

**MULTI-MODEL SYSTEMS IDENTIFICATION
AND APPLICATION**

BY

AHMED ADEBOWALE ADENIRAN

A Dissertation Presented to the
FACULTY OF THE COLLEGE OF GRADUATE STUDIES
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

SYSTEMS ENGINEERING

MARCH 2015

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **AHMED ADEBOWALE ADENIRAN** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY IN DEPARTMENT OF SYSTEMS ENGINEERING**.

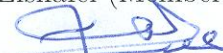
Dissertation Committee



Dr. Sami Elferik (Adviser)



Dr. Moustafa Elshafei (Member)



Dr. Abdul-Wahid Al-Saif (Member)



Dr. Abido, Mohammed Ali Y.
(Member)

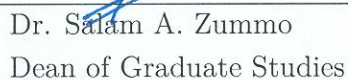


Dr. Deriche, Mohamed (Member)





Dr. Adel Ahmed
For Department Chairman



Dr. Salam A. Zummo
Dean of Graduate Studies

Date

13/4/15



©Ahmed A. Adeniran
2015

To my beloved wife for believing in me and taken good care of the children while in the wilderness of PhD program.

*To my friends, Hameed Abayomi, Taofiq Mursiq, Sikiri Amidu (PhD), who looked after my parents in my absence
-Friends in Need are Friends in Indeed.*

ACKNOWLEDGMENTS

All gratitude belongs to Almighty Allah the lord of the universe, Who teaches man what he knows not. I thank Allah for having spared my life over the years and for bestowing on me His uncompromising blessings, once again. First, I would like to acknowledge the opportunity given to me by King Fahd University of Petroleum and Minerals and especially Department of Systems Engineering to pursue my Doctoral degree in a harmonious environment with state-of the-art facilities.

My immense appreciation and gratitude goes to my advisor Prof. Sami Elferik for his guidance, encouragement and support during this research. Most important of all were his enthusiastic approach towards research and his readiness and eagerness to help me at any time. Additionally his innovative thinking and valuable suggestions greatly inspired me. I thank him, as his is indeed an advisors to reckon with. My thank also goes to my thesis committee members: Professor Elshafei M., Dr. Saif A., Professor Abido M., and Dr. Deriche M, for their supports and suggestions.

I also want to acknowledge the support and suggestions of my friends and colleagues: and many others, especially the Nigerians who in one way or another made my stay at KFUPM and in Saudi Arabia a memorable one.

Special thanks to some of my friends at home: Hammed Abayomi, Dr. Sikiru

Amidu, Taofeek Mursiq, Musliudeen Popoola and others for their encouragement, support and prayers, particularly for taken care of my parents, which have kept my spirit alive while away from home.

Finally, with a deep sense of affection, I offer my sincere thanks to my wife for believing in me and taken care of the children were on the adventure of my PhD program.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	iii
LIST OF TABLES	ix
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xiv
ABSTRACT (ENGLISH)	xvi
ABSTRACT (ARABIC)	xix
CHAPTER 1 INTRODUCTION AND MOTIVATION	1
1.1 Background	1
1.2 Outline of thesis	5
1.3 Thesis Contribution	7
CHAPTER 2 BACKGROUND AND LITERATURE REVIEW	9
2.1 Introduction	10
2.2 General Concept of Multi-model Approach	12
2.3 Multi-model Partition Strategy	19
2.3.1 Model-Based Partition	20
2.3.2 Experimental-Based Partition	23
2.3.3 Data-Based Partition	24
2.4 Multi-model Internal Structure and Parameter Estimation	33

2.5	Validity Computation	36
2.5.1	Gaussian validity	38
2.5.2	Sigmoid validity	39
2.5.3	Residue Approach	39
2.6	Applications	42
2.6.1	Multi-model control	44
2.7	Discussion and Conclusion	46

CHAPTER 3 A MODIFIED COMBINATORIAL PSO BASED MULTI-MODEL IDENTIFICATION OF NONLINEAR SYSTEMS **49**

3.1	Introduction	50
3.2	Problem Formulation	53
3.3	Stage 1: Obtain the number of submodels and the initial partition	56
3.3.1	Particle swarm optimization	57
3.3.2	CPSO based partition	58
3.3.3	Modified CPSO (MCPSO) based partition	60
3.3.4	Estimation of the initial submodels	64
3.4	Stage 2: Obtain the final submodels	64
3.5	Simulation Examples	67
3.5.1	Example 1	70
3.5.2	Example 2	73
3.5.3	Example 3	76
3.5.4	Example 4	80
3.5.5	Example 5	84
3.6	Conclusion	88

CHAPTER 4 CONSTRAINED KALMAN FILTER FOR VALIDITY ESTIMATION **90**

4.1	Introduction	91
4.2	Problem Formulation	93

4.3	Constrained Kalman Filter (CKF) for Validity Computation . . .	94
4.3.1	Constrained Kalman Filter (CKF)	94
4.3.2	Constrained Kalman Filter (CKF) for Validity Estimation of local models	97
4.3.3	Estimating Q and R in CKF algorithm	101
4.4	Simulation Examples	104
4.4.1	Case 1: Suitability of CKF Algorithm as Validity Compu- tation	105
4.4.2	Case 2: Multi-model Identification	109
4.5	Conclusion	122

**CHAPTER 5 OPTIMIZATION OF THE NUMBER OF SUB-
MODELS' PARAMETERS 123**

5.1	Introduction	123
5.2	Problem Formulation	125
5.3	Obtain the number of submodels, number of parameters of sub- models and the initial partition	127
5.3.1	Obtain the initial partition using MCPSO	128
5.3.2	Estimation of the initial submodels	131
5.4	Simulation Examples	133
5.4.1	Example 1	134
5.4.2	Example 2	137
5.5	Conclusion	141

**CHAPTER 6 APPLICATION IN CONTROL AND FAULT DI-
AGNOSTICS 142**

6.1	Multi-model control of nonlinear systems	142
6.1.1	Weighted One-Step-Ahead Controller	145
6.1.2	Fusion of Weighted One-Step-Ahead Controllers	146
6.1.3	Fusion of Model Parameters for Weighted One-Step-Ahead Controller	148

6.1.4	Simulation Examples	149
6.2	Fault detection and Isolation	163
6.3	Conclusion	172
CHAPTER 7 CONCLUSION AND RECOMMENDATION		175
7.1	Main Contribution	175
7.2	Future Recommendation	178
REFERENCES		181
VITAE		213

LIST OF TABLES

2.1	Some modification to LOLIMOT algorithm	26
2.2	Recent Clustering techniques for multi-model partition	32
2.3	Recently used Validity Computation	43
3.1	Stage 2: estimating final submodels	68
3.2	MCPSO parameter settings	70
3.3	Results of Stage 1(initial submodels)	72
3.4	Results of Stage 2 (final submodels)	72
3.5	Validation performance test on common validity estimations . . .	73
3.6	Results of Stage 1 (initial submodels)	75
3.7	Results of Stage 2 (final submodels)	76
3.8	Validation performance test on common validity estimations . . .	77
3.9	Results of Stage 1 (initial submodels)	79
3.10	Results of Stage 2 (final submodels)	79
3.11	Validation performance on common validity estimations	80
3.12	Results of Stage 1 (initial submodels)	82
3.13	Results of Stage 2 (final submodels)	83
3.14	Validation performance on common validity estimations	84
3.15	Results of Stage 1 (initial submodels)	87
3.16	Results of Stage 2 (final submodels)	87
3.17	Validation performance test on common validity estimations . . .	89
4.1	Validity estimation of submodels	102

4.2	Performance Measures Comparison of Different Validity Computations	112
4.3	Performance Measures Comparison of Different Validity Computations	114
4.4	Performance Measures Comparison of Different Validity Computations	116
4.5	Validation performance on common validity estimations	118
4.6	Performance Measures Comparison of Different Validity Computations	120
5.1	MCPSO parameter settings	134
5.2	Results of Stage 1 (initial submodels)	135
5.3	Results of Stage 2 (final submodels)	135
5.4	Validation performance for system 1	136
5.5	Results of Stage 1 (initial submodels)	138
5.6	Results of Stage 2 (final submodels)	139
5.7	Validation performance on common validity estimations	140
6.1	Controllers' parameters (λ)	151
6.2	Performance of controllers	152
6.3	Performance of controllers for optimized R and Q in CKF	152
6.4	Controllers' parameters (λ)	158
6.5	Performance of controllers	158
6.6	Performance of controllers for optimized R and Q in CKF	158

LIST OF FIGURES

2.1	Scope of review of Multi-model framework	13
2.2	Operating space partition	15
2.3	Multi-model identification	18
2.4	Axis-orthogonal and oblique partition for two dimensional input space (u_1, u_2)	29
2.5	Neural Network and Fuzzy logic Validity computation	43
2.6	Fused controller multi-model control	45
2.7	Fused parameters multi-model control	46
3.1	output blended multi-model identification structure	55
3.2	output blended multi-model identification structure	71
3.3	MCPSO objective function convergence plot	73
3.4	Multi-model identification outputs using validation data	74
3.5	Multi-model identification outputs using validation data	75
3.6	MCPSO objective function convergence plot	76
3.7	Multi-model identification outputs using validation data	77
3.8	Multi-model identification outputs using validation data	78
3.9	MCPSO objective function convergence plot	80
3.10	Outputs of multi-model identification using validation data	81
3.11	Outputs of multi-model identification using validation data	82
3.12	MCPSO objective function convergence plot	83
3.13	Outputs of multi-model identification using validation data	85
3.14	Outputs of multi-model identification using validation data	86

3.15	MCPSO objective function convergence plot for 4-paramaters structure	88
3.16	Outputs of multi-model identification using validation data	89
4.1	Actuator Output and segmented outputs	107
4.2	Estimated combined output and validity profiles	108
4.3	System Output	109
4.4	Estimated combined output and validity profiles	110
4.5	CKF based validity computation for output blended multi-model framework	110
4.6	Multi-model identification outputs using validation data	112
4.7	Multi-model identification error using validation data	113
4.8	Multi-model identification outputs using validation data	114
4.9	Multi-model identification error using validation data	115
4.10	Outputs of multi-model identification using validation data	117
4.11	Multi-model identification error using validation data	118
4.12	Outputs of multi-model identification using validation data	119
4.13	Multi-model identification error using validation data	120
4.14	Outputs of multi-model identification using validation data	121
5.1	MCPSO objective function convergence plot	136
5.2	Outputs of multi-model identification using validation data for sys- tem 1	137
5.3	Multi-model identification error using validation data for system 1	138
5.4	MCPSO objective function convergence plot	139
5.5	Outputs of multi-model identification using validation data	140
5.6	Multi-model identification error using validation data	141
6.1	Fused controller multi-model control configuration	147
6.2	Fused parameters multi-model control configuration	149
6.3	Control of time varying system using Type A controller	153
6.4	Control of time varying system using Type B controller	154

6.5	Control inputs for Type A controller	155
6.6	Control inputs for Type B controller	156
6.7	Control of time varying system using Type A controller	159
6.8	Control of time varying system using Type B controller	160
6.9	Control inputs for Type A controller	161
6.10	Control inputs for Type B controller	162
6.11	Multi-model Fault Diagnosis	163
6.12	The Three-Tanks System	165
6.13	Multi-model Fault Detection of Three-Tanks System	166
6.14	Validity profiles for normal mode	167
6.15	Validity profiles for fault on valve v_2	168
6.16	Validity profiles for fault on valve v_2 and v_3 at different time . . .	169
6.17	Validity profiles for fault on valves v_2 and v_3 simultaneously . . .	170
6.18	Validity profiles for fault on valve v_2 with 5% scaled noise	172
6.19	Validity profiles for fault on valve v_2 with 10% scaled noise	173
6.20	Validity profiles for fault on valve v_2 with 20% scaled noise	174

