0.1119
$\sigma = 0.5$	REW-TLS	0.0407	0.0347	0.0398	0.0394	0.0399
	RPEM	0.1569	0.1576	0.1604	0.1596	0.1612

Table 5.4: The comparison of estimation error for recursive EW-TLS and recursive PEM method in noise scenario 4

		$m = 500$	$m = 1000$	$m = 3000$	$m = 5000$	$m = 10000$
$\sigma = 0.1$	REW-TLS	0.0142	0.0105	0.0058	0.0041	0.0030
	RPEM	0.0124	0.0099	0.0058	0.0040	0.0030
$\sigma = 0.2$	REW-TLS	0.0325	0.0234	0.0123	0.0079	0.0055
	RPEM	0.0311	0.0227	0.0123	0.0078	0.0055
$\sigma = 0.3$	REW-TLS	0.0439	0.0340	0.0154	0.0146	0.0098
	RPEM	0.0436	0.0344	0.0154	0.0147	0.0098
$\sigma = 0.4$	REW-TLS	0.0623	0.0395	0.0239	0.0189	0.0149
	RPEM	0.0620	0.0388	0.0239	0.0188	0.0148
$\sigma = 0.5$	REW-TLS	0.0754	0.0518	0.0324	0.0220	0.0138
	RPEM	0.0677	0.0516	0.0305	0.0219	0.0139

Despite the significant improvement in decreasing the estimation error, the results show that REW-TLS converges slower to its local minimum. RPEM might converge to the wrong local minimum but its convergency speed is faster. However, it is also found in some simulation trials for the REW-TLS, the convergency to the right local minimum cannot be achieved. This is due to the fact that the REW-TLS dependents on good initialization in parameters and the second order derivatives (i.e. f''). In the next section , the possible improvement of the REW-TLS is discussed.

5.5.2 The improved REW-TLS using LM algorithm

In this section, throughout the simulations the improvements made by the use of LM algorithm is illustrated. To show the effectiveness of the proposed method, a more complex system is used, and for the noise profile, scenario 1 is considered.

$$\begin{aligned} y(t) = 0.3y(t-1) - 0.2y(t-2) - 0.5y(y-3) \\ - 0.5u(t-1) - 0.3u(t-2) + 0.7u(t-3) + 0.5 \end{aligned} \tag{5.23}$$

For each noise variance, 100 simulations ($N = 1000$) have been run and the convergence percentage is reported with and without the LM algorithm. The advantages of using LM algorithm are shown by comparison in two different cases below. In this simulation the initial point is taken as zero for all the cases.

Case 1: The $f_0(B(0))'' = 10 \times I$.

Case 2: The $f_0(B(0))'' = 5 \times I$.

The rates of successful convergence for each case are illustrated in 5.1.

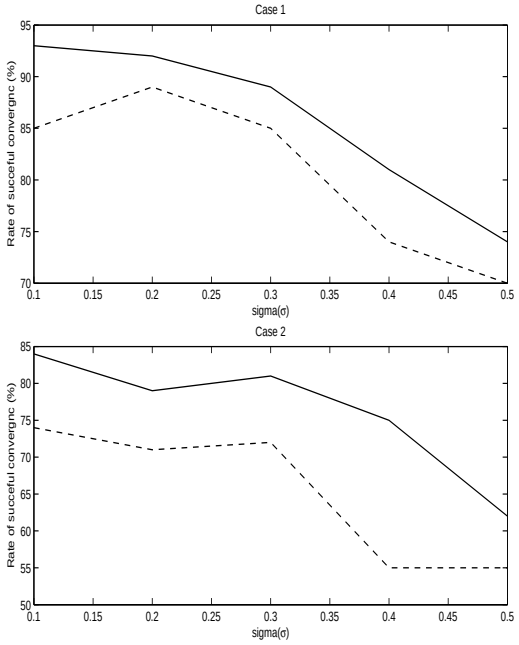

Figure 5.1: The comparison of successful convergence for REW-TLS algorithm with and without LM method. Solid line: REW-TLS with LM, dashed line: REW-TLS without LM

The following observations can be made from fig. 5.1:

Remark 5.6: Using the LM algorithm has increased the percentage of successful convergence of the REW-TLS algorithm for all the cases. This noticeable improvement is gained at a cost of a longer computation time. However, the extra computation time does not have any significant impact on the overall algorithm complexity.

Remark 5.7: The initial value of $f''(0)$ also has an important impact on the rate of successful convergence in REW-TLS algorithm. Normally higher values of $f''(0)$ results in the better convergence rate. Since in the REW-TLS algorithm the term f'' is inverted, the smaller values will result in a bigger change in the estimated parameters that might force the solution to another local minimum. On the other hand, higher $f''(0)$ may result in the slower convergence. Therefore a tradeoff exists between the faster convergency and the correct convergency.

This simulation case study shows that the adoption of the LM algorithm has increased the robustness of the REW-TLS method to both the initial values of $f''(0)$ and the parameters . This is true because the LM method algorithm has been used to prevent any sudden changes or ill conditions in many parameter estimation approaches.

5.5.3 Computation time

As stated before, the complexity of the proposed method is very low and the computation in each iteration is very straight forward and time efficient. In this section a study proves our alligation. The execution time for several different models with different number of parameters to be estimated is measured in millie seconds for each iteration. Table 5.5 shows the average execution time for 100 runs for each model under noise scenario 1. The execution platform is: MATLAB 2008a with Intel core 2Duo @2.39 GHz .

Table 5.5: The average execution time for each iteration in different models with different complexity

No. of parameters	3	4	5	6	7
Execution Time(ms)	0.2665	0.2929	0.3219	0.3328	0.3861

It can be seen that the execution time is suitable for online applications. This execution time may increase but insignificantly as the complexity of the model increases.

5.5.4 REW-TLS and STLS comparison

In this section the REW-TLS method and STLS method are compared by their simulation results. In order to compare a recursive method (REW-TLS) with a batch method (STLS), the last estimated parameters for the recursive method are used. The model used is:

$$y(t) = 1.465y(t-1) - 0.81y(t-2) - 0.5u(t-1) + 2u(t-2) \tag{5.24}$$

The input (u) is a random Gaussian sequence with zero mean and unity variance. The simulation is performed for 1500 data points under two different noise scenarios:

Scenario 1: The input and output noise are not correlated and have constant zero mean and variance equal to 1.

Scenario 2: The input and output noise are correlated, their profiles have zero mean and varying variance following a uniform random numbers with $\mu = 0$ and $\sigma = 1$.

For STLS parameter estimation a MATLAB toolbox developed in [70] is used. After identification, the step response for the estimated model using both methods under the above noise scenarios are ploted. Figure 5.2 shows these step responses.

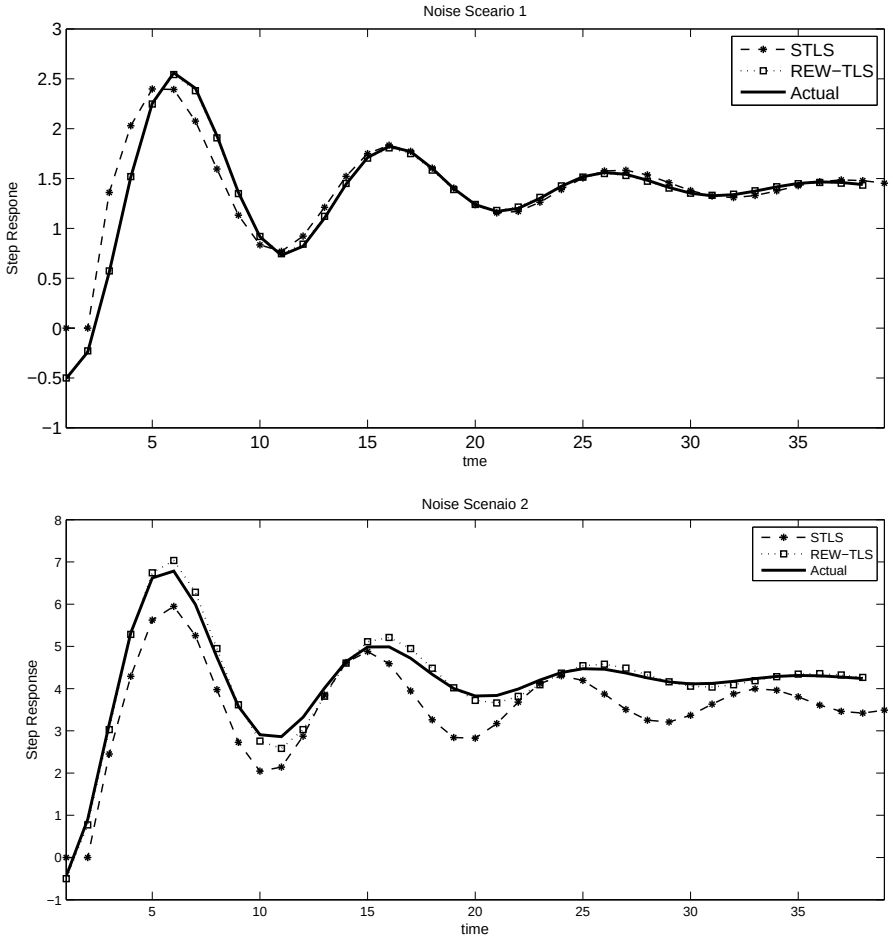

Figure 5.2: The comparison of parameter estimation of two methods: REW-TLS and STLS

It is illustrated that under the noise scenario 1 both REW-TLS and STLS provide similar and accurate parameter estimation. But, the estimation obtained by the purposed REW-TLS is more accurate than STLS in the noise

scenario 2. This shows the advantage of the developed method in parameter estimation when the noise profile is changing over time.

It is important to compare the computation time for these two methods. Using a same machine (MATLAB 2008a with Intel core 2Duo @2.39 GHz) for both algorithms, the computation time for STLS method is near 2 hours and for REW-TLS is near 1 second. This computation time makes the STLS method near impractical for large data sets and the proposed method a good candidate.

Chapter 6

The Recursive Stochastic Algebraic Geometric Approach

[1] Most of the works on the identification of SARX models provide batch algorithms, which are based on a large amount of numerical calculations making them unsuitable for on-line applications. The recursive identification algorithms for SARX models are scars and some of them are very recent. In [36], a recursive version of the AG approach was provided. The author used standard recursive identifier [66] in order to estimate the parameters of the mapped embedded model. The same approach was applied to recover the SARX parameters from the embedded model as in [32]. However, this recursive version is based on the same formulation of HDP which suffers from handling the measurement noise as its batch version. Bako et al. [71] extended the Baysian approach for SARX parameter estimation to the recursive identification method. The requirement of having prior probability densities for the parameters is the major drawback of this method. Lack of mathematical analysis is another issue in their work. Using artificial intelligence in developing recursive identification method for PWA models was discussed in [72] and [73]. In the former the fuzzy techniques were applied and in the latter, am artificial neural-network methodology was adopted.

In this section, a recursive version of the proposed SAG algorithm is introduced. The proposed method can be used to handle the additive measurement noise as discussed in the Section 4.2. The proposed recursive SAG (RSAG) algorithm is summarized into two main steps, as of its batch version:

[1] A version of this chapter is under preparation to be submitted to a journal

The SHDP parameter estimation: After constructing the SHDP for multi-mode models, a recursive procedure is needed to estimate the parameters of the SHDP using the current available data point.

Sub-models parameter retrieving: A retrieving procedure that only utilizes the current data point is needed to retrieve the sub-models' parameters from the SHDP parameters.

6.1 The recursive SHDP

As discussed in the Section 4.2, the formulation of the SHDP results in an error-in-variable (EIV) model with regressors error vector ΔX. The covariance matrix of those regressors error vector is dependent on the measured data and change with time. **Algorithm 1** presented in the Section 4.2, can be used to estimate the parameters of the SHDP using the EW-TLS method. In the Chapter 5, the recursive version of the EW-TLS (REW-TLS) method is presented. Therefore, by using the developed REW-TLS method, the recursive estimation of the SHDP can be developed.