LIST OF ABBREVIATIONS

CSTR	Continuous Stirred Tank Reactor
GP	Gaussian Process
MMF	Multi-model Framework
PSO	Particle Swarm optimization
CPSO	Combinatorial Particle Swarm optimization
MCPSO	Modified Combinatorial Particle Swarm optimization
CKF	Constrained Kalman Filter
CKFP	Constrained Kalman Filter settings $W = P^{-1}$
CKFW	Constrained Kalman Filter settings $W = I$
MSE	Mean Square Error
PMF	Percentage Model Fitness
VAF	Variance-Accounted-For
PWC	Piecewise Continuous Model
PWA	Piecewise Affine
LPV	Linear Parameter Varying
LMN	Local Model Network
LOLIMOT	Local Linear Model Tree
LLNF	Local Linear Neuro-Fuzzy
NARX	Nonlinear Autoregressive with Exogenous inputs
ARX	Autoregressive with Exogenous inputs
T-S	Takagi-Sugeno

OPL	Operating Point Linearization
SNF	Sector Nonlinearity Transformation
VBL	Velocity Based Linearization

THESIS ABSTRACT

NAME: Ahmed Adebowale Adeniran
TITLE OF STUDY: Multi-Model Systems Identification and Application
MAJOR FIELD: Department of Systems Engineering
DATE OF DEGREE: 2015

Modeling of dynamical systems is an important task that cut across many disciplines. Model has been found to be indispensable for rapid development of new systems, analyses of existing systems, simulation of process monitoring, prediction, fault detection and design of process control. Modeling of real life industrial systems is however not a trivial task due their inherent nonlinearities with wide operating ranges and large set point changes. In recent years, much attention has been given to multi-model-based alternative approach to describe nonlinear systems. In contrast to conventional modeling technique, a system is represented by a set of models, that are combined, with different degree of validity, to form the global model. Each model represents the system in a specific region of operation. This thesis concerns with the multi-model identification of nonlinear systems and its applications. Some of the key challenges encounter in representing a nonlinear

system with interpolated multiple models is addressed. One of the challenges of multi-model approach is the partitioning of the system's operating space to a number of sub-spaces. This translate to finding the submodels that can adequately represent the entire operating region of the nonlinear system when combined within the multi-model framework. We presented a heuristic and meta-heuristic data based partition methods for multi-model identification of nonlinear systems. In the proposed approach the structure and the number of submodels are not known a priori. The proposed method consists of two stages. The first stage deals with initial estimate of the number of submodels and their parameters while the final submodels are obtained in the second stage. Another issue of important is finding the weight contribution of the submodels for combining them to completely form multi-model representation of the nonlinear system. In this study, a constrained Kalman filter (CKF) validity computation is developed for interpolation of submodels. The presented method overcomes some of the drawback of commonly used validity computations such as sensitivity to parameter selection, and restriction to partition strategy. The proposed CKF showed good performance and better than some commonly used validity computation. Finally, two important application areas namely, control and fault diagnosis, are investigated on the proposed multi-model framework. In the first case, multi-model weighted one-step ahead reference tracking control algorithms are designed for some of the identified systems. In the second case, the suitability of the CKF algorithm under multi-model framework is tested for fault detection and isolation . In all cases simulated nonlinear systems

examples that had been studied previously in the literature are provided to illustrate the improved performance of the proposed methods.

ملخص الرسالة

الاسم الكامل: احمد اديبولى ادينيران

عنوان الرسالة: : تطبيقات وتعريف للأنظمة بواسطة طريقة النماذج المتعددة

التخصص: قسم هندسة النظم

تاريخ الدرجة العلمية: ٢٠١٥

تعد نمذجة الأنظمة الديناميكية خطوة مهمة لإنجاز العديد المجالات. النموذج الرياضي لا غني عنه لمواكبة التطوير المتسارع في الأنظمة، وتحليل ودراسة الأنظمة الحالية، ومحاكاة ادارة العمليات، والتوقع المستقبلي لمخرجات الأنظمة والاعطال المستقبلية وتصميم المتحكم لهذه العمليات. على الرغم من ذلك فإن نمذجة للعمليات الصناعية على أرض الواقع ليست مهمة سهلة، ويرجع ذلك الى الطبيعة الاخطية للأنظمة بالإضافة للمجال الواسع للمتغيرات والقيم التي تحدد عمل الأنظمة. في السنوات الاخيرة، كثر الأهتمام بتمثيل الأنظمة الاخطية بواسطة طريقة النماذج المتعددة. والتي تختلف عن الطريقة التقليدية بأن الأنظمة تمثل بمجموعة من النماذج لمختلف حالات النظام بحيث تكون مجتمعة تمثل النموذج الكلي للنظام. كل نموذج من هذه النماذج يمثل النظام في مجال عمل صغير ومحدد. هذه الاطروحة تتناول طريقة تعريف الأنظمة وتمثيلها بواسطة النماذج المتعددة للأنظمة اللاخطية وتطبيقاتها. وتتناول بعض التحديات الرئيسية التي تواجه تمثيل الأنظمة اللاخطية عند التشابك بين هذه النماذج المتعددة. يعد تقسيم مجال عمل النظام الى عدد من المجالات الصغيرة المنبثقة عنه من التحديات التي تواجه طريقة النمذجة بواسطة النماذج المتعددة. وهذه الخطوه تتوج بالحصول على نماذج فرعية التي تكفي لتمثيل كامل مجال عمل النظام الاخطي عندما تجتمع معا. نعرض هنا تعريف الأنظمة اللاخطية بواسطة طريقة تجريبية أرشادية وطريقه فوق تجريبية أرشادية. في هذا الطرح نفترض أن هيكلية وعدد النماذج الفرعية معلوم من البداية. وهذا الطرح يتكون من مرحلتين. المرحلة الأولى تتناول التقدير الأولى لعدد النماذج الفرعية ومعطياتها بينما النماذج الفرعية النهائية نحصل عليها من المرحلة الثانية. أمر مهم آخر هو ايجاد نسب الاهمية للنماذج الفرعية لدمجهم معا في نموذج متعدد واحد للنظام فلتر محدود. هذه Kalman اللاخطي. في هذه الدراسة تم تطوير طريقة التحقق لعملية ربط الأنظمة الفرعية ب الطريقة تحل المشاكل المصاحبة لعملية التحقق المعتادة مثل الحساسية للاختيار القيم، والقيود في استراتيجية

المقترحة أداء جيدا أفضل من بعض الطرق التقليدية في حسابات التحقق. أخيرا، CKF التقسيم. أظهرت طريقة تمت دراسة مجالين مهمين باستخدام الانظمة المتعددة وهما التحكم وتحديد الاخطاء في النظام. في الحالة الأولى يتم تغيير قيم نظام النماذج المتعدده خطوة واحده الى الأمام بناء على اشارة من المتحكم المصمم و بناء على النماذج مع نظام النماذج المتعددة لتحديد الأخطاء وعزلها. وفي CKF المعرفة. في الحالة الثانية، تم التأكد من الاستدامة جميع الحالات تم عمل محاكاة تشابه المحاكاة المعمول بها في البحوث السابقة لهذه الدراسة لبيان التحسن والتقدم الذي حصل باستخدام الطرح المقدم في هذه الرسالة

CHAPTER 1

INTRODUCTION AND MOTIVATION

1.1 Background

Modeling of dynamical systems is an important task that cuts across many disciplines. Model has been found to be indispensable for rapid development of new systems, analyses of existing systems, simulation of process monitoring, prediction, fault detection and design of process control. Modeling of real life systems is however not a trivial task due to their complexity and inherent nonlinearities. First-principle models and their analytical approximations for such systems can be very difficult to derive, because they require detailed expert knowledge, which may be lacking. The resulting models are often very complex, labor-intensive, time consuming and hence expensive. Moreover, these models are not always very accurate, because it is difficult to decide which parameters are relevant and

must be included in the model. Therefore, system identification which is a more flexible type of modeling has been embraced.

In system identification, the aim is to estimate mathematical models of a dynamical system directly from observed input and output data with little expert knowledge. System identification often yields compact representative models that are accurate enough to be suitable for optimization, fault detection and model-based control, which has found widespread use in process industry. Over the past years, great number of efforts has been devoted to modeling and identification. Generally, five steps are carried out in system identification process. The first step involves selection of type of model considered suitable for the application at hand. The second step is the design of input perturbation that will influence the ability of the model to capture important system's behavior. Next is to carry out identification experiments to obtain input and output measurements. Then, the parameters of the model selected are estimated from the collected input and output measurements. Finally, the validation of the obtained model is done to ensure correct description of the system.

As noted earlier, the selection of the type of model to represents the system is crucial in system identification process. This decision is usually based on knowledge of the system under study. Although many real-life phenomena have inherent nonlinearities, in practice, the linear time-invariant model is usually used to approximate the behavior of nonlinear systems disregarding a possible nonlinearities. The use of linear model is attractive because they are easy to understand

and interpret. Building linear models usually requires significantly less effort than the estimation of nonlinear models. In addition, the research community provides well established collection of tools suitable for their analysis, monitoring, optimization, identification, and control. Several attractive identification methods for linear system can be found in [1].

Despite these attractive features in linear model, unfortunately their approximations of nonlinear systems are only valid within a small range of input and output. Hence, there has been a considerable efforts towards the development of efficient identification methods for nonlinear systems.

Contrary to linear system identification that may not be adequate for complex nonlinear systems, there exists several nonlinear systems structures in the literature. Some proposed modeling and identification structures including Block-oriented (e.g Wiener and Hammerstein) [2, 3, 4, 5], Volterra [6], and polynomials NARX [7, 8], black-box models, for instance, support vector machine [9], wavelet [10], and neural networks [11, 12, 13] have been proposed. Recent comprehensive survey on these techniques can found in [14]. While these models have proven to be successful in different scenarios, they still suffer from certain limitations. The drawback of these models is that they are complex, and difficult to estimate and analyze. Therefore, other model structures have received considerable attention over the years.

In recent times, interest in multi-model approach has risen as alternative to conventional modeling and identification of complex nonlinear systems, to over-

come the difficulties encountered as mentioned previously. This technique is seen as an effective way of system modeling that relies on problem decomposition strategy. It involves combining a set of models, called local models, with different degree of validity to form the system output. Each local model represents a specific region of operation of the global system model [15, 16]. In control application, it is easier to design a local controller for each of the simplified local model instead of a global model. The past years have witnessed lot of contributions in this effective area of modeling and identification, and has gained lots of momentum in many fields like biochemical [17], process [18], communication [19], power [20], etc. Multi-model approach is appealing owing to its simplicity and transparency. The approach is mathematically tractable and allow direct incorporation of qualitative plant knowledge [21]. In control application, well matured linear model and control analysis can be exploited when the local models are selected linear.

The dissertation is concerned with multi-model identification of nonlinear systems, where complex nonlinear systems can be identified using several models, that can be combined in a particular form such that each model contributes to the system output according to a certain degree of validity. This doctoral thesis has three aims:

- Design of partition strategy for nonlinear system identification in multi-model framework
- Design of validity computation for interpolation of multiple models in multi-model framework

- Apply Multi-model in control and fault detection.

1.2 Outline of thesis

Chapter 1 introduces the problem to be studied in this thesis in a comprehensive manner, by motivating the reader about the significance of modeling and identification of dynamic systems and nonlinearity in real industrial systems.

Chapter 2 presents comprehensive background and detailed literature review to provide insights into avenues, where research is still lacking and where this work can help bridging the gaps in providing a novel methodology for identifying nonlinear systems based on multi-model framework.

Multi-model design involves three steps. The first step involves partitioning of the operating system into smaller regions. In the second step, both the structure and the parameters of the submodel associated with each subregion are determined. Finally, the local models are combined together using weighting function that defines the contribution of each submodel to the representation of real system. These steps might be dependent on one another depending on the adopted approach in the partitioning stage. The first and the second step are discussed in chapter 3 while the third is handled in chapter 4.

Chapter 3 presents new algorithms for the effective partitioning of nonlinear systems from data without prior knowledge of the operating conditions. The algorithm handles the optimization of both the number of submodels and the effective number of parameters needed to identify the original nonlinear system

in the multi-model framework.

In chapter 4, new algorithm for computing the validities of submodels is proposed. We developed constrained Kalman filter (CKF) algorithm to compute the weight needed for interpolating the submodels obtained through the algorithm proposed in the chapter 3. Hence, this chapter completes the steps necessary for the development of multi-model framework. To improve robustness and performance of the algorithm, a methodology on how to select the parameters of the algorithm is suggested. In addition, comparative study is carried out with other commonly used validity computation to highlight its performance.

Chapter 5 extends the partition algorithm proposed in chapter 3 to handle a situation where the number of parameters for each submodels is unknown.

Chapter 6 presents multi-model control and fault diagnosis problems for non-linear systems. Two multi-model controller designs based on weighted one-step ahead controller are developed for reference tracking and investigated using the methods and algorithms developed in the preceding chapters. Furthermore, the CKF algorithm is analyzed for fault detection and isolation of a three tank non-linear system.

Finally, the thesis is concluded in Chapter 7, summarizing the contributions of this work along with future research directions.

1.3 Thesis Contribution

The major concern of this thesis is the use of multi-model framework for identification of complex systems. In addition, this dissertation also consider its application in some areas of interest such as nonlinear system control and fault diagnosis. In this context, the contributions of our work are as follows:

- Optimize the number of partitions (submodels) needed for effective representation of nonlinear systems in multi-model framework.
- Determine the structure and estimate the parameters of the submodels without prior knowledge.
- Develop a method to compute the validity of submodels for effective interpolation.
- Design multi-model control strategy for nonlinear system
- Apply the proposed validity computation for fault diagnosis.

The contributions of this research resulted in the following peer-reviewed works:

1. Ahmed A. Adeniran, Sami El Ferik, "Modeling and Identification of Non-linear Systems: A Review of the Multi-Model Approach" SUBMITTED TO IEEE Transactions on Systems, Man and Cybernetics: Systems (ISI, IF:2.169)

2. Ahmed A. Adeniran and Sami Elferik, "Validity Estimation For Multi-Model Identification Using Constrained Kalman Filter", IASTED on Modeling Identification and Control (MIC) 2014
3. Sami El Ferik, Ahmed A. Adeniran, "Constrained Kalman Filter as Validity for Multi-Model Identification and Fault Diagnosis of Nonlinear Systems" , Submitted to Journal of the Franklin Institute (ISI, IF:2.260)
4. Ahmed A. Adeniran, Sami El Ferik, "A Modified Combinatorial PSO based Multi-model Identification of Nonlinear Systems", Submitted to Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing (ISI, IF:0.553)
5. Ahmed A. Adeniran, Sami El Ferik, "One step Ahead controller design for nonlinear systems using multi-model approach " , (Under Preparation)

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

The efficacy of the multi-model framework in modeling and identification of complex, nonlinear and uncertain systems has been widely recognized in the literature owing to its simplicity, transparency and mathematical tractability, allowing the use of well known modeling analysis and control design techniques. The approach proved to be effective in addressing some of the shortcomings of other modeling techniques such as those based on a single NARX model or neural networks. Great number of researchers have contributed to this active field. In order to provide background for subsequent chapters, this chapter attempt to provides a comprehensive coverage of the multi-model approach for modeling and identification of complex systems. The study contains a classification of different methods, the challenges encountered, as well as recent applications of multi-model framework in various fields. In the literature survey, our main focus is on the multi-model frame-

work where the final system's representation and behavior is generated through the interpolation of several possible local models. This is of prime importance to control designers. All through this chapter, background to multimodel representation of complex systems, different active research areas and open problems are discussed.

2.1 Introduction

In recent years great efforts have been devoted to modeling and identification of nonlinear systems due to inherent nonlinearity in real life industrial plants and processes. For such systems assumption of linearity fails and accurate mathematical model is infeasible as a result of large number of parameters and lack of knowledge of some parameters [22]. In many real applications, approximate linear models have been adopted even though they may not adequately represent the real system and its nonlinearities. Consequently, a great effort has been devoted to the modeling, identification and analysis of nonlinear systems in the literature.

Several nonlinear model structures and methods such as Block-oriented [2, 3, 4, 5], Volterra [6], and polynomials NARX [7, 8] models have been proposed. While these models have proven to be successful in different scenarios, they still suffer from certain limitations. Recent comprehensive survey on these techniques can found in [14]. Alternatively, black-box models, for instance, support vector machine [9], wavelet [10], and neural networks [11, 12, 13] have been proposed. This type of modeling approach mainly lacks transparency and also experiences

course of dimensionality [23]. In addition, utilizing such models for control design might even pose problems and in some cases may be impossible to implement.

Multi-model framework (e.g., [15, 24, 25]) is another approach towards modeling and identification of complex nonlinear systems. It relies upon problem decomposition strategy. In this approach, a global system model is formed by a set of local models integrated with different degree of validity. Each local model represents the dynamic of the system in a specific region of the operating space [16]. Although the multi-model approach has been criticized for creating sub-optimal and input dependent models [26, pp.29], it has continued to attract many researchers because of its potential benefits. The approach is simple, mathematically tractable, and like other techniques, it allows direct incorporation of qualitative plant knowledge [16]. Most importantly, well matured linear model and control analysis can be exploited when the local models are assumed to be linear [21].

The past years have witnessed lot of contributions in this area of modeling and identification, and has gained a great deal of interest in many fields, both in academia and industries. The idea of multi-model has been presented in different context including, regime based multi-model [27], local model networks [28], local radial basis function network [29], Takagi-Sugeno (T-S) fuzzy local model [30], piecewise continuous system [31], etc. Since modeling is an important task in many areas, multi-model approach has found its way into process optimization [32] control [33, 34], fault detection and isolation [35, 36, 37, 38], as well as prediction

[39]. Some of the early developments in multi-model approach can be found in [40]. Also, the analysis of differences and similarities between radial basis function networks, T-S fuzzy models and local model networks can be found in [41].

This chapter presents analysis and review of recent developments to the design challenges encountered in multi-model framework for modeling and identification of nonlinear system. We shall focus on the multi-model framework where the final system's representation and behavior is generated through the interpolation of several possible submodels (or local models). As shown in Fig. 2.1, the scope includes different strategies of operating space partition, submodel structures and validity computations of multi-model, challenges and recent developments in solving them. This chapter provides a basis for the contributions in this thesis.

The rest of this chapter is organized as follows: Section 2.2 describes the general multi-model concept. Recent partition strategies for interpolated multi-model framework (MMF) are discussed in section 2.3. In section 2.4, we discuss the internal structure of MMF and parameter estimation techniques. Validity computations is discussed in section 2.5. In section 2.6, applications of multi-model in various areas are highlighted. Finally, a brief discussion and conclusion are given in section 2.7.

2.2 General Concept of Multi-model Approach

Multi-model framework (MMF) employs a strategy that partitions the entire operating space of the system into a number of operating regions (see for example

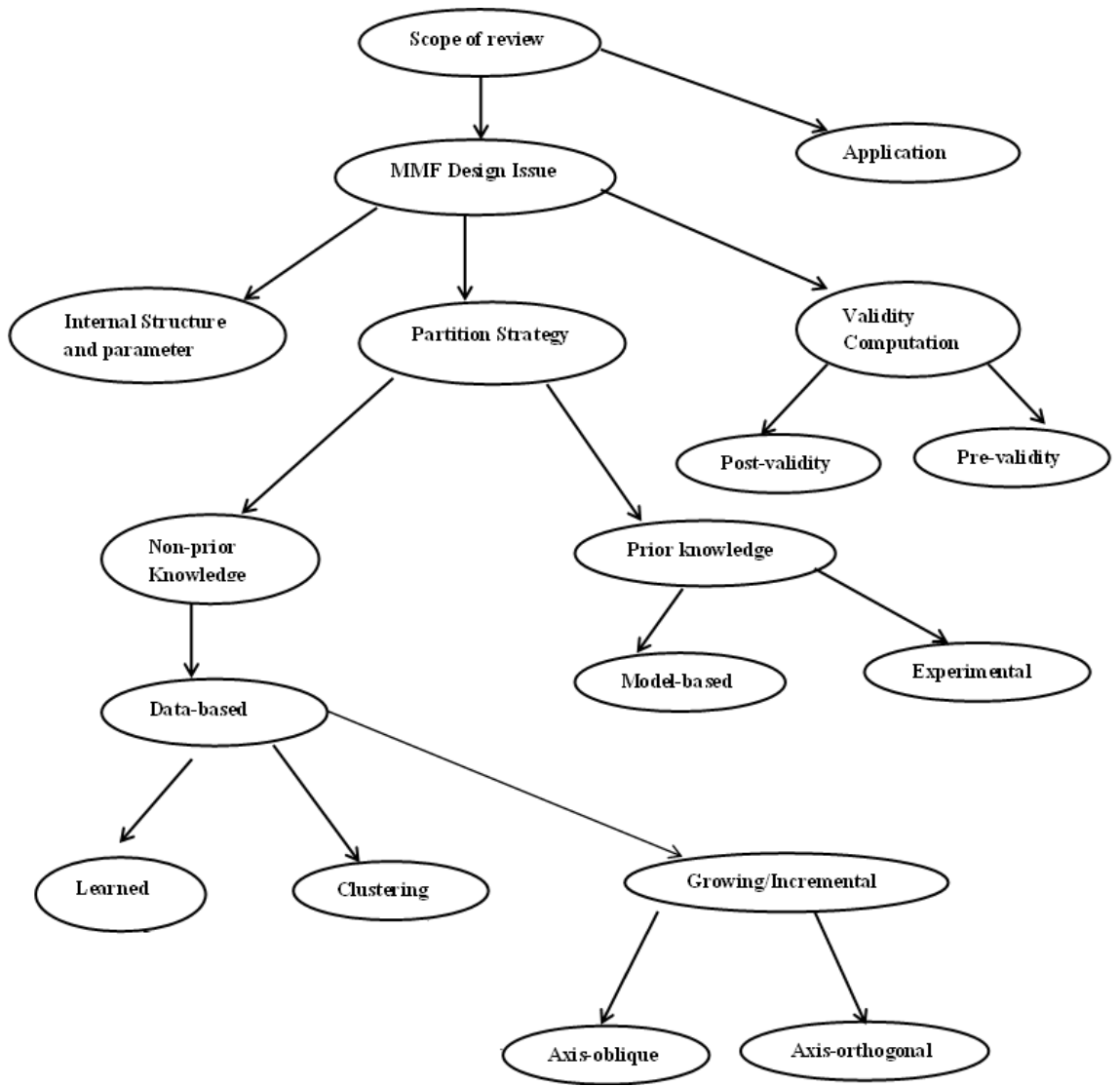


Figure 2.1: Scope of review of Multi-model framework

Fig. 2.2), where each region is associated with a submodel (local model) that depicts the behavior of the system in that specific region. In interpolated MMF, however, a weighted combination of these local models are blended to form the global output of the system. For example, suppose there is a nonlinear system of the form:

$$y(k) = F(y(k-1), y(k-2), \dots, y(k-n), u(k-1), u(k-2), \dots, u(k-m)) \quad (2.1)$$

where $u(k) \in \mathfrak{R}^m$ is the input and $y(k) \in \mathfrak{R}^n$ is the output of the system. The integer m and n are the time lag of the input and output respectively, the multi-model formulation of the system can be represented as

$$y(k) = \sum_{i=1}^M \mathcal{M}_i(k) \phi_i(z(k)) \quad (2.2)$$

where $\mathcal{M}_i(k) = f_i(\varphi_i(k), \Theta_i)$ with $\varphi(\cdot)$ and Θ as the regression and parameter vectors respectively. M are the i^{th} submodel (local model) and number of submodels, respectively. $\phi_i(z(k))$ is the i^{th} weight or validity function. The variable $z(k)$, called scheduling variable, is a subset of the information space, $\varphi(k)$ (regression vector), that defines the operating space of the system. The validity function, $\phi_i(\cdot)$, represents the contribution of each submodel in the composition of $y(k)$. Such function allows smooth dynamic transition in the dynamics when moving from one region to another in the operating space. In order to provide right interpretation of the validity function, it is desirable that the contribution

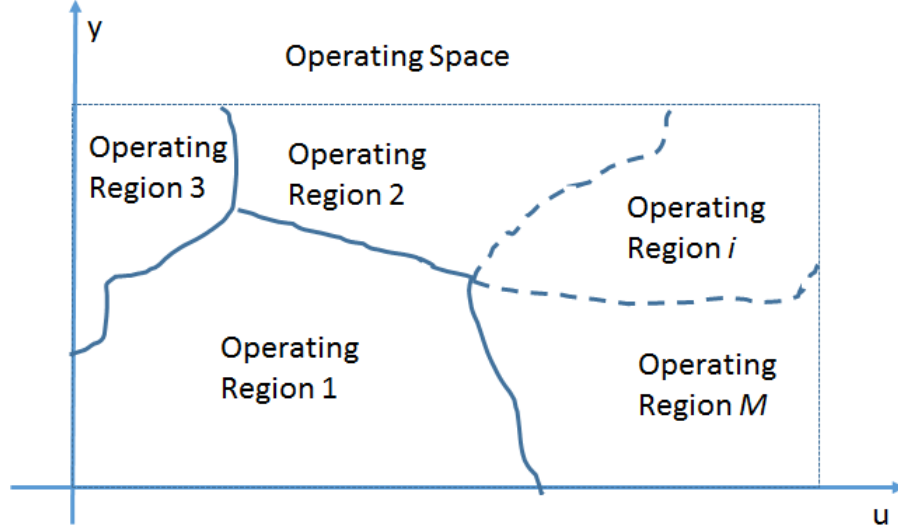


Figure 2.2: Operating space partition

of all submodels sum up to unity anywhere across the operating space. Hence the validity function satisfies the convexity property:

$$\sum_{i=1}^M \phi_i(z(k)) = 1 \quad \forall k \quad (2.3a)$$

$$0 \leq \phi_i(z(k)) \leq 1 \quad \forall k, \quad \forall i \in [1, \dots, M] \quad (2.3b)$$

In general, multi-model approach is a framework that accommodate different types of algorithms and models (either input-output models or state space models). Therefore, several paradigms exist in the literature that relate to the principle of multi-model, such as T-S fuzzy model, piecewise continuous model (PWC), piecewise affine (PWA), linear parameter varying model (LPV), local model network (LMN), etc.. Although these structures have been categorized into homogeneous and heterogeneous [42, 43] based on the submodels in a broad sense, they can be distinguished with respect to four features:

1. Partition strategy: One of the main differences between various approaches is the operating space partition strategy, which defines the operating space and structure of the local models. Popular partition strategies include operating point, axis-orthogonal, axis-oblique and clustering partition. In the partition process, optimizing the number of local models and their parameters is very essential for the correct representation of the system.

2. Submodel structural identification: This involves associating the sub-spaces with local models. The submodels can be selected to be linear, nonlinear or combination of the two. Also linear in-parameter models are preferred, especially in control applications due to their simplicity and evident extension to linear control theory.

3. Transition between models: This is the validity computation that determines the degree of contribution of each local models and more importantly subsumed the nonlinearities in the system. The choice of validity computation play crucial role in the accuracy of the multi-model identification approach [44, 45]. Transition between local models can either be hard or soft. For hard switching between the models, it is required that $\phi_i(z(k))$ is either 1 or 0, depending on whether a model is active or not at any instant. Soft switching allows a smooth transition between the models at the switching boundaries thereby allowing $\phi_i(z(k))$ to assume any value between 0 and 1, as in (2.3b). The value of 0 means no contribution and 1 for maximum contribution. In this review, our focus shall mainly be on multi-model tech-

niques where submodels are interpolated (soft switching).

4. Method of realization: This involves how the submodels are combined along with their validities to form the global system. One realization shown in Fig. 2.3 is the weighted sum of the submodel outputs to form the global system output. The discrete time state-space representation of this realization can be written as:

$$x_i(k+1) = f_i(x_i(k), u(k)) \quad (2.4)$$

$$y_i(k) = h_i(x_i(k), u(k))$$

$$y(k) = \sum_{i=1}^m \phi_i(k) y_i(k)$$

where $f(\cdot)$ is the state transition function, and $h(\cdot)$ is the output function. $x_i \in \mathfrak{R}^{n_i}$ is the state vector of the i^{th} submodel, $u \in \mathfrak{R}^l$ is the input, $y \in \mathfrak{R}^m$ is the output vector. This realization can cope with either homogeneous or heterogeneous sub-models. The weighted parameters is another realization when homogeneous submodels are utilized:

$$x(k+1) = \sum_{i=1}^m \phi_i(k) f_i(x_i(k), u(k)) \quad (2.5)$$

$$y(k) = \sum_{i=1}^m \phi_i(k) h_i(x(k))$$

where $f(\cdot)$ is the state transition function, and $h(\cdot)$ is the output function. $x_i \in \mathfrak{R}^n$ is the state vector of the i^{th} submodel, $u \in \mathfrak{R}^l$ is the input, $y \in \mathfrak{R}^m$

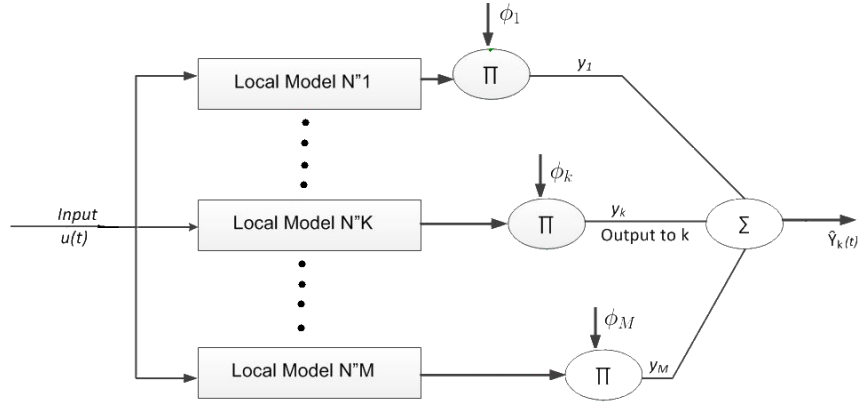


Figure 2.3: Multi-model identification

is the output vector. The linear submodels' case has been treated in [46]. The pros and cons of weighted outputs and weighted parameters realizations are discussed in [41]. Another possible realization in Equation (2.6) is the weighted input [47, 48].

$$x_i(k+1) = f_i(x_i(k), \phi_i(k)u(k)) \quad (2.6)$$

$$y_i(k) = h_i(x_i(k), u(k))$$

$$y(k) = \sum_{i=1}^m y_i(k)$$

where $f(\cdot)$ is the state transition function, and $h(\cdot)$ is the output function. $x_i \in \mathfrak{R}^{n_i}$ is the state vector of the i^{th} submodel, $u \in \mathfrak{R}^m$ is the input, $y \in \mathfrak{R}^m$ is the output vector.

Essentially, multi-model design involves three steps. The first is the partitioning of the operating system into smaller regions based on a selected strategy. In the second step, both the structure and the parameters of the local model associated with each subregion are determined. Finally, the local models are combined together using weighting function that defines the contribution of each local model to the real system. These steps might be dependent on one another depending on the adopted approach in the partitioning stage. It is interesting to know that challenges in the design of interpolated MMF lies in these three steps since its inception to present. Many attempts have been conducted to specify different approaches. In what follows, we shall make an attempt to examine recent contributions in these three key steps.

2.3 Multi-model Partition Strategy

Partitioning of the operating space system to different regions is the first and most critical step in MMF since other steps may depend on the type of partition used. Partitioning involves decomposing the operating space into a number of regions and describe the dynamic of the system using local models for each region. In some literature, this is called model base or model set design. The partition strategy can be categorized into prior-knowledge and non prior-knowledge partition. The prior-knowledge based partitions are sub-categorized into model-based and experimental-based, while the non-prior knowledge based may simply be referred to as data-based partition.

2.3.1 Model-Based Partition

Model-based partition takes into account prior knowledge of the system's nonlinear model and possibly its operating conditions. This partition includes operating point linearization, velocity-based linearization and sector nonlinearity partition.

Operating Point Linearization (OPL)

Partitioning with OPL involves local linearization of the systems's nonlinear model at different operating points covering the entire operating space of the system. OPL can be dynamic (linearization along a trajectory) as found in [49] or off-equilibrium [50] linearization. In dynamic OPL, linearization is based on nominal trajectory of the system to produce a linear time-varying system. The resulting model presents some difficulties during controller design [40]. Off-equilibrium linearization is based on a set of points in the off-equilibrium points to produce an affine local linear model. Although the scheme is more flexible than the dynamic linearization, it lacks direct relationship between the dynamics of the blended multiple model system and that of the local models. In addition, there is lack of linearity (in the sense of linear control theory) of the local models [45].

The gap metric concept has also been used as guide for selecting linearization points on steady state map of nonlinear system [51]. One important problem that arises here is the selection of optimal number of operating points that will be adequate representation of the original system. Recent solutions to this problem via the use of gap metric concept can be found in [52, 53, 54, 55, 56], where the

selection of sufficient numbers of submodels from different operating points of a nonlinear system are integrated into submodels controller design.

Velocity Based Linearization (VBL)

[45] developed VBL partition to circumvent previously mentioned drawbacks associated with off-equilibrium linearization that produces affine local linear model. VBL provides local models that are velocity based, linear and continuous in time. The global dynamics are directly related to the local model. It represents the system at every operating point in contrast to only equilibrium point in OPL. [44] investigated VBL approach by implementing continuous stirred tank (CSTR) with VBL and proposed construction of VBL from process data. They concluded that the approach is promising but unable to accurately model the steady state of the system. They also proposed alternative validity function (piecewise linear weighting function) and in [57] they showed that Gaussian validity function is not always the best option. [58] developed a discrete-time version of VBL which is originally a continuous time approach.