Consider the EIV representation of the SHDP:

$$Z_1{}^T \approx \tilde{H}^T \begin{bmatrix} Z_2{}^T \\ \vdots \\ Z_I{}^T \\ \vdots \\ Z^T_{M_n(K)} \end{bmatrix} \tag{6.1}$$

where, $\tilde{H}$ is the estimated model parameter vector and $Z_I, I = 1 \ \cdots \ M_N(k)$ are the mapped version of the data regressors using Veronese variety. Z_I are defined in (4.21) with more details. By following the **Algorithm 2** in the Section 5.4, the recursive updating equation for $\tilde{H}(t)$ can be obtained:

$$\tilde{H}(t) = \tilde{H}(t-1) - [f_t''(\tilde{H}(t)) + \lambda \ \mathrm{diag}\{f_t''(\tilde{H}(t-1))\}\ ]^{-1} g_t^T(\tilde{H}(t-1)), \tag{6.2}$$

where the function $g_t(\cdot)$ is calculated as follows.

$$g_t(\tilde{H}(t-1)) = (2\check{z}_{1t} r_t^T(\tilde{H}) Q_t^{-1}(\tilde{H}) - r_t^T(\tilde{H}) Q_t^{-1}(\tilde{H})\ G_t\ Q_t^{-1}(\tilde{H}) r_t(\tilde{H})) \tag{6.3}$$

where $Q_t(\tilde{H}) = [1 \ \ -\tilde{H}^T] \times V_{\Delta Z_t} \times \begin{bmatrix} 1 \\ -\tilde{H} \end{bmatrix}$, $r_t(\tilde{H}) = \check{z}_{1t} - \tilde{H}^T[\check{z}_{2t} \ \ \cdots \ \ \check{z}_{M_n(K)t}]^T$, $G_t = 2(V_{\Delta Z_t 1} - V_{\Delta Z_t 21}\tilde{H})$, $\check{z}_{It} = z_{It} - E[z_{It}]$, and $V_{\Delta Z_t}$ is partitioned as :

$$V_{\Delta Z_t} = \begin{bmatrix} var(z_{1t}) & cov(z_{1t}, Z_t^c) \\ cov(Z_t^c, z_{1t}) & var(Z_t^c) \end{bmatrix} = \begin{bmatrix} V_{\Delta Z_t 1} & V_{\Delta Z_t 12} \\ V_{\Delta Z_t 21} & V_{\Delta Z_t 2} \end{bmatrix}, \tag{6.4}$$

where $Z_t = [z_{1t} \ \ z_{2t} \ \ \cdots \ \ z_{M_n(K)t}]^T = [z_{1t} \ \ Z_t^{cT}]$. Also, the following equations provide the double derivative term f_t'' for the SHDP:

$$\begin{aligned} f_t''(\tilde{H}(t-1)) =& 2\left(\check{z}_{1t} Q_t^{-1} \check{z}_{1t}^T - \check{z}_{1t} Q_t^{-1} G_t^T Q_t^{-1} r_t\right) \\ & - (f''_{2_1 t} + f''_{2_2 t} + f''_{2_3 t} + f''_{2_4 t} + f''_{2_5 t}) \end{aligned} \tag{6.5}$$

where,

$$\begin{aligned} f''_{2_1 t} &= Q_t^{-1}(G_t) Q_t^{-1} r_t \\ f''_{2_2 t} &= -r_t^T Q_i^{-1}(G_t^T) Q_t^{-1} \otimes ((G_t) Q_t^{-1} r_t) \\ f''_{2_3 t} &= r_t^T Q_t^{-1} (\frac{\partial G_t}{\partial \tilde{H}^T})(I_n \otimes (Q_t^{-1} r_t)) \\ f''_{2_4 t} &= -r_t^T Q_t^{-1}(G_t) Q_t^{-1}(G^T) Q_t^{-1} r_t \\ f''_{2_5 t} &= r_t^T Q_t^{-1}(G_t) Q_t^{-1} \check{z}_{1t}^T \end{aligned} \tag{6.6}$$

where $\frac{\partial G_t}{\partial \tilde{H}^T} = 2V_{\Delta Z_t 1}$. Note that, the centered mapped data points $\check{Z}_t = Z_t - E[\Delta Z_t]$ is used in calculation of g_t and f_t''.

6.2 Retrieving the sub-models' parameters

In the Section 3.2 two methods have been developed for retrieving sub-models parameters from the estimated parameters of the HDP. It is discussed in the Section 4.2 that these method can also be applied on the SHDP. However, these methods require all the available data points in the calculation. Therefore they are mostly suitable for off-line applications. Since, all the data points in the past time are available at the current time in the online applications, these method can be applied by a moving window approach. On the other hand, the calculation time is always an important factor in online applications. Therefore, developing a parameter retrieving method that can estimate

sub-models' parameters by using just the SHDP parameters at the current time is motivating. Two methods are developed to achieve this goal and the are presented in the following sub-sections. The derivations of EW and MW methods developed in the Section 3.2 are followed. The first proposed method in this chapter is only suitable for two-modes models and the second method is applicable to general multi-mode models.

6.2.1 The two-mode models retrieving procedure

Consider the following two-mode switching ARX model:

$$\begin{cases} b_1^T X_t = \varepsilon_t \\ b_2^T X_t = \varepsilon_t \end{cases},$$

and its corresponding SHDP as: $SP_2(X_t) = (b_1^T X_t) \times (b_2^T X_t)$. The first derivative of the SHDP with respect to X_t is:

$$D^{(1)} SP_2(X_t) = \frac{\partial SP_2(X_t)}{\partial X_t} = b_1(b_2^T X_t) + b_2(b_1^T X_t). \tag{6.7}$$

and consequently the second derivative is calculated as:

$$D^{(2)} SP_2(X_t) = \frac{\partial SP_2(X_t)}{\partial X_t \partial X_t^T} = b_1(b_2^T) + b_2(b_1^T). \tag{6.8}$$

It should be noted that a matrix-wise equation can be obtained from (6.8) that does not depend on the data point X_t. From this matrix-wise equation the SARX model parameters can be calculated. This is demonstrated in the following illustrative example.

Illustrative example 6.1: Consider the following EIV representation of the two-mode model:

$$\begin{cases} x_{1t} = \alpha_1 x_{2_t} + \beta_1 x_{3t} \\ x_{1t} = \alpha_2 x_{2_t} + \beta_2 x_{3t} \end{cases} = \begin{cases} b_1^T X_t = 0 \\ b_2^T X_t = 0 \end{cases} \tag{6.9}$$

where $b_i^T = [b_{i1} \;\; b_{i2} \;\; b_{i3}] = [1 \;\; -\alpha_i \;\; -\beta_i]$, $i = 1, \; 2$. The SHDP for the above

example is:

$$\begin{aligned} SP_2(X_t) &= (x_{1t} - \alpha_1 x_{2_t} - \beta_1 x_{3t}) \times (x_{1t} - \alpha_2 x_{2_t} - \beta_2 x_{3t}) \\ &= H_1 {x_{1t}}^2 + H_2 x_{1t} x_{2_t} + H_3 x_{1t} x_{3t} + H_4 {x_{2_t}}^2 + H_5 x_{2_t} x_{3t} + H_6 {x_{3t}}^2 \end{aligned} \tag{6.10}$$

Where H is the parameters vector of the SHDP. Therefore the corresponding first and second derivative of the above SHDP with respect to $X_t = [x_{1t} \;\; x_{2_t} \;\; x_3 t]$ are:

$$D^{(1)} SP_2(X_t) = \begin{bmatrix} 2H_1 x_{1t} + H_2 x_{2_t} + H_3 x_{3t} \\ H_2 x_{1t} + 2H_4 x_{2_t} + H_5 x_{3t} \\ H_3 x_{1t} + H_5 x_{2_t} + 2H_6 x_{3t} \end{bmatrix}, \tag{6.11}$$

and

$$D^{(2)} SP_2(X_t) = \begin{bmatrix} 2H_1 & H_2 & H_3 \\ H_2 & 2H_4 & H_5 \\ H_3 & H_5 & 2H_6 \end{bmatrix}, \tag{6.12}$$

Therefore the matrix-wise equation (6.8) for the *illustrative example 6.1* is found as:

$$\begin{aligned} A = D^{(2)} SP_2(X_t) &= \begin{bmatrix} 2H_1 & H_2 & H_3 \\ H_2 & 2H_4 & H_5 \\ H_3 & H_5 & 2H_6 \end{bmatrix} \\ &= \begin{bmatrix} 2(b_{11}b_{21}) & b_{11}b_{22} + b_{21}b_{12} & b_{11}b_{23} + b_{21}b_{13} \\ b_{12}b_{21} + b_{22}b_{11} & 2(b_{12}b_{22}) & b_{12}b_{23} + b_{22}b_{13} \\ b_{13}b_{21} + b_{23}b_{11} & b_{13}b_{22} + b_{23}b_{12} & 2(b_{13}b_{23}) \end{bmatrix} \end{aligned} \tag{6.13}$$

considering $b_{11} = b_{21} = 1$ and defining A_{ij} as the element of A at row i and column j, the above equation is written as:

$$\begin{aligned} A_{12} &= b_{12} + b_{22} \rightarrow b_{12} = A_{12} - b_{22} \\ A_{13} &= b_{13} + b_{23} \rightarrow b_{13} = A_{13} - b_{23}, \end{aligned} \tag{6.14}$$

and,

$$\begin{aligned} A_{22} &= 2b_{12}b_{22} \rightarrow A_{22} = 2(A_{12} - b_{22})b_{22} \rightarrow 2b_{22}^2 - 2A_{12}b_{22} + A_{22} = 0 \\ A_{33} &= 2b_{13}b_{23} \rightarrow A_{33} = 2(A_{13} - b_{23})b_{23} \rightarrow 2b_{23}^2 - 2A_{13}b_{23} + A_{33} = 0. \end{aligned} \tag{6.15}$$

Therefore, the following solutions for b_{22} and b_{23} are obtained:

$$\begin{aligned} b_{22} &= \frac{A_{12} \pm \sqrt{A_{12}^2 - 2A_{22}}}{2} \\ b_{23} &= \frac{A_{13} \pm \sqrt{A_{13}^2 - 2A_{33}}}{2} \end{aligned} \tag{6.16}$$

By choosing the plus sign for both of the equations above, the parameters vector b_2 is retrieved and consequently the vector b_1. A general formula for finding each element of sub-model vectors b_1 and b_2 is then summarized as follows:

$$\begin{aligned} b_{21} &= 1 \\ b_{2i} &= \frac{A_{1i} + \sqrt{A_{1i}^2 - 2A_{ii}}}{2} \\ b_{1i} &= A_{1i} - b_{2i}, \end{aligned} \tag{6.17}$$

where $i = 2, \cdots, k$ and k is the number of the parameters in the sub-model. Note that, matrix A in (6.13) only depends on the estimated parameters of SHDP. Therefore, the retrieving procedure for the two-modes models in the online application does not depend on any data point. The calculations are easy and straightforward.

6.2.2 The Multi-mode model retrieving procedure

First, an illustrative example for a three-mode model is used to demonstrate the retrieving procedure more clearly, then the general formula is represented. *Illustrative example 6.2: a three-mode model* Consider the following three-mode SARX model:

$$\begin{cases} b_1^T X_t = \varepsilon_t \\ b_2^T X_t = \varepsilon_t \\ b_3^T X_t = \varepsilon_t \end{cases} \tag{6.18}$$

where $X_t = [x_{1t} \quad x_{2t} \quad x_{3t}]$. In this section for simplicity the subscript t is omitted from x_{it}, i.e. denote $x_i \equiv x_{it}$. The corresponding SHDP for the above model is:

$$SP_3(X_t) = (b_1^T X_t) \times (b_2^T X_t) \times (b_2^T X_t), \tag{6.19}$$

and the first, second and third derivative of $SP_3(X_t)$ is calculated as follow:

$$D^{(1)}SP_3(X_t) = \frac{\partial SP_3(X_t)}{\partial X_t} = b_1(b_2^T X_t)(b_3^T X_t)+b_2(b_1^T X_t)(b_3^T X_t)+b_3(b_1^T X_t)(b_2^T X_t), \tag{6.20}$$

$$\begin{aligned} D^{(2)}SP_3(X_t) &= \frac{\partial SP_3(X_t)}{\partial X_t \partial X_t^T} \\ &= b_1 b_2^T (b_3^T X_t) + b_2 b_1^T (b_3^T X_t) + b_1 b_3^T (b_2^T X_t) + b_3 b_1^T (b_2^T X_t) \\ &\quad + b_2 b_3^T (b_1^T X_t) + b_3 b_2^T (b_1^T X_t) \end{aligned} \tag{6.21}$$

$$\begin{aligned} D^{(3)}SP_3(X_t) &= \frac{\partial SP_3(X_t)}{\partial X_t \partial X_t^T \partial X_t} \\ &= b_1 b_2^T \otimes b_3 + b_2 b_1^T \otimes b_3 + b_1 b_3^T \otimes b_2 + b_3 b_1^T \otimes b_2 \\ &\qquad + b_2 b_3^T \otimes b_1 + b_3 b_2^T \otimes b_1 \\ &= (b_1 b_2^T + b_2 b_1^T) \otimes b_3 + (b_1 b_3^T + b_3 b_1^T) \otimes b_2 + (b_2 b_3^T + b_3 b_2^T) \otimes b_1 \end{aligned} \tag{6.22}$$

By following similar procedure in the Section 3.2.2, and defining $B = [b_1 \quad b_2 \quad b_3]$, $e_i,\ i = 1,\ 2,\ 3$ as a unit vector with length 3 which is equal to zero except its i^{th} element, and $b_i = Be_i$ the following is obtained from equation (6.22):

$$(b_i b_j^T + b_i b_j^T) = B\,[e_i\ e_j] \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} e_i^T \\ e_j^T \end{bmatrix} B^T = BC_{ij}B^T \tag{6.23}$$

where $i, j = 1,\ 2,\ 3$. Using the above equation in (6.22) results in,

$$D^{(3)}SP_3(X_t) = BC_{12}B^T \otimes Be_3 + BC_{13}B^T \otimes Be_2 + BC_{23}B^T \otimes Be_1 \tag{6.24}$$