Despite its performance, VBL has some challenging requirements such as the need for the derivative of the input signal, the determination of validity function and the scheduling mechanisms, which can affect the accuracy of the approach. To address these difficulties, a fixed structure Gaussian process (GP) model [59] is recently proposed, which merges VBL with the GP modeling approach. Each GP model is used to represent an element of the unknown parameters of the local models from the VBL, thus, producing an LPV model. The approach has

automatic mechanisms for interpolating the values of the local model parameters owing to smoothness property of GP models. Furthermore, the selection of the scheduling variable is based on relevance-parameter detection capability of GP models. However, this approach fails to identify model outside the equilibrium regions [59].

Sector Nonlinearity Transformation (SNT)

Sector nonlinearity first appeared in [60] as a possible partition for T-S fuzzy model construction. It has since caught the attention of many researchers especially in the control community [61, 62, 63, 64, 65, 66, 67, 68]. SNT is a systematic and analytical procedure to transform a nonlinear system into a quasi-linear parameter varying (quasi-LPV) using a convex polytopic transformation with unmeasurable premise variable. This transformation is considered not suffers any loss of information and produces a system having the same trajectory as the original one [69]. Each quasi-LPV is associated with a particular set of premise variables. One drawback of this scheme is the non-uniqueness of the transformation as several equivalent quasi-LPV forms can be constructed for a given nonlinear system. Since each quasi LPV is associated with a particular set of premise variables, hence the selection of a particular premise variable is critical for it affects both the number of submodel, as well as the global model. Recently, a generalized sector nonlinearity approach has been proposed in [70, 71, 72] to make the selection of quasi-LPV easier. The obtained system is tailored along a particular objective such as stability or performance analysis, controller or observer design. The approach gives

a systematic procedure to choose the best quasi-LPV form that leads to a specific property needed in the global model. The contributions in [73, 74] worth mentioning for modeling T-S fuzzy with sector nonlinearity using input-output data.

2.3.2 Experimental-Based Partition

The experimental-based partition [e.g., 75, 76, 17, 77] assumes prior knowledge of the operating conditions of the system. It involves careful design of experiments for each known region of the operating space. An input excitation signal is designed around some chosen operating point of the process. Data are collected and local models are identified for each operating region. This type of partition is common with multi-model LPV system (MM-LPV) [see for example, 78, 79, 80, 81, 82, 83]. Static characterization [23, 41] is one of the strategies used in experimental-based partition. The system is excited only in the region of the operating space close to the target operating point. The curve of equilibria is plotted and a minimum number of operating point is carefully selected to cover the entire operating space. Separate sets of data are collected close to the vicinity of these operating points to identify submodels with centers lying on the equilibrium curve. Similar to other partition strategies, the choice and number of operating region to completely represent the system is challenging. One effort to this direction can be found in [84], where the authors gave the experimental conditions for the linear identification at each operating point to optimize the location of the operating points at which

local linear models can be identify.

2.3.3 Data-Based Partition

Data-based partition is the one that has attracted most attention in the literature. The entire system input-output data is used for identification of system while extracting the operating regions (partition) of the system from the data. In contrast to experimental partition, there is very little or no prior knowledge associated with partition extracted. Many approaches exist in the literature such as axes-orthogonal, axes-oblique, clustering and learned partition. In general these strategies can be categorized into Incremental, clustering and learned partition.

Incremental partition

Incremental partition are tree based algorithms where the system's data is partition iteratively in order to add a submodel at a time based on some criteria. Two common incremental partition are axis-orthogonal and axis-oblique partition.

Axis-orthogonal partition

Axis-orthogonal partition strategy involves splitting the data in a direction parallel to the axis of the input space hyper-rectangle(see Figure 2.4a). The algorithm in [24], subsequently refer to as (J&F) algorithm, and local linear model Tree (LOLIMOT) algorithm for training Local Linear Neuro-Fuzzy (LLNF) Model [29] are the two early partitions utilizing this strategy. Both algorithms produce local linear model weighted by normalized Gaussian function. Although the two algo-

rithms seems similar, they do have significant differences in the determination of which regime to decompose. In J&F algorithm many decompositions along the axis of the hyper-rectangle are possible and the search is within the sets of position leading to minimum error. On the other hand, LOLIMOT algorithm splits the input space is only into two halves along the hyper-rectangle axis. The parallel direction selected for splitting is determined by the minimum global error. Further decomposition is done by considering the region with the highest local error. Another difference resides in the estimation of the local model parameters. LOLIMOT employs weighted least square while singular value decomposition is used in J&F algorithm. Although LOLIMOT has the advantage of being computationally efficient compared to J&F algorithm it gives suboptimal model [85] and is very sensitive to curse of dimensionality [86].

[87] proposed an improved algorithm called Polynomial Model Tree (POLYMOT). Unlike LOLIMOT, a linearly parameterized higher degree polynomial based local models are used. The idea stems from the fact that, as the degree of the polynomial increases, the number of local models required for a given accuracy decreases. The main improvement in the algorithm is that at every iteration, there is a choice of either to split the worse local model or increase its number of parameters, whichever tends to lower the global model error. POLYMOT has been shown to give better accuracy and less local model compared to LOLIMOT . [88] also proposed similar algorithm to POLYMOT, where choice is made to either increase the complexity of the local model or the number of local models.

Table 2.1: Some modification to LOLIMOT algorithm

Author	Technique	Modification	Problem solved
[89]	expectation maximization (EM)	Use EM algorithm for identification of local models	provide covariance information about the model mismatch
[90]	Particle Swarm Optimization (PSO)	Use PSO to optimize the parameters of LOLIMOT	the unknown standard deviation of Gaussian validity function is optimized
[91]	particle swarm optimization (PSO)	Use PSO to find the best divisions of input space	search for the best axis-orthogonal partition of the input space
[92]	Simulated Annealing (SA)	Use SA to find the best divisions and provide merging of local models	Reduce the number of models in LOLIMOT
[93]	combine Piecewise Linear Network (PLN) and LOLIMOT algorithm	provide pruning strategy in form of merge and split to formerly divided local linear models	Reduce the number of models in LOLIMOT
[94]	state space identification	apply subspace identification method of N4SID to optimize the parameters of the local models	input-output local linear models is transformed to the Locally Linear State Space models (LLSSM)
[95]	EM and generalized total least squares (GTLS)	Use GTLS for parameter optimization and EM for determination of the region of validity for the local models	provide consistent estimate with input and output noisy data
[96]	LLNF as local models	use LLNF as local models to reduce both the number of models and parameters usually experienced in complex systems	fewer local models and parameters
[97]	clustering and model statistics	use clustering for partition and provide local model statistic which helps to estimate the reliability of the obtained model	less computational effort, fewer local models and generally uniform confidence intervals

However, the way this choice is made is different. They introduced a mechanism of orthogonal least square (OLS) with A-optimality to make that decision, by determining the significant terms of the two local models emanating from the split action. If the selected terms belong to both local models, then the number of local model is increased otherwise the complexity of the model is increased. The authors also showed that the algorithm is more accurate with less effective parameters than LOLIMOT. Generally, axis-orthogonal partition strategy tends to be sensitive to curse of dimensionality. As shown in Table 2.1, several other modifications to the LOLIMOT algorithm have been proposed in the literature, either for specific improvement or to be able to tailor it to a specific application.

Axis-oblique partition

Contrary to parallel splitting in axis-orthogonal, axis-oblique strategy splits the data at an angle to the axes of the input space. Figure 2.4 shows the difference between axis orthogonal and axis oblique partition with two dimensional input space. The axis-oblique strategy is first introduced in [98] using the hinging hyperplanes concept. The hinging hyperplanes are based on hinged basis function that are composed of two hyperplanes joined together at the point of intersection called hinge. The major task is to approximate the basis functions through function expansion, while optimizing the position and direction of the hinges. The localization of the hinges is actually an axis-oblique partition of the operating space. This is one of the strategies used in the concept of continuous piecewise linear (CPWL) models (e.g., [31, 99, 100]) which is out of scope of this study due to lack of interpolation of the submodels generated. To overcome hinging hyperplanes drawback of non-differentiability at the hinge point, [101] extended the hinging hyperplanes to the concept of LMN by interpolating the hinges with Sigmoid smooth function. [102] further modified the smooth hinging plane in [101] by introducing hinging hyperplanes tree, to decouple the parameters of the local models and the hinge directions (input space partition). Ernst's algorithm utilized a binary tree construction motivated by LOLIMOT algorithm. At each iteration the operating range is partitioned along the hinge into two halves representing two local submodels and the worst local model is replaced by a new hinge function. Since the hinge functions are nonlinear, a nonlinear optimization (gradient descent) is required to

estimate unknown parameters. This makes the algorithm more computationally demanding than LOLIMOT.

Hierarchical Local Model Tree (HILOMOT) is introduced in [103] by modifying Ernst’s algorithm. In HILOMOT, the idea of hinging function optimization is eliminated. In a binary tree construction, sigmoid function is introduced to split the worst local model and to automatically determine the validity region of the local models. The direction of split is optimized by using a nonlinear optimization to estimate the parameters of the Sigmoid function. The approach eliminates both the curse of dimensionality as well as local minima problems experienced in LOLIMOT. [104] proposed a refinement to HILOMOT algorithm to remove the unpleasant effect of overlap of the validity functions. In contrast to a prior fixed smoothness of the parameters, an automatic smoothness adjustment of the validity function parameters is developed. Further improvement to the algorithm is carried-out in [105] to reduce the training time of the nonlinear optimization (quasi-Newton) used for split position. This is achieved by replacing the numerical gradient calculation in the quasi-Newton method with analytical gradient. In [106], a similar strategy called supervised hierarchical clustering (SuhiClust) is developed. However, it differs from HILOMOT by only the method used to optimize the direction of split. Indeed, instead of using the sigmoid function for the split, Gustafson-Kessel (GK) fuzzy clustering is used to split the worst local model into two halves. Consequently, normalized membership function from the GK clustering result is used as weighting function for the local models created

by the split. This idea is similar to that used in [107], where fuzzy c-regression clustering is used to optimize the position of the hinges. A seemingly related idea is the modification of the classification and regression tree (CART) suggested by [108], where the decomposition of the tree is based on regression error and the parameters of the sigmoidal membership function are tuned by back propagation algorithm taking into consideration the global error.