By using Kronecker property (2.14c) described in the Section 2.2 and the fact that e_i is a vector, the above equation is written as:

$$\begin{aligned}
&D^{(3)}SP_3(X_t)\\
&= (B\otimes B)\big(C_{12}B^T\otimes e_3\big)+(B\otimes B)\big(C_{13}B^T\otimes e_2\big)+(B\otimes B)\big(C_{23}B^T\otimes e_1\big)\\
&= (B\otimes B)\Big(\big(C_{12}\otimes e_3\big)\big(B^T\otimes 1\big)+\big(C_{13}\otimes e_2\big)\big(B^T\otimes 1\big)+\big(C_{23}\otimes e_1\big)\big(B^T\otimes 1\big)\Big)\\
&= (B\otimes B)\big(C_{12}\otimes e_3 + C_{13}\otimes e_2 + C_{23}\otimes e_1\big)B^T\\
&= (B\otimes B)\begin{bmatrix} C_{23}\\ C_{13}\\ C_{12}\end{bmatrix}B^T
\end{aligned} \tag{6.25}$$

The equation (6.25) is a matrix-wise equation independent from the data points X_t. This can be illustrated by calculating the third order derivative of its left hand side. The muli-monomial representation of the SHDP is as follows:

$$\begin{aligned}
SP_3(X_t) = {} & H_1x_1^3 + H_2x_1^2x_2 + H_3x_1^2x_3 + H_4x_1x_2^2 + H_5x_1x_2x_3 + H_6x_1x_3^2\\
& + H_7x_2^3 + H_8x_2^2x_3 + H_9x_2x_3^2 + H_{10}x_3^3.
\end{aligned} \tag{6.26}$$

The first derivative of the above with respect to X_t is calculated as:

$$D^{(1)}SP_3(X_t) = \begin{bmatrix} 3H_1x_1^2 + 2H_2x_1x_2 + 2H_3x_1x_3 + H_4x_2^2 + H_5x_2x_3 + H_6x_3^2\\ H_2x_1^2 + 2H_4x_1x_2 + H_5x_1x_3 + 3H_7x_2^2 + 2H_8x_2x_3 + H_9x_3^2\\ H_3x_1^2 + H_5x_1x_2 + 2H_6x_1x_3 + H_8x_2^2 + 2H_9x_2x_3 + 3H_{10}x_3^2.\end{bmatrix}, \tag{6.27}$$

The second derivative is:

$$D^{(2)}SP_3(X_t) =$$
$$\begin{bmatrix} 6H_1x_1 + 2H_2x_2 + 2H_3x_3 & 2H_2x_1 + 2H_4x_2 + H_5x_3 & 2H_3x_1 + H_5x_2 + 2H_6x_3\\ 2H_2x_1 + 2H_4x_2 + H_5x_3 & 2H_4x_1 + 6H_7x_2 + 2H_8x_3 & H_5x_1 + 2H_8x_2 + 2H_9x_3\\ 2H_3x_1 + H_5x_2 + 2H_6x_3 & H_5x_1 + 2H_8x_2 + 2H_9x_3 & 2H_6x_1 + 2H_9x_2 + 6H_{10}x_3\end{bmatrix}, \tag{6.28}$$

and the third derivative is obtained as:

$$\Lambda = D^{(3)}SP_3(X_t) = \begin{bmatrix} \begin{bmatrix} 6H_1 \\ 2H_2 \\ 2H_3 \end{bmatrix} & \begin{bmatrix} 2H_2 \\ 2H_4 \\ H_5 \end{bmatrix} & \begin{bmatrix} 2H_3 \\ H_5 \\ 2H_6 \end{bmatrix} \\ \begin{bmatrix} 2H_2 \\ 2H_4 \\ H_5 \end{bmatrix} & \begin{bmatrix} 2H_4 \\ 6H_7 \\ 2H_8 \end{bmatrix} & \begin{bmatrix} H_5 \\ 2H_8 \\ 2H_9 \end{bmatrix} \\ \begin{bmatrix} 2H_3 \\ H_5 \\ 2H_6 \end{bmatrix} & \begin{bmatrix} H_5 \\ 2H_8 \\ 2H_9 \end{bmatrix} & \begin{bmatrix} 2H_6 \\ 2H_9 \\ 6H_{10} \end{bmatrix} \end{bmatrix}. \tag{6.29}$$

Therefore the following equation does not depend on the the data points X_t and only depends on the parameters of the SHDP:

$$\Lambda_{9\times 3} = (B \otimes B)_{9\times 9} \begin{bmatrix} C_{23} \\ C_{13} \\ C_{12} \end{bmatrix}_{9\times 3} B^T_{3\times 3} \tag{6.30}$$

The solution to the above matrix-wise equation for $\tilde{H}(t)$ at each data point is the retrieved SARX sub-models' parameters at that point. This equation can be solved by using "fminunc" MATLAB function using the following cost function:

$$\begin{aligned} &\left\| \Lambda - (B \otimes B) \begin{bmatrix} C_{23} \\ C_{13} \\ C_{12} \end{bmatrix} B^T \right\|_2 = 0 \\ &\text{S.T.:} \quad B_{1i} = 1 \quad i \in \{1, 2, 3\} \end{aligned} \tag{6.31}$$

6.2.3 The General formula

In this section, by following the procedure demonstrated in illustrative example 6.2 and the derivation steps presented in the Section 3.2.2, a general retrieving equation is derived.

Assume a multi-mode model to have n different modes. The n^{th} derivative of the SHDP for his model $D^{(n)}SP_n(X_t)$ is calculated as follows: when n is an

even number,

$$\begin{aligned} D^n SP_n(X_t) &= \frac{\partial^n SP_n(X_t)}{\partial X_t \partial x_t^T \partial X_t \partial x_t^T \cdots \partial X_t \partial x_t^T} \\ &= \sum_{(i_k,i_j)\in {}_nP_n} \left(\coprod (b_{i_k} b_{i_j}^T) \right) \end{aligned} \tag{6.32}$$

and when n is an odd number,

$$\begin{aligned} D^n SP_n(X_t) &= \frac{\partial^n SP_n(X_t)}{\partial X_t \partial x_t^T \partial X_t \partial x_t^T \cdots \partial X_t} \\ &= \sum_{(i_k,i_j)\in {}_nP_{n-1} | i_l \notin {}_nP_{n-1}} \left(\coprod (b_{i_k} b_{i_j}^T) \otimes b_{i_l} \right) \end{aligned} \tag{6.33}$$

where ${}_nP_n$ is all permutation sets of n numbers from the set of $\{1, \cdots, n\}$, and $(i_k, i_j) \in {}_nP_n$ is defined as the pairs of numbers in each arrangement. Also ${}_nP_{n-1}$ is all permutation sets of $n-1$ numbers from the set of $\{1, \cdots, n\}$, and in each arrangement the number that does not belong to that permutation is equal to i_l. The following example clarifies this notation.

Illustrative example 6.3: Consider the set of $\{1, 2, 3, 4, 5\}$. The ${}_5P_4$ is all the different arrangement of four number from the set $\{1, 2, 3, 4, 5\}$ (i.e. $\{1, 2, 3, 4\}$, $\{2, 1, 3, 4\}$, $\{3, 1, 5, 2\}$). The pairs (i_k, i_j) are constructed by selecting the numbers inside each arrangement as pairs (i.e. $\{(1, 2), (3, 4)\}$, $\{(2, 1), (3, 4)\}$, $\{(3, 1), (5, 2)\}$). Then, the i_l is the 5^{th} number that is not selected in each arrangement(i.e. $\{(1, 2), (3, 4)\}$ $i_l = 5$, $\{(2, 1), (3, 4)$ $i_l = 5\}$, $\{(3, 1), (5, 2)\}$ $i_l = 4$).

Assuming that ${}_n\acute{P}_{n-1}$ and ${}_n\acute{P}_n$ represents sets of permutations that the order inside each pair of numbers is not important
(i.e. the set $\{(1, 2), (3, 4)\}$ $i_l = 5$, $\{(2, 1), (3, 4)\}$ $i_l = 5$, $\{(1, 2), (4, 3)\}$ $i_l = 5$ and $\{(2, 1), (3, 2)\}$ $i_l = 5$ are considered as the same), then the n^{th} derivative of the SHDP is written as the following equations. When n is even,

$$\begin{aligned} D^n SP_n(X_t) &= \frac{\partial^n SP_n(X_t)}{\partial X_t \partial x_t^T \partial X_t \partial x_t^T \cdots \partial X_t \partial x_t^T} \\ &= \sum_{(i_k,i_j)\in {}_n\acute{P}_n} \left(\coprod (b_{i_k} b_{i_j}^T + b_{i_j} b_{i_k}^T) \right), \end{aligned} \tag{6.34}$$

and when n is odd,

$$\begin{aligned} D^n SP_n(X_t) &= \frac{\partial^n SP_n(X_t)}{\partial X_t \partial x_t^T \partial X_t \partial x_t^T \cdots \partial X_t} \\ &= \sum_{(i_k,i_j)\in_n \acute{P}_{n-1} | i_l \notin_n \acute{P}_{n-1}} \left(\coprod (b_{i_k} b_{i_j}^T + b_{i_j} b_{i_k}^T) \otimes b_{i_l} \right) \end{aligned} \tag{6.35}$$

The e_{i_j} is defined as a unit vector with length n in which, the i_j^{th} element is equal to one, and other elements equal to zero. Considering, $B = [b_1 \ , \ \cdots \ , \ b_n]$, and $\begin{bmatrix} e_{i_k} & e_{i_j} \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} e_{i_k}^T \\ e_{i_j}^T \end{bmatrix} = C_{i_k i_j}$ the above derivatives are changed to the followings:

$$D^n SP_n(X_t) = \sum_{(i_k,i_j)\in_n \acute{P}_n} \left(\coprod B C_{i_k i_j} B^T \right) \tag{6.36}$$

when n is even, and

$$D^n SP_n(X_t) = \sum_{(i_k,i_j)\in_n \acute{P}_{n-1} | i_l \notin_n \acute{P}_{n-1}} \left(\coprod B C_{i_k i_j} B^T \otimes B e_{i_l} \right) \tag{6.37}$$

when n is odd. By using the Kroneker product property (2.14c) and the **Lemma 2.1** in (2.2), the above equations are changed to:

$$\Lambda = D^n SP_n(X_t) = (\coprod_{l=1}^{n/2} B) \times \left(\sum_{(i_k,i_j)\in_n \acute{P}_n} (\coprod C_{i_k i_j}) \right) \times (\coprod_{l=1}^{(n)/2} B^T) \tag{6.38}$$

when n is even, and

$$\begin{aligned} \Lambda &= D^n SP_n(X_t) \\ &= (\coprod_{l=1}^{(n-1)/2} B) \times \left(\sum_{(i_k,i_j)\in_n \acute{P}_{n-1} | i_l \notin_n \acute{P}_{n-1}} (\coprod C_{i_k i_j}) \otimes e_{i_l} \right) \times (\coprod_{l=1}^{(n+1)/2} B^T) \end{aligned} \tag{6.39}$$

when n is odd. The solutions to the following minimizations are the retrieved

SARX parameters of the sub-models.

$$\left\|\Lambda - (\coprod_{l=1}^{n/2} B) \times \left(\sum_{(i_k,i_j)\in {}_n\acute{P}_n} (\coprod C_{i_k i_{k+1}}) \right) \times (\coprod_{l=1}^{(n)/2} B^T)\right\|_2 = 0 \qquad (6.40)$$
$$\text{S.T.:} \quad B_{1i} = 1 \quad i \in \{1, \cdots, n\}$$

when n is even, and

$$\left\|\Lambda - (\coprod_{l=1}^{(n-1)/2} B) \times \left(\sum_{(i_k,i_j)\in {}_n\acute{P}_{n-1} | i_l \notin {}_n\acute{P}_{n-1}} (\coprod C_{i_k i_j}) \otimes e_{i_l} \right) \times (\coprod_{l=1}^{(n+1)/2} B^T)\right\|_2 = 0$$
$$\text{S.T.:} \quad B_{1i} = 1 \quad i \in \{1, \cdots, n\} \qquad (6.41)$$

when n is odd. The "fminunc" function in MATLAB can be used for solving the above equations. Each column of B represents one of the parameters vectors for the sub-models.

Remark 6.1: It is important to discuss he differences between the equations for retrieving sub-models' parameter, provided in this chapter and in the Section 3.2, despite their similarities. In the derivation of the equations in the Section 3.2 all the available data points are used. This requires solving k total least square optimization, using SVD approach, in the EW method and $k(k-1)(k-2)\cdots(k-n)$ total least square optimization for MW method. It is illustrated that using all the data points decreases the estimation error variance. However, it requires more computation. Therefore it's mostly suitable for off line methods. The equations provided in this chapter, does not require solving any total least square optimization in prior. Thus, they are more suitable for online applications. However, only using the current data point might decrease the retrieving performance. Therefore, in the applications that sampling time is not fast, the methods delivered in the Section 3.2 are preferred.

6.3 The RSAG algorithm

In summary the following algorithm is provided:

Algorithm 3: *Recursive stochastic algebraic geometric (RSAG) algorithm*

Initialization:

1. construct the SHDP.
2. Find an initial approximation for $\tilde{H}$ using the methods developed in the Section 4.
3. Choose a suitable initialization values for $f_0''(0)$, λ and $\nu > 1$.