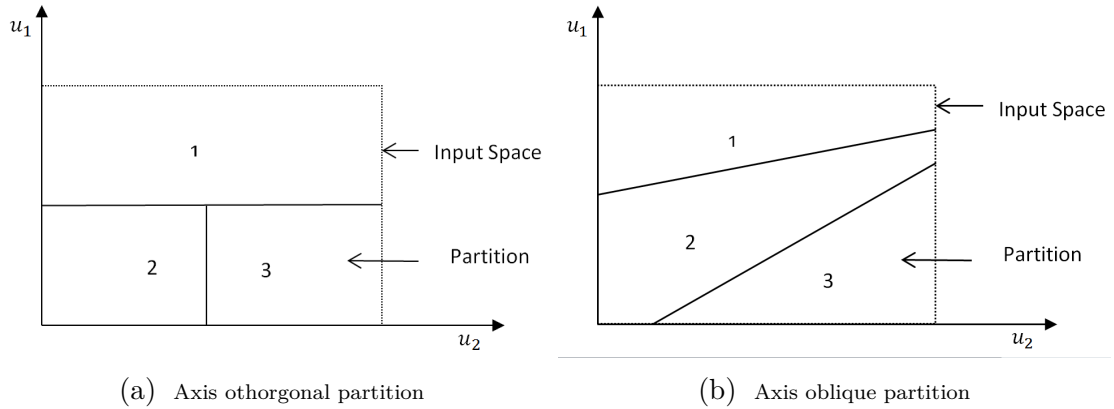


Figure 2.4: Axis-orthogonal and oblique partition for two dimensional input space (u_1, u_2)

Clustering partition

Clustering is another commonly used strategy that have also been exploited for the partitioning of process data. Clustering is based on unsupervised classification of a set of identification data set related to the underlying system. As a partition strategy, it enables the division of the complex nonlinear region into simpler subspaces, which are then associated with submodels. A number of clustering algorithms such as fuzzy c-mean [22], fuzzy k-mean [109], K-mean [110], Gustafson-Kessel [106, 111], Gath-Geva [112] etc. have been used for multi-model

identification. Broadly, they can be divided into two categories: input space clustering and product space clustering.

Input space clustering are those clustering algorithms that are based only on the input space data. One drawback of this strategy is that the local model form is based on input data distribution which is devoid of the process behavior [103]. Another drawback is the determination of sufficient number of input variables to adequately partition the data. For example in [109], the number of variables used for clustering is based on a try and error approach. This can also be seen as a scheduling variable problem (which will be discuss in next section), where the variables from the information space has to be determined for use in the validity function.

Product space clustering (e.g. Gustafson-Kessel (GK) and Getha-Geva (GG)), which is commonly used in the T-S fuzzy model [113, 114, 112], tries to overcome the drawbacks of the input space clustering by jointly considering both the input and the output data in the clustering process. It has the ability to identify local hyperplanes characterized by linear clusters. Thus, each cluster is suitable to model a local linear region of the complex nonlinear process. A possible drawback of GG and GK algorithm respectively is that of high sensitivity to initial value and inability to identify clusters of approximately unequal volumes [115]. Sensitivity to initialization can be reduced by using other input-space clustering techniques such as fuzzy c-mean clustering result as initialization of the prototype [115].

Similar to other partition strategies, a general challenge in using clustering

algorithms for partitioning of operating space is the determination of number of clusters to sufficiently represent the underlying system. Clustering into large number of partitions may represent the system behavior accurately but in a non-parsimonious way. In control applications, such representation can make controller design cumbersome. Also, a very small number of clusters may not represent the system adequately. In recent time, several methods have been proposed to solve the problem. [22] utilized both cluster validity (e.g. partition coefficient, partition entropy, partition density, Xie-Ben, etc) and model validity (Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Final Prediction Error (FPE)) methods to find the optimal number of clusters. [116] proposed a loss function, linear model based reconstruction error (LMRE), to determine optimal number of clusters resulting to minimum model error. Since both LMRE and model validity methods are based on model construction, they are time consuming and the computational effort increases with number of data samples and model dimensions. [117, 118] utilized a heuristic method based on the number of neurons in the output layer of Kononen network. This approach may not be free of the drawback of the previous methods as the training of the network has to be repeated manually until a satisfactory number of clusters is obtained. Moreover, the training of the network is very slow and increases with data samples.

[21, 119, 120] utilized subtractive clustering that automatically determines the number of clusters. The accuracy of this algorithm has also been known to depend

Table 2.2: Recent Clustering techniques for multi-model partition

Type of Clustering	Category	Papers
Fuzzy C-means	Input space	[22]
Fuzzy K-means	Input space	[110, 109]
k-means	Input space	[110, 109]
G.K	Product space	[116, 111, 123, 124]
Gath-Geva	Product space	112

on proper selection of its parameters [121, 122]. In [123, 124] large number of clusters is initially assumed and later reduced by merging similar clusters. Unlike in [125], where merging based on euclidean distance is suggested, the authors considered the merging based on stability of the local models using predictor error and gap metric criteria. [111] proposed an iterative incremental partitioning algorithm using G-K clustering. Number of clusters are iteratively increased by splitting the worst modeled cluster if its standard deviation error is greater than a certain threshold. [109] proposed the use of a separate clustering algorithm called rival penalized competitive learning (RPCL) neural network. Adequate number of clusters are determined by considering only clusters' centers that are enclosed by the data distribution when the initial number of cluster is larger than the real number of operating clusters. With all these proposals over the years, determination of optimal number of clusters is still an open area for research.

Learned Partition

This type of partition is based on parameterization of the operating partitions along with the parameters of the submodels. The parameters of the submodels

along with their partitions, and that of the validity function are learned directly from data. The submodels are usually specified and interpolated in traditional T-S fuzzy model fashion. Contributions in this area only differ in the approach used for estimation of parameters, such as gradient methods [126, 43].

2.4 Multi-model Internal Structure and Parameter Estimation

This section presents the identification of the submodels within the multi-model framework, when the partition of the system is derived from experimental and data-based partitions. In the literature, several structures, such as linear, non-linear, mechanistic, empirical, neural networks, polynomial, or hybrid, have been proposed. Recently, Gaussian process (GP) models is introduced as a local model structure [23, 76]. This approach gives a number of advantages like robustness to ill-conditioning, and provides a measure of uncertainty in the prediction. The structure of the submodels is the most flexible part of the multi-model framework, for there are no specific requirements other than a satisfactory approximation of the local regime [24].

In general, the submodels can be homogeneous or heterogeneous. Homogeneous submodels refers to models of the same structure, such as the one used in the well known Takagi-Sugeno (TS) models [30], while heterogeneous refers to submodels of different structure, commonly used in local model networks [41, 109].

Homogeneous submodels are mostly favored because their learning and optimization techniques are the same. On the other hand, heterogeneous submodels may require different learning and optimization techniques appropriate to each submodel. However, heterogeneous submodels are more flexible and can cope with curse of dimensionality unlike homogeneous submodels [43].

The estimation of local model's parameters given a particular model structure is generally done by either a global or local learning cost function [27, 41]. The objective of global learning is to minimize the error between the system's output and that of the multi-model's output. Hence, global learning estimates all the local models' parameters together. In (2.2), if N is the number of training data, the global learning criterion can be expressed as:

$$J_G = \frac{1}{2} \sum_{k=1}^N (\hat{y}(k) - y(k))^2 \quad (2.7)$$

where $y(k)$ and $\hat{y}(k)$ represent the actual system output and estimated system's output respectively. Global learning is accurate for a well chosen model structure. However, it is usually difficult to obtain a suitable model. In addition, it required large computational efforts for large training samples and produces less transparent models, since each submodel cannot be interpreted separately [127, 125].

Local learning as an alternative takes care of the disadvantages of global learning by focusing only on locally useful information from data. It minimizes the error between the system's output and all local model's outputs. Thus, it produces in-

dependent estimation of parameters of each submodel. The local learning can be defined as:

$$J_L = \frac{1}{2} \sum_i^M \sum_{k=1}^N \phi_i(k) (\hat{y}_i(k) - y_i(k))^2 \quad (2.8)$$

where $y_i(k)$ and $\hat{y}_i(k)$ are respectively the actual system's output and estimated output corresponding to the i^{th} submodel. Although local learning demonstrates superior performance to global learning, it has a disadvantage of discarding the useful global information from data. A comprehensive comparison between the two learning schemes can be found in [127].

Combined local and global learning has been suggested by [128] to make a compromise and also take advantage of the strength of both learning schemes. The combined criterion is defined as:

$$J = \alpha J_G(\theta) + (1 - \alpha) J_L(\theta), \quad \alpha \in [0, 1] \quad (2.9)$$

In recent times, efforts have been made to further investigate this idea in a different multi-model structure. [129] investigated the combined learning algorithm as a multi-objective optimization on T-S fuzzy model multi-model paradigms. Influence of α on the interpretation of the global model is examined and indicates some modeling conflict/sensitivity issues. Suggestions on detection and solutions to these conflicts are also pointed out. [18] presented a combined learning algorithm on polynomial local model with implementation on a thermal process. The algorithm combined both local and global cost functions to provide a trade-off

between local interpretation and global fitting. [43] further investigated three learning algorithms (local, global, and combined) on heterogeneous state-space models and advised that combined learning algorithm is well suited for a strongly overlapped Gaussian validity function.

In order to solve any of the optimization criteria (Equation 2.7, 2.8,2.9), different algorithms can be used for the estimation of submodels parameters depending on the structure of the submodels. Most commonly used identification algorithm is the least square (LS) algorithm or its recursive version (RLS) for linear in parameters models. Other algorithms employed as of recent include prediction error [82], expectation maximization (EM) algorithm [78, 80, 83, 130, 130, 131], gradient based algorithm [126, 132, 43] and sub-space method [94].

2.5 Validity Computation

Determination of the validity computation is another challenge in the interpolated multi-model framework. As discussed earlier, the weight or validity function in (2.2) describes the contributions of all the local models to the multi-model output and allows smooth transition between the local models. Thus the choice of this function can affect the accuracy of the representation [133, 45]. In general, two categories of validity computation can be identified in the literature. The first is pre-validity computation where the determination of the validity is done during the partition of the operating space. Its computation is therefore dependent on the partition strategy employed. The validity may be employed directly in the

estimation of the local model parameters, for example, by using weighted least square (WLS) where the weights in the WLS are the validities of the local models. The other category called post-validity computation is when the validity is computed after the local models have been identified and therefore independent of the partitioning strategy. Central to the two categories is the determination of the scheduling variable vector, that define the operating region of multi-model system and assist in its blending. It is required that scheduling vector should be a subset of the information space (e.g. regression variable) [134] to reduce the curse of dimensionality. However, [41] showed that reduced dimension of the vector can result in a decrease of accuracy and sometimes produces a discontinuous global model. On the other hand, an extended scheduling vector can result to an off-equilibrium problem. Therefore, an automatic way of identifying the best variables in the information space is still a challenging task. However, there are some validity computation, such as simple and reinforced residues, that are not specified as function of a scheduling variable. Such computations remove the burden of the determination of the scheduling variable. Table 2.3 highlight some recently used validity computation and their categories. Although several methods of computing validity, such as polynomial function [135, 80, 46], cubic spline function [136, 137], piecewise linear function [133, 48], continuously differentiable function [138], gap metric [139, 140, 141], exist in the literature, in what follows we discuss recent commonly used validity computations.

2.5.1 Gaussian validity

This is one of the most commonly used validity function. Its popularity is due to its smooth property. It is mainly a pre-validity computation which can be used in the determination of the parameters of the local model. However, it may be used like a post-validity computation. Although other forms are possible, commonly used Gaussian function can be define as:

$$\phi_i(k) = \exp\left(-\frac{1}{2}(z(k) - c_i)^T \sigma_i^{-2}(z(k) - c_i)\right) \quad (2.10)$$

where c_i and σ_i are the center and width of the Gaussian function for the i_{th} local model respectively. z is the scheduling variable. The determination of the center and the width is quite important in the accuracy of the identified system. For experimental-based partition, the center can be selected as the operating point of the data collected [e.g., 48, 136, 80, 78, 82]. However, this can be challenging for other partition strategies, hence different strategies have been adopted in their determination. In [75] the center of data is used as the center of a Gaussian function and optimized the width of the function by minimization of a mean square error over the training data. [88] used the center and the width of the hypercube of the data to determine the center and the width of the Gaussian function respectively. In [111, 124] the cluster center and fuzzy covariance matrix are utilized for the center and width of the Gaussian function, respectively. [81] used steepest descent method to determine the width while [83] uses expectation maximization algorithm. This indicates that there are no specific approach to

determine these variables.