Main Algorithm:

1. Calculate $g_t(\tilde{H}(t-1))$ from equation (6.3).
2. Calculate $f_t''(\tilde{H}(t-1))$ from (6.5).
3. Update

$$\tilde{H}(t) = \tilde{H}(t-1) - [f_t''(\tilde{H}(t-1)) + \lambda \text{ diag}[f_t''(\tilde{H}(t-1))]\,]^{-1} g_t^T(\tilde{H}(t-1)).$$

4. If $f_t(\tilde{H}(t)) < f_t(\tilde{H}(t-1))$ then $(\lambda = \lambda/\ \nu)$ else $(\lambda = \lambda \times \nu)$
5. Recover the parameters of the sub-models using one of the recovering algorithms discussed in the Section 6.2.
6. repeat for the next point

6.4 Simulation results

6.4.1 The performance comparison of the RSAG and RAG

In this section the performance on the developed recursive method is evaluated with respect to the method proposed in [36]. Consider the following model:

$$\text{Simulation Example 1: } \begin{cases} y_t^0 = -0.9 y_{t-1}^0 + u_{t-1}^0 \\ y_t^0 = 0.7 y_{t-1}^0 - u_{t-1}^0 \end{cases} \tag{6.42}$$

which is introduced in [36]. The input signal is $\mathcal{N}(0,1)$ and the output measurement noise have a Gaussian distribution $\mathcal{N}(0,\sigma^2)$. The initial parameters

Table 6.1: The mean of estimated parameters for two-mode model using SAG and AG

		α_1	β_1	α_2	β_2
$\sigma = 0.1$	The RSAG approach	-0.9012	1.0040	0.6957	-0.9829
	The RAG approach	-0.8993	1.0099	0.6820	-0.9895
$\sigma = 0.2$	The RSAG approach	-0.8961	0.9965	0.6901	-0.9812
	The RAG approach	-0.8886	0.9971	0.7138	-1.0204
$\sigma = 0.3$	The RSAG approach	-0.8953	0.9732	0.6630	0.9801
	The RAG approach	-0.9061	0.9431	0.6475	-0.9992

for the developed RSAG approach are estimated using 100 data points for training data. 100 independent executions are performed under different noise conditions: $\sigma = [0.1 \;\; 0.2 \;\; 0.3]$. The mean of estimated SARX sub-models' parameters after 1000 samples are reported in table 6.1. For comparing purpose, the results reported in [36] are also mentioned.

Also the recursive updated SHDP parameters values ($\tilde{H}$) for the above model under different noise conditions for one execution are reported in figures 6.1, 6.2 and 6.3. It is demonstrated that the parameters of the SHDP under all three different noise conditions approach to the true estimated value, using the proposed RSAG approach. The retrieved β_i parameter values at each point is also reported in figure 6.4 ($\sigma = 0.1$).

It is illustrated in the figure 6.4 that, the RSAG approach delivers more robust recursive estimation with respect to the RAG approach.

6.4.2 The retrieving procedures comparison

In order to compare the performance of the retrieving procedures delivered in this chapter and in the Section 3.2, another simulation study has been performed.

In this study 100 independent executions are executed for the same model with different noise scenarios: $\sigma = [0.1 \;\; 0.2 \;\; 0.3]$. For retrieving procedures, both methods developed in this chapter (called as one-point EW (OEW) and method developed in the Section 3.2 are used. The mean of estimated parameters for sub-models using both methods are presented in table 6.2, and the variance of estimation error is reported in the table 6.3.

Although both method estimate the sub-models' parameters accurately,

Table 6.2: The mean of estimated sub-models' parameters using OEW and EW

		α_1	β_1	α_2	β_2
$\sigma = 0.1$	The OEW approach	-0.9012	1.0040	0.6957	-0.9829
	The EW approach	-0.9014	1.0032	0.7012	-0.9906
$\sigma = 0.2$	The OEW approach	-0.8961	0.9965	0.6901	-0.9812
	The EW approach	-0.9051	1.0062	0.7085	-1.0151
$\sigma = 0.3$	The OEW approach	-0.8953	0.9732	0.6630	0.9801
	The EW approach	-0.9053	0.9695	0.7312	-0.9815

Table 6.3: The variance of sub-models' parameters estimation error using OEW and EW

		α_1	β_1	α_2	β_2
$\sigma = 0.1$	The OEW approach	0.0006	0.0008	0.0012	0.0003
	The EW approach	0.0007	0.0006	0.0009	0.0004
$\sigma = 0.2$	The OEW approach	0.0035	0.0081	0.0421	0.0018
	The EW approach	0.0012	0.0018	0.0434	0.0009
$\sigma = 0.3$	The OEW approach	0.0087	0.0146	0.0891	0.0024
	The EW approach	0.0026	0.0037	0.0264	0.0013

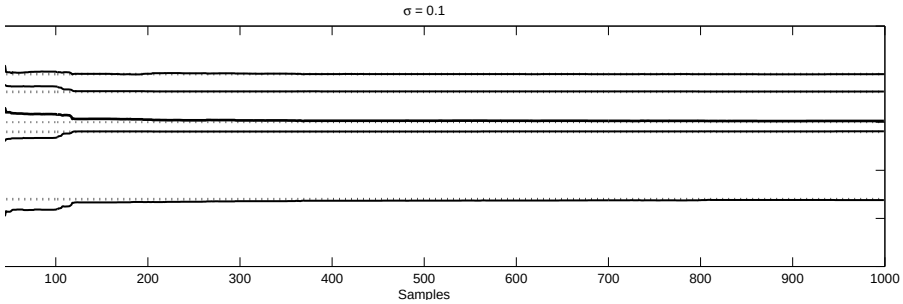

Figure 6.1: The recursive estimation of the SHDP paramters, $\sigma = 0.1$

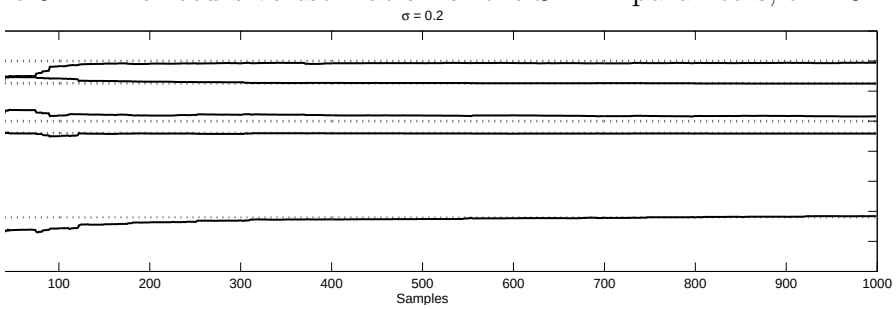

Figure 6.2: The recursive estimation of the SHDP paramters, $\sigma = 0.2$

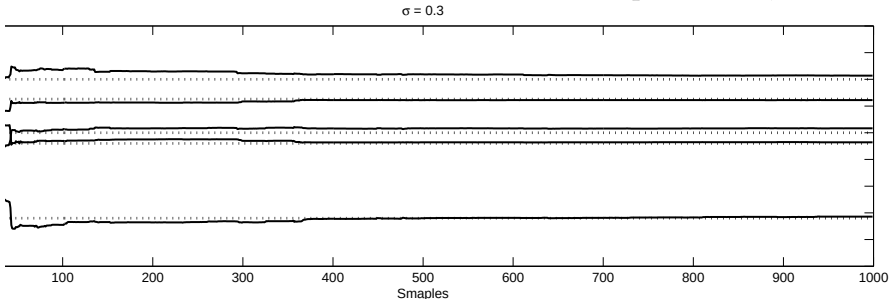

Figure 6.3: The recursive estimation of the SHDP paramters, $\sigma = 0.3$

but it can be seen in table 6.2 and table 6.3 that the variance of the sub-models' parameters estimation error are lower using EW method. This shows the advantage of using more data points in retrieving the sub-models' parameters. However, the EW method requires more computation time. Therefore, if the sampling time is very fast the batch version cannot be used.

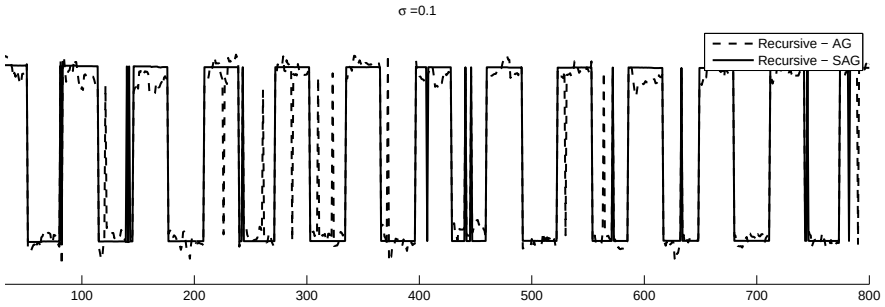

Figure 6.4: The recursive estimation of the β_i parameter, $\sigma = 0.1$

Chapter 7

Fault Detection Application

[1] The International Federation of Automation Control (IFAC) SAFEPROCESS technical committee defines a fault as an unpermitted deviation of at least one of the parameters or properties of the system from its normal or standard conditions [74]. This deviation can occur in sensor, actuator or any system component. A fault detection algorithm is used to detect the occurrence of a malfunction. The isolating procedure tries to identify the location or nature of the fault. A method which aims to detect and isolate a malfunction is referred to as the fault detection and isolation (FDI) procedure.

There are vast amount of different FDI methods and algorithms in the literature, and many different surveys have been published summarizing these methods [74, 75]. The hybrid system field of research is no exception to this rich literature. In [76], a model based methodology is introduced for online tracking and diagnosis of hybrid systems. Another model-based method is presented in [77] that detects and identifies the actuator faults in the switching systems. Simani et al. developed a method to detect the fault in the pitch sensors in the turbine blade system of the wind turbines [78]. Estimating sub-models' parameters for residual generation adds additional complexity to the mentioned method. In general, the complex nature of hybrid systems dictates complexity to the proposed FDI methods.

In this section a FDI method, which is independent of sub-models' parameters estimation, is developed for switching multi-mode models. This allows us to introduce a simpler yet more robust fault detection methodology. The change in the parameters of each sub-model results in the change of the pa-

[1] A version of this chapter has been submitted to the IEEE transaction of control systems technology journal.

rameters of the "lifted" representation of the switching model (SHDP) which is independent from the switching sequence. Therefore, by tracking the changes in the parameters of the SHDP the occurrence of a fault can be detected. The deviation of the parameters in the SHDP from their nominal values can be detected and isolated via the local approach. The local approach has been shown to have good performance for detecting small faults [74]. This approach is first introduced in [79] and improved and completed in [80]. This approach has been considered in different applications [81, 82].

The fault defined in this section is any abrupt change in the parameters of the sub-models in the multi-mode models. This change can be reflecting a system fault, in most of the cases, or actuator failures in the plants. In the proposed method, it is shown that without identifying the parameters in the sub-models the change in these parameters can be detected and isolated. The performance and robustness of the proposed FDI method are illustrated with simulation results.

7.1 The local approach

A residual in a fault detection algorithm is a signal generated from input $u(t)$ and output $y(t)$ [74]:

$$r(t) = g(u(t), y(t)) \tag{7.1}$$

In the model based fault detection, a common practice is to define $r(t)$ as the difference between the output $y(t)$ and the predicted output $\hat{y}(t)$ from the model,

$$r(t) = y(t) - \hat{y}(t) \tag{7.2}$$

Hence, a vector-valued function $K(B, X_t)$ with a finite-dimension can be used as a valid residual for monitoring the changes in vector B, if it is differentiable in B, and if there exists a neighborhood $\omega(B^0)$ such that [80]:

$$\begin{aligned} &\mathbf{E}_{B^0}[K(B, X_t)] = 0 \quad if B = B^0 \\ &\mathbf{E}_{B^0}[K(B, X_t)] \neq 0 \quad if B \notin \omega(B^0) \end{aligned} \tag{7.3}$$

where $\mathbf{E}_{B^0}$ is the expected value when the actual system parameter value is B^0. In the statistical evaluation of residuals, possess knowledge about $K(B, X_t)$ is required. However, due to the complexity of the residuals, the distribution of $K(B, X_k)$ is most likely unknown. Therefore, the following approximated residual has been considered in the local approach [80]:

$$\zeta_N(B) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} K(B, X_t) \tag{7.4}$$

And, the following hypothesis test is defined:

$$\begin{aligned} &\text{Normal Operation} \quad \mathbf{H}_0 : \ B = B^0 \\ &\text{Faulty Operation} \quad \mathbf{H}_a : \ B = B^0 + \frac{\Upsilon}{\sqrt{N}} \end{aligned} \tag{7.5}$$

where Υ is an arbitrary fixed vector with a small value and the same dimension as B. Also, for isolation purpose, the following mean deviation is defined:

$$\mathcal{M}(B^0) = -\frac{\partial}{\partial B} \sum_{t=1}^{N} K(B, X_t) \Big|_{B=B^0} \tag{7.6}$$

Using the central limit theorem [83], the residual defined in (7.4), under mild conditions [80], is asymptotically Gaussian distributed with zero mean when the system is at no fault state, and with mean $M\Upsilon$ when the system is at a faulty state. Also, the covariance matrix of (7.4) remains the same for both normal and faulty modes.