2.5.2 Sigmoid validity

The Sigmoid function has been used in [103, 142] as a splitting function in axes-oblique partitioning algorithm. The Sigmoid function use in [103, 142] is :

$$\phi_i(z) = \frac{1}{1 + e^{-\kappa(v_{i,o} + v_{i,1}z_1 + \dots + v_{i,n}z_n)}} \quad (2.11)$$

where the vector v_i determines the direction of the soft split, the offset term $v_{i,o}$ determines the position of the split, and κ determines the smoothness of the split. While parameter κ is chosen heuristically, a nonlinear optimization technique is used to optimize the vector v_i , which make it computationally expensive.

2.5.3 Residue Approach

Residue approach [143, 110, 109, 144, 117] utilized the distance between the current output of the system and that of the local models. This validity computation is commonly used as post-validity, for the output of the local models are used in the computation. The residue is computed as :

$$r_i = \|y - y_i\| \quad i = 1, \dots, M \quad (2.12)$$

where y is the output of the system and y_i is the output of the local model. Commonly used residue approach are highlighted below.

Simple Residue Approach

The simple residue is given by

$$\phi_i = \frac{1 - \bar{r}_i}{M - 1} \quad (2.13)$$

where \bar{r}_i is the normalized residue is given by

$$\bar{r}_i = \frac{r_i}{\sum_{j=1}^M r_j} \quad (2.14)$$

Reinforced Residue Approach

The reinforced computation is expressed as

$$\bar{\Phi}_i = v_i \prod_{j=1, j \neq i}^M (1 - v_j) \quad (2.15)$$

where $v_i = 1 - \bar{r}_i$. The actual reinforced validity is given by normalizing $\bar{\Phi}_i$ as

$$\phi_i = \frac{\bar{\Phi}_i}{\sum_{j=1}^M \bar{\Phi}_j} \quad (2.16)$$

The residue approach is simple and free of the scheduling variable problem. However, it sometimes lack precision and not recommended for use in complex and ill-defined systems [144].

Quadratic Form

The quadratic form validity computation is proposed by [144, 118]. The idea is inspired by fuzzy c-means objective function based on minimization of quadratic criterion. The validity is shown to perform better than both simple and reinforced residues. However, it can only be used if clustering techniques are employed for the partitioning. The Quadratic validity computation can be written as

$$\phi_i(k) = \frac{1}{\sum_{l=1}^M (A_i^2(k)/A_l^2(k))} \quad (2.17)$$

where $A_i^2(k) = \|y(k) - c_i\|^2$, y is the output of the systems, and c_i is the cluster center of the i^{th} local model.

Bayesian Validity

The Bayesian validity [145, 146] employed the past history of residuals to obtain posterior probability of each model. A normalized posterior probability is then assigned to each model. The Bayesian validity is computed as:

$$P_{ri}(k) = \frac{\exp(-\frac{1}{2}\varepsilon_i^T(k)\Gamma\varepsilon_i(k))P_{ri}(k-1)}{\sum_{j=1}^M \exp(-\frac{1}{2}\varepsilon_j^T(k)\Gamma\varepsilon_j(k))P_{rj}(k-1)} \quad (2.18)$$

where $\varepsilon = y(k) - y_i(k)$ represents the residual between the measurement and the output prediction of the i^{th} local model at the k^{th} instant. $p_{ri}(k)$ is the posterior probability of the measurement. Γ is a time invariant weighting matrix known as convergence matrix and typically chosen to be diagonal. In the sense of normal

distribution, Γ is interpreted as the inverse of the residual covariance matrix. Higher values of diagonal elements of \mathbf{K} indicate a small residual variance and thus greater confidence in the residual of each model. The higher the values of the elements of Γ , the faster is the rejection of models with large residuals. The user-defined Γ allows strategies ranging from a winner-take-all approach (large Γ) to a non-discriminating averaging approach (small Γ). Finally, the validity corresponding to each model may be obtained as:

$$\phi_i(k) = \frac{P_{r_i}(k)}{\sum_{j=1}^M P_{r_j}(k)} \quad (2.19)$$

Neural Network and Fuzzy logic Validity

Neural networks and fuzzy logic validities are proposed in [119] and [147] respectively. Both methods use the residue and its variance for the prediction of model validity. In both methods, each local model validity is computed separately (a neural network and a fuzzy logic model for each local model is designed) and all the validities are normalized to satisfy the partition of unity. The idea is depicted in Fig. 2.5, where r_i is the residue, Δr_i is the variance of the residue and v_i is the estimated validity of the i^{th} local model, $i = 1, \dots, M$.

2.6 Applications

As mentioned earlier, due to the importance of modeling in many disciplines, multi-model approach has been applied in many areas. The rising number of

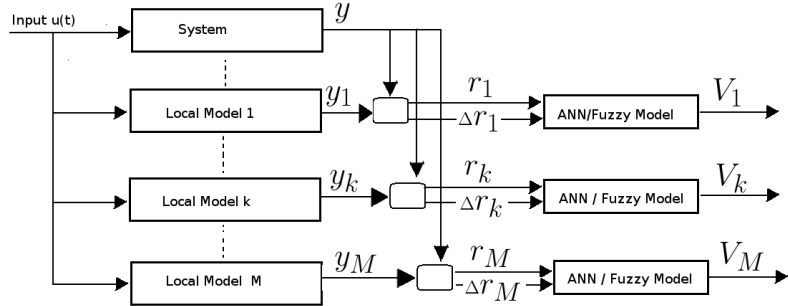


Figure 2.5: Neural Network and Fuzzy logic Validity computation

Table 2.3: Recently used Validity Computation

Validity Computation	Category	Paper
Gaussian function	Pre-validity	[75, 48, 136, 80, 78, 82, 81, 88, 111]
Sigmoid function	Pre-validity	[103, 142]
cubic spline	Pre-validity	[136, 137]
polynomial function	Post-validity	[80, 46]
Piecewise linear function	Post-validity	[133, 48]
Simple residue	Post-validity	[110, 109, 144, 117]
Reinforced residue	Post-validity	[110, 109, 144, 117]
Bayessian	Post-validity	[145, 146]
Neural network	Post-validity	[119]
Fuzzy logic	Post-validity	[147]
Quadratic criterion	Post-validity	[144, 118]
Gap metric	Post- validity	[139, 140, 141]

these applications is due to the increased awareness of the different communities in exploring the flexibility of the multi-model-based design. The range of applications of multi-model is getting wider and has been implemented among others in process optimization, prediction, fault detection, state estimation and control areas. In the remainder of this section, contributions on control application of interpolated multiple models is briefly discussed.

2.6.1 Multi-model control

multi-model framework has been exploited for nonlinear system control in order to avoid substantial demand in terms of design and implementation presented by nonlinear controls. Multi-model controller usually employ linear control to benefit from their easy implementation and rich linear control methodologies. In general interpolated multi-model framework deals with nonlinear system control through a fusion procedure of previously designed local controllers. At first, the nonlinear system is decomposed into a set of local linear models using any of the partitions and parameter estimations discussed earlier in section 2.3 and 2.4 respectively. Based on each local model $f_i(\cdot)$, a corresponding local controllers c_i is designed using well-known linear control techniques. Subsequently these controllers are fused together using their respective validity (weight) to form a global controller. Two popular methods exist for fusion controllers: partial fusion between the controllers and a fusion of the control-parameters.

In the partial fusion of controllers (for examples see [148, 149, 150]), the

outputs of local controllers are weighted, based on the contribution of each model, to obtain the final control signal for the system. Thus, the overall controller can be described by

$$u(k) = \sum_{i=1}^m u_i(k)\phi_i(k) \quad (2.20)$$

where $\phi_i(k)$ is the validity of each model obtained from the validity computation and u_i is the output of controller c_i . A pictorial description of this method is shown in Figure 2.6. This configuration allows different control algorithms to be design for each model representing the system. In fusion of the control-parameters

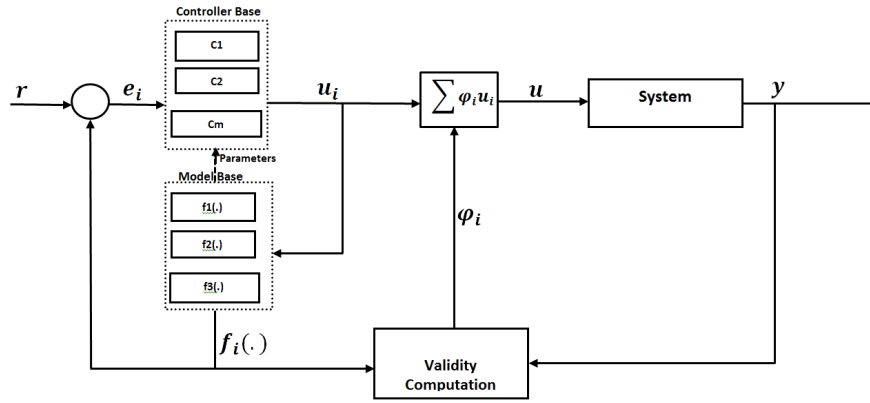


Figure 2.6: Fused controller multi-model control

(see [151]), the global control is computed by a fusion of the parameters of the local controllers weighted by the respective validity indexes. The global controller in this case is described by

$$p(k) = \sum_{i=1}^m P_i(k)\phi_i(k) \quad (2.21)$$

where p_i is a control parameter of the model $f_i(\cdot)$ and p is the global control

parameter. It should be noted that in order to be able to do fusion of controller parameters, same type of controller must be used for all local models. Figure 2.7 shows a pictorial representation of this methods. Another similar method of ob-

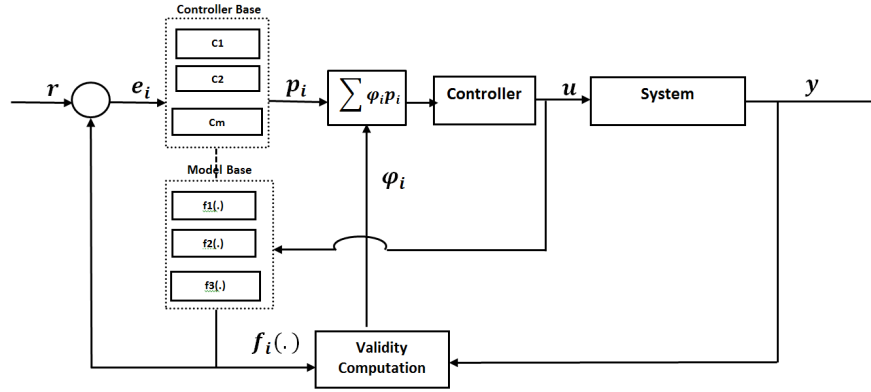


Figure 2.7: Fused parameters multi-model control

taining a global controller is by designing a single global controller from weighted multi-model output representation of the nonlinear system, rather than designing multiple controllers. This scheme is commonly used in model predictive controller [152, 146, 124, 153, 154, 155]. This method can ease the computational load in MPC optimization algorithm by solving only one control input sequence.