If the function $G(B, X_t)$ is used for estimating the model parameters B via following minimization:

$$\hat{B} = \arg\min_{B} \sum_{t=1}^{N} G(B, X_t), \tag{7.7}$$

then the gradient of the $G(B, X_t)$ can be used as the the function $K(B, X_t)$. Hence,

$$\frac{\partial}{\partial B} G(B, X_t) = K(B, X_t). \tag{7.8}$$

Therefore, the Hessian matrix of this cost function is used in constructing $\mathcal{M}$:

$$\mathcal{M}(B) = -\frac{\partial^2}{\partial B \partial B^T} \sum_{t=1}^{N} G(B, X_t) \tag{7.9}$$

Also, the covariance of this residual can be determined as:

$$\begin{aligned} \Sigma(B) = \lim_{N\to\infty} \Sigma_N(B), \quad \Sigma_N(B) &= \mathbf{E}_{B^0}[\zeta_N(B)\zeta_N^T(B)] \\ &\approx \frac{1}{N} \sum_{k=1}^{N} \sum_{j=1}^{N} \mathbf{E}_{B^0}[K(B, X_k)K^T(B, X_j)] \end{aligned} \tag{7.10}$$

7.2 Residual construction for SARX models

7.2.1 A motivating discussion

The problem of fault detection and isolation for SARX models is more complicated than single ARX models. A fault can occur in one of the sub-models while other operating sub-models remain under normal conditions. One trivial approach for detecting the fault for SARX models is to investigate the fault occurrence in each of the sub-models individually. Consider the following EIV representation of a SARX model:

$$\begin{cases} X_{1t} \approx {b_1}^T X_{2t} \\ X_{1t} \approx {b_2}^T X_{2t} \\ \quad \vdots \\ X_{1t} \approx {b_n}^T X_{2t} \end{cases} \tag{7.11}$$

where $b_i,\ i = 1,\ \cdots,\ n$, are the parameter sets for each sub-model. X_{1t} and X_{2t} are defined in the Section 2.1. Following the definition of the residual (7.2), n residuals have to be constructed for the n sub-models in (7.11):

$$\begin{cases} r_1(t) = X_{1t} - \hat{X}_{1t}^1 = X_{1t} - \hat{b}_1^T X_{2t} \\ r_2(t) = X_{1t} - \hat{X}_{1t}^2 = X_{1t} - \hat{b}_2^T X_{2t} \\ \vdots \\ r_n(t) = X_{1t} - \hat{X}_{1t}^n = X_{1t} - \hat{b}_n^T X_{2t} \end{cases} \tag{7.12}$$

where $\hat{b}_i,\ i = 1,\ \cdots,\ n$ are the estimated parameter sets for each sub-model, and $\hat{X}^i_{1t},\ i = 1,\ \cdots,\ n$ are the predicted outputs using each sub-model's estimated parameters. Only one of the above sub-models represents the SARX model at time t; therefore only one of the residuals is near zero and the rest are nonzero and can be possibly large numbers:

$$\begin{cases} r_{\lambda_t}(t) = \epsilon_t \\ \\ r_i(t) = e_i(t) \qquad i \neq \lambda_t \end{cases} \tag{7.13}$$

where $E[\epsilon_t] = 0$ and can be considered as the measurement noise. The e_i is the error caused by the estimated outputs from the sub-models that are not effective at time t and $E[e_i] \neq 0$. A mathematical method should be used to synthesize the above residuals. For instant, the minimum of the above residuals can be used as follow:

$$r(t) = \min_i r_i(t) \tag{7.14}$$

Therefore $r(t)$ is always a small value in normal condition and $E[r] = 0$. When a fault happens, all the sub-models in (7.11) are no longer valid. Hence, all the residuals $r_i(t),\ i = 1,\ ,\cdots,n$, produce large values. The Deviation of $E[r]$ from small numbers can be interpreted as a fault.

The above approach is simple yet has several disadvantages:

- The parameters of the sub-models have to be estimated prior to the residual construction. This is a complicated process for SARX models.
- n residuals have to be calculated and using the minimum residual is not necessarily best option for combining these residuals.
- In occurrence of a fault, the first step to isolate the fault is to detect the abnormal mode of operation. Since all the sub-models are generating deviated residuals, finding the faulty mode is not a straightforward process. After detecting the faulty mode, the isolation has to be performed.

The switching sequence in the SARX models increases the complexity of the parameter estimation problem and consequently the complexity of the FDI

problem. Hence, developing a FDI method independent of switching sequence is the objective of this chapter.

7.2.2 The SHDP based residual

As discussed in the Section 4.2, a promising methodology for eliminating the switching sequence in the multi-mode switching models is to embed all the sub-models into a "lifted" model. It is shown that if the inputs and output are corrupted with measurement noise, the "lifted" model, called "SHDP", describes all the sub-models by eliminating the switching sequence:

$$\begin{aligned} SP_n(X_t) &= \prod_{i=1}^{n}((X_{1t} - \Delta X_{1t}) - b_i^T(X_{2t} - \Delta X_{2t})) \\ &= \sum_{I=1}^{M_n(K)} h_I^0 z_{It} = H^{0^T}\vartheta_n(X_t - \Delta X_t) = 0. \end{aligned} \tag{7.15}$$

where $X_t = \begin{bmatrix} X_{1t} \\ X_{2t} \end{bmatrix}$ is the available data point vector at time t and $\Delta X_t = \begin{bmatrix} \Delta X_{1t} \\ \Delta X_{2t} \end{bmatrix}$ is the additive measurement noise using the same assumptions in the Section 4.2. Therefore, the following minimization delivers a consistent estimation for H:

$$\begin{aligned} & f_t(\tilde{H}) = r_t^T(\tilde{H})Q_t^{-1}(\tilde{H})r_t(\tilde{H}) \\ & \min_{\tilde{H}} \sum_{t=1}^{N} f_t(\tilde{H}) \end{aligned} \tag{7.16}$$

where $Q_t(\tilde{H}) = [1 \quad -\tilde{H}^T] \times V_{\Delta Z_t} \times \begin{bmatrix} 1 \\ -\tilde{H} \end{bmatrix}$, $r_t(\tilde{H}) = \check{z}_{1t} - \tilde{H}^T[\check{z}_{2t} \ \cdots \ z_{M_n(K)t}]^T$ and $H = [1 \quad -\tilde{H}]$. $V_{\Delta Z_t}$ is the covariance matrix of the mapped data points and can be calculated following the steps in the Section 4.3.2.

In the normal condition, one of the multiplied brackets in (7.15) is equal to zero. Consequently, the right hand side of the equation is always equal to zero regardless of current active mode of operation. When the active mode is faulty, none of the multiplied brackets is equal to zero resulting in a deviation

of the right hand side from zero. Hence, the impact of the fault in one or more of the sub-models can be observed in the SHDP. This deviation can be used for the residual generation for application of the local approach. As discussed in the Section 7.1, since (7.16) estimates the parameters for the SHDP, the gradient of the cost function (7.16) can be used as the residual vector. This gradient is calculated in the Section 5.2, and the local approach residual for the SHDP is as follows:

$$\begin{aligned}\zeta_N(\tilde{H}) &= \frac{1}{\sqrt{N}}\sum_{t=1}^{N}(2\tilde{z}_{1t}r_t^T(\tilde{H})Q_t^{-1}(\tilde{H}) - r_t^T(\tilde{H})Q_t^{-1}(\tilde{H})\ G_t\ Q_t^{-1}(\tilde{H})r_t(\tilde{H})) \\ &= \frac{1}{\sqrt{N}}\sum_{t=1}^{N} f_t'(\tilde{H})\end{aligned} \tag{7.17}$$

where $G_t = 2(V_{\Delta Z_t 1} - V_{\Delta Z_t 21}\tilde{H})$ and $V_{\Delta Z_t}$ is partitioned as

$$V_{\Delta Z_t} = \begin{bmatrix} V_{\Delta Z_t 1} & V_{\Delta Z_t 12} \\ V_{\Delta Z_t 21} & V_{\Delta Z_t 2} \end{bmatrix}. \tag{7.18}$$

Consequently, the mean deviant introduced in (7.6) can also be calculated from the Hessian matrix of the cost function (7.16). Therefore, the mean deviant is calculated as:

$$\mathcal{M}_N(\tilde{H}) = \frac{1}{N}\frac{\partial^2}{\partial\tilde{H}\partial\tilde{H}^T}\sum_{t=1}^{N} f_t(\tilde{H}) = \frac{1}{N}\sum_{t=1}^{N} f_t''(\tilde{H}), \tag{7.19}$$

where $\frac{\partial^2}{\partial X \partial X^T} f(\tilde{H})$ has been calculated in the Section 5.2.

Remark 7.1: By eliminating the necessity of knowing the switching sequence, the detection of the fault is simplified without the need to identify each sub-model. In other word, the fault can be detected in early stage and there is no need to have exact information about each sub-model individually.

Remark 7.2: The isolation of the fault is also possible without the need to identify each sub-model. Each parameter of the individual sub-model has influence on different SHDP parameters. By isolating the change in the SHDP parameters, one can isolate which parameter in the sub-models has changed.

Remark 7.3: Practically, the true parameters of the model ($\tilde{H}^0$) are not available. Therefore, an estimation of ($\tilde{H}$) from the training data is used. This might result in a bias error in the residual (7.17) [84]. To by pass this bias error, the expected value of the residual from the training data can be subtracted from the residual (7.17) using:

$$k_0 \approx \frac{1}{T}\sum_{t=1}^{T} f_t'(\tilde{H}) \tag{7.20}$$

where T is the number of training samples [81].

Remark 7.4: The faulty mode has to be visited frequently enough so that the detection and the isolation of the fault is possible. This is trivial since occurring a fault in a mode that does not operate would have no impact on the residual.

7.3 The fault detection and isolation

The FDI problem to be solved in this chapter is as follows:

The FDI Problem: By using available inputs and output corrupted with the additive measurement noise for a multi-mode switching model, detect any small change of the parameters in the SHDP representing the multi-mode switching model, and also isolate the parameters of the SHDP that have been changed.

7.3.1 The fault detection

Assume that the parameter vector $\tilde{H}$ is estimated by using a set of training data for normal operation of the system. In addition, assume that $\tilde{H} + \frac{1}{\sqrt{N}}\Upsilon$ represents the parameters of the SHDP model when a small change is occured. Υ is a non-zero vector with the same dimension as $\hat{H}$. The following hypothesis test formulates the change detection (fault detection) problem described above:

$$\begin{aligned} &\text{Normal Operation} \quad \mathbf{H}_0: \ \tilde{H}_c = \tilde{H} \\ &\text{Faulty Operation} \quad \mathbf{H}_a: \ \tilde{H}_c = \tilde{H} + \frac{\Upsilon}{\sqrt{N}} \end{aligned} \tag{7.21}$$

where $\tilde{H}_c$ is the parameter set of the SHDP using the current data. As discussed in the Section 7.2.2, a good residual candidate for local approach is (7.17), which has the following distribution:

$$\begin{aligned} &\zeta_N(\tilde{H}) \backsim \mathcal{N}(0, \Sigma(\tilde{H})) \qquad \text{under } \mathbf{H}_0 \\ &\zeta_N(\tilde{H}) \backsim \mathcal{N}(-\mathcal{M}(\tilde{H})\Upsilon, \Sigma(\tilde{H})) \qquad \text{under } \mathbf{H}_a \end{aligned} \tag{7.22}$$

where $\mathcal{M}(\tilde{H})$ is calculated in (7.19) and $\Sigma(\tilde{H})$ is calculated as:

$$\Sigma(\tilde{H}) \approx \frac{1}{N} \sum_{k=1}^{N} \sum_{j=1}^{N} \mathbf{E}_{\tilde{H}}[f'_k(\tilde{H}) f'^T_j(\tilde{H}))] \tag{7.23}$$

which can be approximated by [85]:

$$\begin{aligned} \Sigma(\tilde{H}) \approx & \frac{1}{N} \sum_{j=1}^{N} f'_j(\tilde{H}) f'^T_j(\tilde{H}) \\ & + \sum_{i=1}^{I} \frac{1}{N-i} \sum_{j=1}^{N-i} \Big(f'_j(\tilde{H}) f'^T_{j+i}(\tilde{H}) + f'_{j+i}(\tilde{H}) f'^T_j(\tilde{H})) \Big) \end{aligned} \tag{7.24}$$

where the value of I should be selected according to the correlation of the signals. One can gradually increase the value of I until the result converges [81].

Equation (7.22) suggests that the detection of small changes in the parameters $\tilde{H}$ is asymptotically equivalent to the detection of changes in the mean of the Gaussian vector. It is shown in [80] that the generalized likelihood ratio test of H_a against H_0 can be written as the following $\mathcal{X}^2$ test:

$$\begin{aligned} \mathcal{X}^2_D = & \big(\zeta^T_N(\tilde{H}) \Sigma^{-1}(\tilde{H}) \mathcal{M}_N(\tilde{H})\big) \\ & \Big(\mathcal{M}^T_N(\tilde{H}) \Sigma^{-1}(\tilde{H}) \mathcal{M}_N(\tilde{H})\Big)^{-1} \\ & \big(\mathcal{M}^T_N(\tilde{H}) \Sigma^{-1}(\tilde{H}) \zeta_N(\tilde{H})\big) \end{aligned} \tag{7.25}$$

where $\mathcal{X}^2_D$ is a central $\mathcal{X}^2$ distribution under normal conditions ($\mathbf{H}_0$), and noncentral $\mathcal{X}^2$ distribution under faulty conditions ($\mathbf{H}_a$). The length of the vector $\zeta_N(\tilde{H})$ (number of the parameters in SHDP) is the degree of freedom of the $\mathcal{X}^2_D$. The $\mathcal{X}^2_\alpha$ is found from the $\mathcal{X}^2$ table and can be used as the threshold

value, where α is the false alarm rate. If the $\mathcal{X}_D^2$ value is greater than $\mathcal{X}_\alpha^2$ the change in the SHDP is detected which can be interpreted as a faulty condition.

7.3.2 The fault isolation

As indicated in the *FDI problem* definition, after a fault is detected it is important to find which parameters of the SHDP have changed. Define Υ_i as a vector such that its i^{th} value is nonzero and the rest elements are zero. The dimension of Υ_i is same as Υ. Therefore, the isolation of the deviated parameters can be described by the following statistical hypotheses:

$$\begin{aligned} &\text{No Change in the } i^{th} \text{ parameter} \qquad \mathbf{H}_0: \ \Upsilon_i = \mathbf{0} \\ &\text{Change in the } i^{th} \text{ parameter} \qquad \mathbf{H}_a: \ \Upsilon_i \neq \mathbf{0} \end{aligned} \tag{7.26}$$

where $i = 1, \cdots, M_n(K)$ which is the length of the parameter vector $\tilde{H}$. Define the $\mathcal{M}_i$ as the i^{th} column of the $\mathcal{M}_N(\tilde{H})$. Also, define $\mathcal{M}_i^c$ as a reduced form of $\mathcal{M}_N(\tilde{H})$ when its i^{th} column is deleted. The minmax test [80] is performed by defining the following matrices:

$$\begin{aligned} F_{ii} &= \mathcal{M}_i^T \Sigma^{-1}(\tilde{H}) \mathcal{M}_i, \qquad & F_{ii^c} &= \mathcal{M}_i^T \Sigma^{-1}(\tilde{H}) \mathcal{M}_i^c \\ F_{i^c i} &= \mathcal{M}_i^{cT} \Sigma^{-1}(\tilde{H}) \mathcal{M}_i, \qquad & F_{i^c i^c} &= \mathcal{M}_i^{cT} \Sigma^{-1}(\tilde{H}) \mathcal{M}_i^c \\ \tilde{\zeta}_i &= \mathcal{M}_i^T \Sigma^{-1}(\tilde{H}) \zeta_N(\tilde{H}), \qquad & \tilde{\zeta}_i^c &= \mathcal{M}_i^{cT} \Sigma^{-1}(\tilde{H}) \zeta_N(\tilde{H}) \end{aligned} \tag{7.27}$$

Also define:

$$\begin{aligned} \tilde{\zeta}_i^* &= \tilde{\zeta}_i - F_{ii^c} F_{i^c i^c}^{-1} \tilde{\zeta}_i^c \\ F_i^* &= F_{ii} - F_{ii^c} F_{i^c i^c}^{-1} F_{i^c i} \end{aligned} \tag{7.28}$$

Then the minmax test is written as:

$$\mathcal{X}_i^{2*} = \tilde{\zeta}_i^{*T} F_i^{*-1} \tilde{\zeta}_i^* \tag{7.29}$$

where $\mathcal{X}_i^{2*}$ has approximately $\mathcal{X}^2$ distribution [85] with a degree of freedom equal to the number of the parameters in SHDP.

Remark 7.5: Due to the approximative nature of $\mathcal{X}_i^{2*}$, if there are a gap between the larger and smaller values, instead of defining a threshold, one can select the most largest values of $\mathcal{X}_i^{2*}$ as indication of the parameters deviation.

In summary, the following FDI algorithm is presented for multi-mode switching models:

Algorithm 4: *The Fault detection and isolation algorithm for multi-mode switching models:*

Initialization:

1. Construct the SHDP using (7.15).
2. Estimate the parameters for SHDP ($\tilde{H}$) using the cost function (7.16) and following the **Algorithm 1**.

Main Algorithm:

1. Calculate the residual $\zeta_N(\tilde{H})$ using (7.17).
2. Calculate the mean deviation $\mathcal{M}_N(\tilde{H})$ using (7.19).
3. Calculate the covariance matrix $\Sigma(\tilde{H})$ using (7.24).
4. Calculate the $\mathcal{X}^2_D$ using (7.25).
5. Define the threshold $\mathcal{X}^2_\alpha$ from the $\mathcal{X}^2$ table with the degree of freedom equal to the number of SHDP parameters and specified false alarm rate, α.
6. If $\mathcal{X}^2_D$ is grater than the threshold, generate an alarm indicating a fault has occured and continue to the next step. Otherwise, no change in parameter vector has been detected.
7. Calculate $\mathcal{X}^{2*}_i$ for all of the parameters in the SHDP parameter vector and select the ones that have largest values with a gap from the lower values.

The assumptions in the local approach introduced in the Section 7.1 is valid if the number of the data points N remains large enough. Therefore, a moving window of sufficient size should be considered for the on-line applications.

7.4 Simulation results

7.4.1 A numerical example

Consider the following three-mode model:

$$\text{Simulation Example 1:} \begin{cases} y_t^0 = -0.9y_{t-1}^0 + u_{t-1}^0 \\ y_t^0 = 0.7y_{t-1}^0 - u_{t-1}^0 \\ y_t^0 = 0.5y_{t-1}^0 - 0.6u_{t-1}^0 \end{cases} \tag{7.30}$$

where the output is considered to have additive white Gaussian noise with normal distribution of zero mean and σ^2 variance, while the inputs are noise free. The input has a uniform distribution of $[-1 \quad 1]$ and the switching sequence is a discrete random number between 1, 2 and 3. The simulation is performed under different noise condition: $\sigma = [0.1 \quad 0.2 \quad 0.3]$. An actuator fault is modeled as 10% increase in the sub-models parameters that are associated with the inputs. There are 9 parameters in the $\tilde{H}$. Thus, the threshold $\mathcal{X}_\alpha^2$ is defined by $\mathcal{X}^2$ value with 9 degree of freedom and false alarm rate of $\alpha = 0.01$ is 21.67.

In order to find the false alarm and the detection rates, 100 independent executions are performed in a normal operation. Additionally, another 100 independent executions are performed under faulty condition. In each execution, the first 500 data points are used for estimating the $\tilde{H}$ nominal value under normal condition. The second 500 data points are used to test the method for calculating the false alarm rate (FAR) in normal operation, as well as the detection rate (FDR) under faulty condition. Table 7.1 summarizes the results:

Table 7.1: The percentage of the false alarm rate (FAR) in 100 independent executions in normal operation and fault detection rate (FDR) in 100 independent executions under faulty condition for the above numerical example

σ	0.1	0.2	0.3
FAR	0	2	2
FDR	100	97	82

The parameters of the SHDP model for this example are:

$$\begin{aligned}\tilde{H} = [&-(\alpha_1+\alpha_2+\alpha_3) \quad -(\beta_1+\beta_2+\beta_3) \quad (\alpha_1\alpha_2+\alpha_1\alpha_3+\alpha_2\alpha_3) \\ &(\alpha_1\beta_2+\alpha_1\beta_3+\alpha_2\beta_1+\alpha_2\beta_3+\alpha_3\beta_1+\alpha_3\beta_2) \quad (\beta_1\beta_2+\beta_1\beta_3+\beta_2\beta_3) \\ &-(\alpha_1\alpha_2\alpha_3) \quad -(\alpha_1\alpha_2\beta_3+\alpha_1\alpha_3\beta_2+\alpha_2\alpha_3\beta_1) \quad -(\beta_1\beta_2\beta_3)]\end{aligned} \tag{7.31}$$

where the i^{th} model is defined as $y_t^0 = \alpha_i y_{t-1}^0 + \beta_i u_{t-1}^0$. It can be seen that the actuator failure does not have any impact on the first, third and sixth parameters. Figure 7.1 shows the box plot of the $\mathcal{X}_i^{2*}$ for the SHDP parameters with $\sigma = 0.1$. As it can be seen from Figure 7.1, there is indeed no influence of

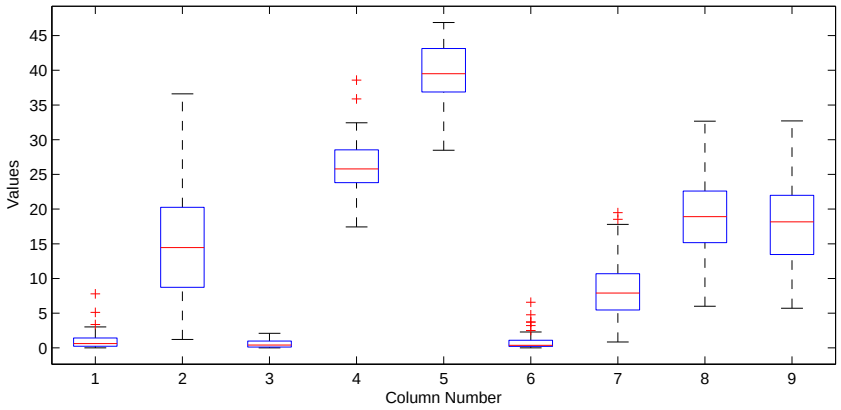

Figure 7.1: The boxplot representation of the $\mathcal{X}_i^{2*}$ values in 100 execution for each parameter

the actuator fault on the first, third and sixth parameters. This can be used to isolate the fault. The boxplot figure represents the statistics of $\mathcal{X}_i^{2*}$ values in hundred executions. The low value of $\mathcal{X}_i^{2*}$ is interpreted as no influence of the fault on that parameter. Figure 7.1 shows that the first, third and sixth parameters of SHDP have lowest values in hundred executions. This means that the simulated fault has no effect on these parameters. By referring to the (7.31), one can notice that the parameters β_i have no effect on the first, third and sixth parameters. Therefore, the result illustrated by Figure 7.1 exactly fulfills the expectations.

7.4.2 The blender process

The blending process defined in the Section 4.4.2 is used to illustrate the practicality of our approach. In this simulation, the first sensor q_1 is assumed to have malfunction which is modeled as 10% additive change in the parameters associated to the first input. It is assumed that all the measurements are corrupted with additive noise as described in the Section 4.4.2.

The switching occures every 30 samples while the inputs q_1 and q_2 are considered to have uniform distribution $[-1 \;\; 1]$. The simulation is executed with: $\sigma_1 = \sigma_2 = \sigma_3 = [0.1 \;\; 0.2 \;\; 0.3]$. In each case, 100 independent executions are performed in normal operation to calculate the false alarm rate (FAR) and another 100 independent executions are performed under faulty condition to calculate the fault detection rate (FDR). In each execution, 500 data points are used to estimate the nominal $\tilde{H}$ value and additional 500 data point are used for the FAR or FDR evaluation. The number of the parameters in $\tilde{H}$ is 5, therefore the corresponding threshold for 1% false alarm is $\mathcal{X}^2_\alpha = 15.09$. Table 7.2 summarizes the results:

Table 7.2: The percentage of the false alarm rate (FAR) in 100 independent executions in normal operation and fault detection rate (FDR) in 100 independent executions under faulty condition for the blending system

σ	0.1	0.2	0.3
FAR	1	4	5
FDR	100	100	86

The SHDP parameters for the blending process is:

$$[(-\alpha_1^0 - \alpha_2^0) \;\; (-\beta_1^0 - \beta_2^0) \;\; (\alpha_1^0\alpha_2^0) \;\; (\alpha_1^0\beta_2^0 + \alpha_2^0\beta_1^0) \;\; (\beta_1^0\beta_2^0)]$$

where $q^0_{3t} = \alpha_i q^0_{1t} + \beta_i q^0_{2t}$. It can be seen that the first input does not have any influence on the second and the fifth parameter. Therefore the fault occurrence should not impact these two. As it is illustrated in figure 7.2, the $\mathcal{X}_i^{2*}$ values for the second and the fifth parameters are the lowest ones.

As discussed in 7.3.2, the proposed method can be used in online applications with moving window approach. This is shown in the figure 7.3. In this simulation, a moving window with length of 200 data points is used for fault

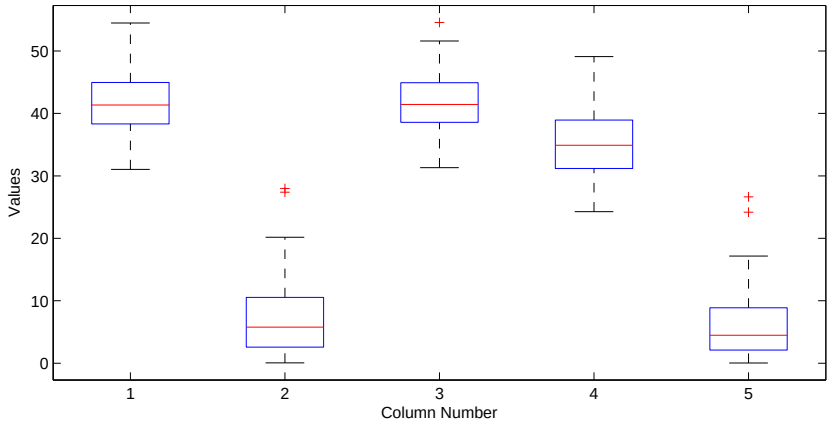

Figure 7.2: The boxplot representation of the $\mathcal{X}_i^{2*}$ values in 100 execution for each parameter

detection. The fault occures at 1000^{th} sample and as it is shown in the figure 7.3 the fault is detected after a delay of 68 samples.

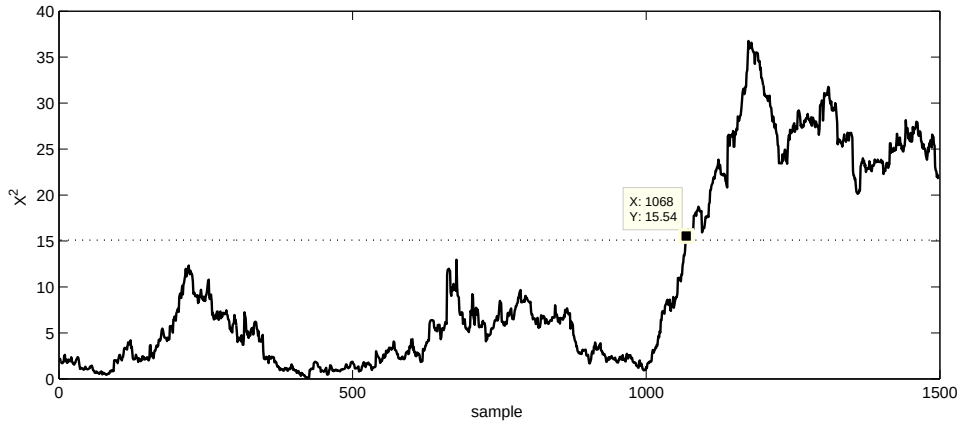

Figure 7.3: The online implementation of fault detection for the blending process

Chapter 8

Conclusion and Future Work

8.1 Thesis conclusion and contributions

The complexity of the parameter estimation problem for Switching ARX (SARX) models is inherited from the discrete switching sequence and corresponding mix-integer optimization. The existing Algebraic Geometric (AG) approach deals with this complexity by providing an alternative problem formulation. The parameter estimation consists of two consecutive steps. The first step estimates the parameters of a linear ARX model, which is constructed through embedding all the sub-models. The second step retrieves the parameters of the sub-models from the estimated model in the first step. Although the AG method delivers exact parameter estimation in the deterministic situations, it suffers from lack of accuracy in the presence of noise.

As discussed in this thesis, constructing a "lifted" model that describes all the sub-models of a SARX model has several advantages. To preserve these advantages when the data points are corrupted with measurement noise, the Stochastic Algebraic Geometric (SAG) approach is developed. The contributions of this thesis can be summarized as follows:

- In order to develop a method to handle the noise properly, first the fundamental problems of the AG approach are investigated. It is illustrated that when noise is involved, both steps of the AG method suffer from inaccuracy:
  - The constructed "lifted" model is mathematically correct under deterministic conditions. However, the noise induced-error inside the "lifted" model grows dramatically. Hence, under noisy conditions,

the parameter estimation of the embedded model suffers from bias and inaccuracy.

- The existing retrieving step requires clustering one data point for each sub-model. Clustering this data point under noisy conditions is prone to error. Also, it is assumed that the prediction error of the chosen data point is zero, which only holds true under deterministic conditions.

- The proposed SAG approach provides an error-in-variable representation of the "lifted" model by reformulating the problem. The developed EIV model represents all the sub-models when the available inputs and output data are corrupted with additive measurement noise. The characteristics of the"lifted" EIV model are closely investigated. It is illustrated that the covariance of the mapped data points error vector varies by the data points. Therefore, to provide a consistent parameter estimation for the proposed model, the element-wise total least square (EW-TLS) method is utilized. The EW-TLS method requires the calculation of covariance matrix of the error vector in each data point. In this thesis, statistical analysis are used to derive general formula for these covariance matrices. The complexity of the parameter estimation in this step is not increased with respect to the number of estimated parameters. The proposed estimation algorithm is non-iterative and efficient.

- In the sub-models' parameters retrieving step, two methods are proposed to improve the retrieving performance in the presence of noise. By proposing these methods, the need to cluster one data point for each sub-model is eliminated. The element-wise method (EW) provides an analytical solution for the calculation of the sub-models' parameters from the "lifted" EIV model, yet is only applicable to two-mode models. The Matrix-Wise approach is applicable to multi-mode models, and results in a nonlinear matrix-wise equation. This equation can be solved by numerical methods, or simply the "fminunc" MATLAB function. In both methods, all the data points are used, which leads to promising robust retrieving results.

- The scalability (by using multi-mode models), practicality (by using practical examples and simulations) and applicability (by using experi-

mental data) of the proposed SAG method are illustrated and evaluated with various simulation experiments. It is demonstrated that the proposed method delivers improved results with respect to performance in comparison with the original AG approach. The improvements measured, can be as high as 50% in some cases. The proposed sub-models' parameters retrieving procedures also show vast improvements, especially in reducing the variance of the estimation error. Due to the use of all the data points in EW and MW approaches, the variance is decreased dramatically.

- The recursive version of the element wise - total least square (EW-TLS) algorithm is derived. This method can estimate the parameters for EIV models with correlated noisy inputs and output, and when the noise profile varies over time. Simulation under different noise scenarios has demonstrated that the proposed REW-TLS method is able to deliver satisfactory online parameters estimation. The convergence of this method depends on the initialization step; however, the use of the Levenberg - Marquardt algorithm has increased the convergency rate. It is shown that the proposed approach is computationally suitable for online applications.

- The recursive version of the proposed SAG method is developed for online applications. The parameters of the "lifted" EIV model are estimated using the proposed recursive EW-TLS approach. The MW and EW approaches are modified to adopt the online conditions. The online versions of these two retrieving approaches does not use any data point, and only use the current estimated parameters of the "lifted" EIV model. The performance of the provided online algorithm is evaluated by simulation results.

- A novel fault detection and isolation algorithm is developed for SARX models. In the proposed algorithm, small changes in the parameters of the sub-models are detected and isolated without the need of estimating them. The developed method applies the local approach, which is a robust residual evaluation method. The performance of the proposed method is investigated via simulation results with practical application potential.

- In this thesis, the noise is defined as input/ouptut additive measurement noise. The results can be generalized to the case when the disturbance is present. However, this generalization needs extra statistical analysis. The effect of disturbance has to be considered in the covariance calculation for SAG method. This calculation is more complex and requires more mathematical calculations, since the disturbance has dynamics.

8.2 Recommendations for future work

- As discussed in this dissertation , a "lifted" EIV (SHDP) model has been developed that captures the dynamics of the SARX sub-models in one embedded model. The characteristics of each model have a direct impact on the parameters of the developed EIV model. The changes in the parameters result in a change of the "lifted" EIV model parameters. Taking advantage of this EIV model, as a representative of the whole SARX model, opens an interesting research topic in adaptive control, fault detection and isolation, and soft sensor design.

  For instance, an adaptive controller with a hierarchy structure can be designed for SARX models. The higher-level controller, designed to control the overall performance and stability, can use the "lifted" EIV model as a representative of the whole SARX model. Then, the lower-level controllers refine the inputs using the local sub-models, so that the desired performance can be achieved. A schematic of this concept is shown in figure 8.1

- A potential application for the EW-TLS method is the estimation of model parameters when the regressors are corrupted with multiplicative noise . The literature is sparse on parameter estimation with multiplicative noises. The use of the conventional least square (LS) and the weighted least square (WLS) to find parameters in linear regression problems in the case of multiplicative noise was investigated in [86–88]. Some researchers have extended the simulation extrapolations method (SIMEX), which was originally used to correct the bias of additive measurement error, to deal with the multiplicative errors [89, 90]. The

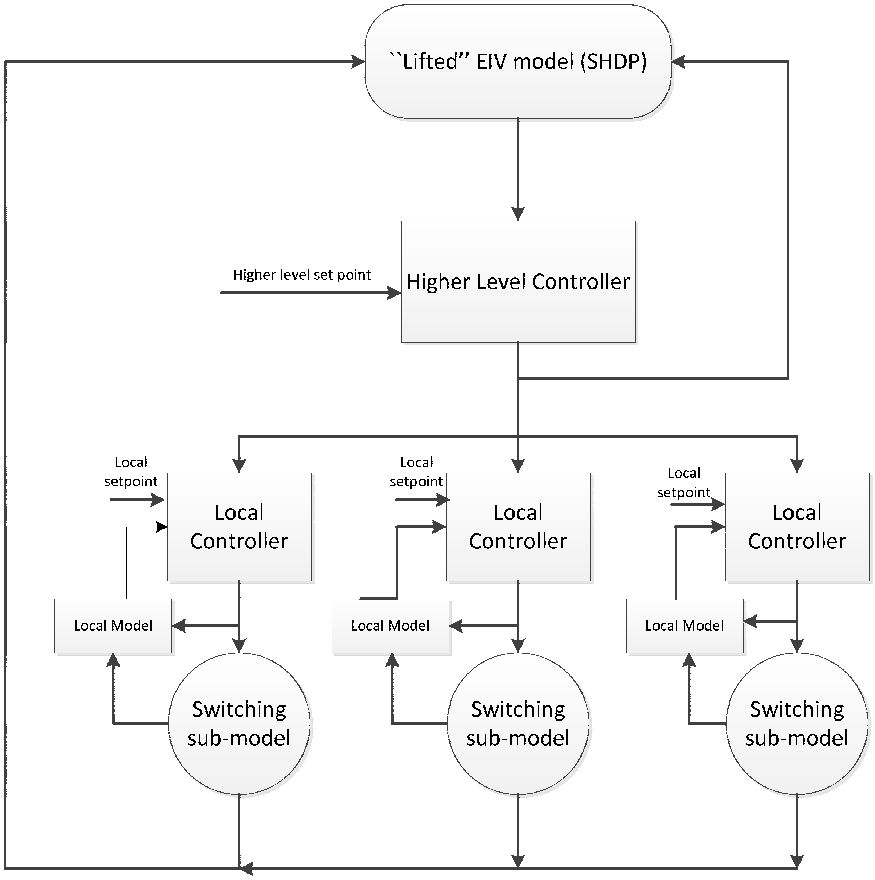

Figure 8.1: The schematic of the hierarchial adaptive controller using "lifted" EIV (SHDP) model

parameter estimation with multiplicative noise is defined as follow:

$$\begin{aligned} &X_1^0 \approx B^0 X_2^0, \qquad X_1 = X_1^0 \odot \nu_1 \quad X_2 = X_2^0 \odot \nu_2 \\ &X_1 \approx B X_2 \end{aligned} \tag{8.1}$$

where ν_1 and ν_2 are vectors of measurement noise with mean equal to one and constant variance, and $\odot$ is an element-wise matrix multiplication. Assume that $\varepsilon_i = 1 - \nu_i$. As shown in this thesis, the variances of $\varepsilon_\mathbf{i}$ and ν_i are equal; therefore the following equalities hold:

$$\begin{aligned} &X_1 = X_1^0 \odot \boldsymbol{\nu_1} \rightarrow X_1 = X_1^0 \odot (1 - \varepsilon_1) = X_1^0 - X_1^0 \odot \varepsilon_1 = X_1^0 + \Delta X_1 \\ &X_2 = X_2^0 \odot \boldsymbol{\nu_2} \rightarrow X_2 = X_2^0 \odot (2 - \varepsilon_2) = X_2^0 - X_2^0 \odot \varepsilon_2 = X_2^0 + \Delta X_2 \\ &X_1 \approx B X_2 \end{aligned}$$

(8.2)

The equation (8.2) is the definition of the EIV parameter estimation with additive error $\Delta X = [\Delta X_1 \quad \Delta X_2]$. The covariance of the error matrix ΔX changes by the data points, and in this thesis, a systematic approach has been presented to find this covariance. Therefore, using the EW-TLS with the above approach is an interesting subject for further research.

- The moving window concept has been applied to the proposed fault detection approach for use in online applications. There is also an opportunity to investigate usage of the proposed recursive SAG algorithm in detecting and isolating faults. In that situation, the algorithm may only use the current data point to detect faults, as opposed to a batch of past data.

Bibliography

[1] C. Heitmeyer and N. Lynch, "The generalized railroad crossing: A case study in formal verification of real-time systems," in *Real-Time Systems Symposium*. IEEE, 1994, pp. 120–131.

[2] Z. Manna and A. Pnueli, *The temporal logic of reactive and concurrent systems: Specification*. springer, 1992, vol. 1.

[3] R. Alur, C. Courcoubetis, T. Henzinger, and P. Ho, "Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems," *Hybrid systems*, pp. 209–229, 1993.

[4] J. Lygeros, C. Tomlin, S. Sastry *et al.*, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349–370, 1999.

[5] J.-R. Abrial, E. Börger, and H. Langmaack, *Formal methods for industrial applications: Specifying and programming the steam boiler control*. Springer, 1996, vol. 1165.

[6] M. S. Branicky, V. S. Borkar, and S. K. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *IEEE Transactions on Automatic Control*, vol. 43, no. 1, pp. 31–45, 1998.

[7] G. Labinaz, M. M. Bayoumi, and K. Rudie, "A survey of modeling and control of hybrid systems," *Annual Reviews in Control*, vol. 21, pp. 79–92, 1997.

[8] W. Heemels, J. M. Schumacher, and S. Weiland, "Linear complementarity systems," *SIAM journal on applied mathematics*, pp. 1234–1269, 2000.

[9] A. J. van der Schaft and J. M. Schumacher, "The complementary-slackness class of hybrid systems," *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 9, no. 3, pp. 266–301, 1996.

[10] J. Shen and J.-S. Pang, "Linear complementarity systems: Zeno states," *SIAM Journal on Control and Optimization*, vol. 44, no. 3, pp. 1040–1066, 2005.

[11] M. Heymann, F. Lin, G. Meyer, and S. Resmerita, "Analysis of zeno behaviors in a class of hybrid systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 376–383, 2005.

[12] B. De Schutter and B. De Moor, "The extended linear complementarity problem and the modeling and analysis of hybrid systems," *Hybrid systems V*, pp. 635–636, 1999.

[13] B. De Schutter, T. Van den Boom *et al.*, "On model predictive control for max-min-plus-scaling discrete event systems," Technical Report bds: 00-04, Control Lab, Fac. ITS, Delft Univ. Techn., Tech. Rep., 2000.

[14] B. De Schutter, "Optimal control of a class of linear hybrid systems with saturation," *SIAM Journal on Control and Optimization*, vol. 39, no. 3, pp. 835–851, 2000.

[15] A. Bemporad, M. Morari *et al.*, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[16] J. Anders, J.-K. Bremer, and W. Mathis, "Mixed-logic dynamical system modeling of $\sigma\delta$-modulators and its application to stability analysis," in *IEEE International Symposium on Circuits and Systems*. IEEE, 2009, pp. 3122–3125.

[17] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat, "An mpc/hybrid system approach to traction control," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 541–552, 2006.

[18] J. Busch, J. Oldenburg, M. Santos, A. Cruse, and W. Marquardt, "Dynamic predictive scheduling of operational strategies for continuous processes using mixed-logic dynamic optimization," *Computers & chemical engineering*, vol. 31, no. 5, pp. 574–587, 2007.

[19] E. Sontag, "Nonlinear regulation: The piecewise linear approach," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 346–358, 1981.

[20] W. P. Heemels, B. De Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085–1091, 2001.

[21] A. Bemporad, G. Ferrari-Trecate, and M. Morari, "Observability and controllability of piecewise affine and hybrid systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 10, pp. 1864–1876, 2000.

[22] S. Weiland, A. L. Juloski, and B. Vet, "On the equivalence of switched affine models and switched arx models," in *45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 2614–2618.

[23] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems: A tutorial," *European Journal of Control*, vol. 13, no. 2-3, pp. 242–260, 2007.

[24] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "A bounded-error approach to piecewise affine system identification," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1567–1580, 2005.

[25] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, "A clustering technique for the identification of piecewise affine systems," *Automatica*, vol. 39, no. 2, pp. 205–217, 2003.

[26] R. Vidal, S. Soatto, and A. Chiuso, "Applications of hybrid system identification in computer vision," in *Proceedings of the European Control Conference, Kos, Greece*, 2007.

[27] A. Juloski, W. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J. Niessen, "Comparison of four procedures for the identification of hybrid systems," *Hybrid systems: computation and control*, pp. 354–369, 2005.

[28] A. Garulli, S. Paoletti, and A. Vicino, "A survey on switched and piecewise affine system identification," in *System Identification*, vol. 16, no. 1, 2012, pp. 344–355.

[29] L. Bako, "Identification of switched linear systems via sparse optimization," *Automatica*, vol. 47, no. 4, pp. 668 – 677, 2011.

[30] I. Maruta, T. Sugie, and T.-H. Kim, "Identification of multiple mode models via distributed particle swarm optimization," in *World Congress*, vol. 18, no. 1, 2011, pp. 7743–7748.

[31] Y. Cheng, Y. Wang, M. Sznaier, N. Ozay, and C. Lagoa, "A convex optimization approach to model (in) validation of switched arx systems with unknown switches," in *51st Annual Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 6284–6290.

[32] Y. Ma and R. Vidal, *Identification of deterministic switched ARX systems via identification of algebraic varieties*. Springer, 2005, pp. 449–465.

[33] A. L. Juloski, S. Weiland, and W. Heemels, "A bayesian approach to identification of hybrid systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1520–1533, 2005.

[34] F. Lauer and G. Bloch, "Switched and piecewise nonlinear hybrid system identification," *Hybrid Systems: Computation and Control*, pp. 330–343, 2008.

[35] J. Roll, A. Bemporad, and L. Ljung, "Identification of piecewise affine systems via mixed-integer programming," *Automatica*, vol. 40, no. 1, pp. 37–50, 2004.

[36] R. Vidal, "Recursive identification of switched arx systems," *Automatica*, vol. 44, no. 9, pp. 2274–2287, 2008.

[37] T. Söderström, "Errors-in-variables methods in system identification," *Automatica*, vol. 43, no. 6, pp. 939–958, 2007.

[38] ——, "System identification for the errors-in-variables problem," *Transactions of the Institute of Measurement and Control*, vol. 34, no. 7, pp. 780–792, 2012.

[39] ——, "A generalized instrumental variable estimation method for errors-in-variables identification problems," *Automatica*, vol. 47, no. 8, pp. 1656–1666, 2011.

[40] I. Markovsky and S. Van Huffel, "Overview of total least-squares methods," *Signal processing*, vol. 87, no. 10, pp. 2283–2302, 2007.

[41] S. Beghelli, R. Guidorzi, and U. Soverini, "The frisch scheme in dynamic system identification," *Automatica*, vol. 26, no. 1, pp. 171–176, 1990.

[42] I. Markovsky, J. C. Willems, S. Van Huffel, B. De Moor, and R. Pintelon, "Application of structured total least squares for system identification and model reduction," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1490–1500, 2005.

[43] K. Usevich and I. Markovsky, "Variable projection for affinely structured low-rank approximation in weighted 2-norm," *arXiv preprint arXiv:1211.3938*, 2012.

[44] I. Markovsky, S. V. Huffel, and A. Kukush, "On the computation of the multivariate structured total least squares estimator," *Numerical linear algebra with applications*, vol. 11, no. 5-6, pp. 591–608, 2004.

[45] S. Van Huffel and J. Vandewalle, *The total least squares problem: computational aspects and analysis*. Society for Industrial Mathematics, 1987, vol. 9.

[46] I. Markovsky, "Bibliography on total least squares and related methods," *Statistics and Its Interface*, vol. 3, pp. 329–334, 2010.

[47] J. H. Manton, R. Mahony, and Y. Hua, "The geometry of weighted low-rank approximations," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 500–514, 2003.

[48] I. Markovsky and S. Van Huffel, "Left vs right representations for solving weighted low-rank approximation problems," *Linear algebra and its applications*, vol. 422, no. 2, pp. 540–552, 2007.

[49] I. Markovsky, M. Luisa Rastello, A. Premoli, A. Kukush, and S. Van Huffel, "The element-wise weighted total least-squares problem," *Computational statistics & data analysis*, vol. 50, no. 1, pp. 181–209, 2006.

[50] M. Schuermans, I. Markovsky, P. D. Wentzell, and S. Van Huffel, "On the equivalence between total least squares and maximum likelihood pca," *Analytica Chimica Acta*, vol. 544, no. 1, pp. 254–267, 2005.

[51] M. Schuermans, I. Markovsky, and S. Van Huffel, "An adapted version of the element-wise weighted total least squares method for applications in

chemometrics," *Chemometrics and intelligent laboratory systems*, vol. 85, no. 1, pp. 40–46, 2007.

[52] A. Premoli and M. L. Rastello, "The parametric quadratic form method for solving tls problems with elementwise weighting," *Total least squares and errors-in-variables modeling: Analysis, algorithms and applications*, pp. 67–76, 2002.

[53] A. Weinmann, *Uncertain models and robust control*. Springer, 1991.

[54] C. B. R. Jagdish K. Patel, *Handbook of the Normal Distribution:Second Edition, Revised and Expanded*. Marcel Dekker Inc., 1996, vol. 150.

[55] J. Harris, *Algebraic geometry: a first course*. Springer, 1992, vol. 133.

[56] R. Vidal, "Identification of pwarx hybrid models with unknown and possibly different orders," in *Proceedings of the American Control Conference*, vol. 1. IEEE, 2004, pp. 547–552.

[57] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[58] Y. Shi and R. Eberhart, *Parameter selection in particle swarm optimization*, 1998, pp. 591–600.

[59] B. Birge, "Psot-a particle swarm optimization toolbox for use with matlab," pp. 182–186, 2003.

[60] J. Niessen, A. Juloski, G. Ferrari-Trecate, and W. Heemels, "Comparison of three procedures for the identification of hybrid systems," in *IEEE International Conference on Control Applications*, vol. 1. IEEE, 2004, pp. 643–648.

[61] X. Jin and B. Huang, "Robust identification of piecewise/switching autoregressive exogenous process," *AIChE Journal*, vol. 56, no. 7, pp. 1829–1844, 2010.

[62] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, and F. J. Doyle III, *Process dynamics and control*. Wiley, 2010.

[63] A. L. Juloski, W. Heemels, and G. Ferrari-Trecate, "Data-based hybrid modelling of the component placement process in pick-and-place machines," *Control Engineering Practice*, vol. 12, no. 10, pp. 1241–1252, 2004.

[64] H.-F. Chen, "Recursive identification for multivariate errors-in-variables systems," *Automatica*, vol. 43, no. 7, pp. 1234–1242, 2007.

[65] J. G. Linden, T. Larkowski, and K. J. Burnham, "An improved recursive frisch scheme identification algorithm," in *International Conference on Systems Engineering*. IEEE, 2008, pp. 65–70.

[66] L. Ljung and T. Söderström, *Theory and practice of recursive identification*, ser. MIT Press series in signal processing, optimization, and control. MIT Press, 1985.

[67] L. Ljung, *System identification: Theory for the user*, 1987, vol. 7632.

[68] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[69] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly Journal of Applied Mathmatics*, vol. II, no. 2, pp. 164–168, 1944.

[70] I. Markovsky, "Software for weighted structured low-rank approximation," *Pre-print*, 2012.

[71] L. Bako, K. Boukharouba, E. Duviella, and S. Lecoeuche, "A recursive identification algorithm for switched linear/affine models," *Nonlinear Analysis: Hybrid Systems*, vol. 5, no. 2, pp. 242–253, 2011.

[72] T. Alizadeh, K. Salahshoor, M. R. Jafari, A. Alizadeh, and M. Gholami, "On-line identification of hybrid systems using an adaptive growing and pruning rbf neural network," in *IEEE Conference on Emerging Technologies and Factory Automation*, 2007, pp. 257–264.

[73] T. Alizadeh, A. Alizadeh, S. Sepasi, and S. Barzegary, "Comparison of two on-line hybrid system identification methods," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 2, 2010.

[74] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 636–653, 2010.

[75] R. Isermann, "Model-based fault-detection and diagnosis–status and applications," *Annual Reviews in control*, vol. 29, no. 1, pp. 71–85, 2005.

[76] S. Narasimhan and G. Biswas, "Model-based diagnosis of hybrid systems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 37, no. 3, pp. 348–361, 2007.

[77] Y. Hu and N. H. El-Farra, "Robust fault detection and monitoring of hybrid process systems with uncertain mode transitions," *AIChE Journal*, vol. 57, no. 10, pp. 2783–2794, 2011.

[78] S. Simani, P. Castaldi, and M. Bonfe, "Hybrid modelbased fault detection of wind turbine sensors," in *Proceedings of IFAC World Congress*, 2011, pp. 7061–7066.

[79] M. Basseville and I. Nikiforov, "Detection of abrupt changes in signals and dynamics systems," 1986.

[80] M. Basseville, "On-Board Component Fault Detection and Isolation Using the Statistical Local Approach," INRIA, Research Report RR-3252, 1997.

[81] B. Huang, "Detection of abrupt changes of total least squares models and application in fault detection," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 2, pp. 357–367, 2001.

[82] J. Li, C. Chan, and H.-Y. Zhang, "Asymptotic local approach in fault detection based on predictive filters," *Journal of guidance, control, and dynamics*, vol. 28, no. 6, pp. 1112–1122, 2005.

[83] J. A. Rice, *Mathematical statistics and data analysis*. Duxbury press, 2007.

[84] Q. Zhang, M. Basseville, and A. Benveniste, "Early warning of slight changes in systems," *Automatica*, vol. 30, no. 1, pp. 95–113, 1994.

[85] ——, "Fault detection and isolation in nonlinear dynamic systems: a combined input-output and local approach," 1996.

[86] P. Xu and S. Shimada, "Least squares parameter estimation in multiplicative noise models," *Communications in Statistics-Simulation and Computation*, vol. 29, no. 1, pp. 83–96, 2000.

[87] J. T. Hwang, "Multiplicative errors-in-variables models with applications to recent data released by the us department of energy," *Journal of the American Statistical Association*, vol. 81, no. 395, pp. 680–688, 1986.

[88] A.-l. Lin, "Estimation of multiplicative measurement-error models and some simulation results," *Economics letters*, vol. 31, no. 1, pp. 13–20, 1989.

[89] S. Lechner, "The multiplicative simulation-extrapolation approach," *Center for Quantitative Methods and Survey Research University of Konstanz Working Paper*, 2007.

[90] E. Biewen, S. Nolte, and M. Rosemann, "Perturbation by multiplicative noise and the simulation extrapolation method," *AStA Advances in Statistical Analysis*, vol. 92, no. 4, pp. 375–389, 2008.