2.7 Discussion and Conclusion

Multi-model techniques for modeling and identification of complex nonlinear systems have attracted lots of attention over the years with different paradigms in the literature. In this chapter, we reviewed recent development in interpolated multi model techniques where the operating space is decomposed into a number of operating regimes and associated submodels in these regions are weighted and

combined in a way that represents the current system. Rather than enumerating all methodologies in this area, we have focused on three key challenging areas of multi-model design, which we are going to be our focus in this thesis, namely: partitioning, internal structure and parameter estimation as well as validity computation. We also review recent applications of interpolated multi-model framework. In general, it is observed that recent algorithms and methods are mainly in the partition and validity computation aspect of the multi-model framework.

The partition strategy has been broadly categorized into prior knowledge-based and non-prior knowledge-based. The prior knowledge partition includes experimental and model-based while the non-prior knowledge is basically the data-based partition. After evaluating a large number of papers, we observed that the current research trend in partition strategies is focused on data-based partition where no prior knowledge of the operating space is assumed. However, for all partition strategies, future research still needs to focus on optimizing the number of partitions which is directly related to the number of submodels. Another area that has not been giving much attention is the online partitioning scheme for online identification. Two efforts in this direction include [156] and [157] which utilized, respectively, evidential evolving GustafsonKessel algorithm and adaptive subtractive clustering for online partitioning of the operating region.

Validity computation is another design area that has several contributions due to its effect on accuracy of the multi-model representation. We have categorized all contributions into pre-validity and post-validity based on the dependency on

the partition strategy employed. It is therefore important to note that the partition strategy is one of the driving force in the determination of suitable validity computation. Therefore, future research in this area would necessarily focus on design and selection of suitable validity computation with respect to partition strategy. For example, the Gaussian function has become the de-facto for the homogeneous T-S multi-model and the local model network representation. However, as mentioned in section 2.3.1 this may not necessarily be the best choice.

CHAPTER 3

A MODIFIED

COMBINATORIAL PSO

BASED MULTI-MODEL

IDENTIFICATION OF

NONLINEAR SYSTEMS

Chapter 3 proposes a two-stage approach for the operating space partition, in order to obtain representative submodels for identification of a nonlinear system in multi-model framework. The approach uses a modified combinatorial particle swarm optimization and hybrid K-means to determine the number of submodels and their parameters. The main advantage of the proposed framework is in its automatic optimization of the number of submodels with respect to the submodel

complexity. This allows partitioning the operating space and generate a parsimonious number of submodels with little prior knowledge. Simulation examples are given to illustrate the effectiveness of the proposed algorithms.

3.1 Introduction

Several industrial systems are characterized by high nonlinearities with wide operating ranges and large set point changes. Identification and representation of these systems represent a challenge especially for control engineers. In recent years, much attention has been given to multi-model-based alternative approach to describe nonlinear systems. In contrast to conventional modeling technique, a system is represented by a set of models, that are combined, with different degree of validity, to form the global model. Each model represents the system in a specific region of operation. Owing to its potential benefits, this effective field of research has received several contributions, and has gained lots of interest in many fields of application such as biochemical [17], process control [18], communication [19], power systems [20], etc. Despite its benefits, the approach still faces several challenges.

As mentioned earlier, One major challenge of multi-model approach is the partitioning of the system's operating space to a number of sub-spaces. This further raises the question of how many submodels are required to adequately represent the entire operating region of the nonlinear system when combined within the multi-model framework. One solution to this problem is the design of identifi-

cation experiments for known operating spaces [e.g; 75, 76, 17, 77]. Data are collected for each operating space and a submodel is identified for each a priori known region. However, the knowledge involved in any industrial systems and processes, especially chemical process, is often incomplete. In addition they may be subjected to unknown parameter variations and exhibit wide operating ranges. Therefore, in such situations, partitioning of the operating space and identification of the submodels can be very challenging due to lack of prior information on the system's operating conditions.

Indeed, the problem of identifying the parameters of the submodels is coupled with the data partition problem, whereby each data point needs to be associated with the most suitable submodel. In the partitioning process, optimization of number of submodels and their parameters is very crucial for the correct identification of the system. While too few submodels can deteriorate the systems, increasing the number of submodels does not necessarily improve the performances obtained [158].

In the literature, many interesting algorithms have been proposed to address this challenge. [22] used fuzzy clustering for operating space partition and utilized both cluster and model validities methods to find the optimal number of clusters. The approach is manual and repeats the procedure for a number of submodels until a satisfactory number is obtained. [116] proposed a loss function, linear model based reconstruction error (LMRE), to determine the optimal number of clusters resulting to minimum model error. [117, 118] utilized a heuristic method

based on the number of neurons in the output layer of Kononen network. In this approach the network has to be trained repeatedly until a satisfactory number of clusters is obtained. Moreover, the training of the network is very slow and increases with the number of data samples. [21, 119, 120] utilized subtractive clustering that automatically determines the number of clusters. The accuracy of this algorithm was known to depend on proper selection of its parameters [122].

In [123, 124] large number of clusters was initially assumed and later reduced by merging similar clusters. [111] proposed an iterative incremental partitioning algorithm using G-K clustering. Number of clusters are iteratively increased by splitting the worst modeled cluster if its standard deviation error is greater than a certain threshold. [109] proposed the use of a separate clustering algorithm called rival penalized competitive learning (RPCL) neural network. Adequate number of clusters are determined by visual consideration of only clusters' centers that are enclosed by the data distribution when the initial number of clusters is larger than the real number of operating clusters. In general, all the aforementioned methods partitioned the operating space and/or determined the number of submodels based on data distribution only which may not reflect the complexity of the system's behavior.

In this chapter, an efficient method for obtaining the operating space partition without prior knowledge of the operating conditions is proposed. The proposed method utilized a two-stage method to obtain the partition and parameters of the submodels in the multi-model representation. In the first stage, estimation of

initial parameters and number of submodels are both obtained through a modified combinatorial Particle Swarm Optimization (PSO). To identify more efficient submodels hybrid K-means is used to obtain the final submodels in the second stage. The main advantage of the proposed framework lies in its automatic optimization of the number of submodels with respect to submodel complexity. This implies that the operating space of the system can be partitioned into a parsimonious number of submodels and the structure of the submodels can be assumed without prior knowledge. Thus, the algorithm can automatically find a good compromise between the number of submodels and complexity. Another interesting advantage is that the partition and selection of number of submodels are not only based on data distribution but also on the linearity of the operating region with respect to the linear submodels structure assumed. Benchmark simulation examples are provided to illustrate the effectiveness of the proposed method.

The rest of this chapter is organized as follows: section 3.2 describes problem formulation. In section 3.3, the first stage of the proposed multi-model approach is discussed, followed by the second stage in section 3.4. Simulation examples are provided in section 3.5, to demonstrate the effectiveness of the proposed method. Finally, a brief conclusion is given in section 3.6.

3.2 Problem Formulation

Generally, multi-model representation of complex nonlinear system involves interpolation of a number of submodels to form the global system (see Figure 3.1).

Considering a nonlinear system of the form:

$$y(k) = F(y(k-1), y(k-2), \dots, y(k-na), u(k-1), u(k-2), \dots, u(k-nb)) \quad (3.1)$$

where $u(k) \in \mathfrak{R}^{nb}$ is the input and $y(k) \in \mathfrak{R}^{na}$ is the output of the system. The integer nb and na are the time lag of the input and output respectively. The multi-model representation of the system can be describe by

$$y(k) = \sum_{i=1}^m f_i(x(k))\phi_i(k) \quad (3.2)$$

where m is the number of submodels, $f_i(\cdot)$ and $\phi_i(k)$ are the i^{th} submodel and validity function, respectively. The validity function describes the contribution of each submodel to the observed output and allows smooth transition between the local models when the system moves from one operating point to another. For easy interpretation, the validity function satisfies the convexity property [159, 132]:

$$\sum_{i=1}^m \phi_i(k) = 1 \quad \forall k \quad (3.3)$$

$$0 \leq \phi_i(k) \leq 1 \quad \forall k, \quad \forall i \in 1, \dots, M \quad (3.4)$$

Given a set of input-output data, the problem of obtaining the representative submodels is coupled with the data partition problem, whereby each data point needs to be associated with the most suitable submodel. In the partitioning process, optimization of number of submodels and their parameters is very crucial

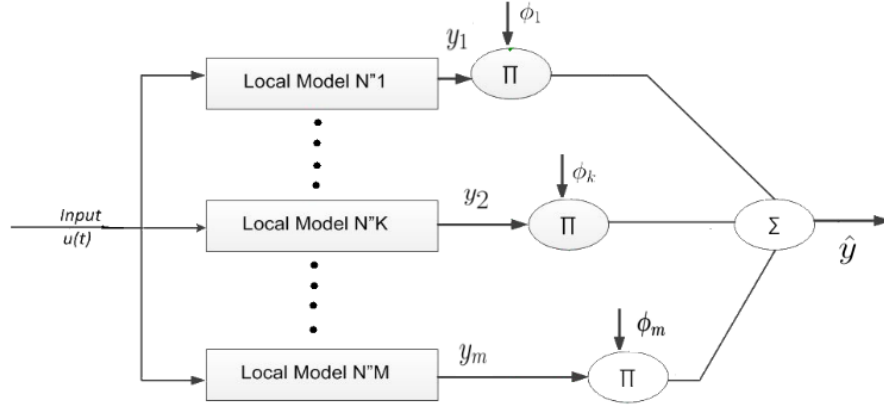


Figure 3.1: output blended multi-model identification structure

for the correct identification of the system. This is to estimate (i) number of submodels (M), and (ii) the parameters of each submodel (local model) ($f(\cdot)$).

Note that submodel $f(\cdot)$ can be linear, nonlinear or combination of the two submodels. In this study linear submodels are considered in order to exploit the linear control methodology when necessary. Therefore, the function $f(x_i(k))$ can be written as

$$f_i(x(k)) = x_i(k)\theta_i^T(k) \quad (3.5)$$

where θ_i is the vector of parameters of i^{th} the submodel which can be estimated from the data pairs:

$$z(k) = \{x(k), y(k) : k = 1, \dots, N\} \quad (3.6)$$

where $x(k) = [y(k-1), y(k-2), \dots, y(k-na), u(k-1), u(k-2), \dots, u(k-nb)]$ is the regressor vector. It is also possible that the regressor, $x(k)$, is affine such that $x(k) = [1, y(k-1), y(k-2), \dots, y(k-na), u(k-1), u(k-2), \dots, u(k-nb)]$.

Obtaining the submodels representative for multimodel identification of complex nonlinear systems from a finite set of input-output data is quite involving since the submodels' identification is dependent on the data partition. In order to solve the aforementioned problem, a two stage algorithm is proposed:

1. *Obtaining the number of submodels and the initial submodels' parameters.*

This stage involves application of modified combinatorial particle swarm optimization (MCPSO) to obtain m number of partitions and the representative data sets for each partition. The m number of clusters obtained from MCPSO is then used to estimate the initial submodels and initial cluster centers.

2. *Obtain the final submodels.* In this stage hybrid K-means criterion is applied to the result of the previous stage to refine the submodels, which can be presented for interpolation.

3.3 Stage 1: Obtain the number of submodels and the initial partition

The aim of this stage is to determine the number of partitions and to evolve a partition representing a possible grouping of the given data set. That is, given a data set $Z = [z_1, z_2, \dots, z_N]^T$ in \mathcal{R}^d , i.e. N points each with d dimension ($d = na + nb$), we need to simultaneously find the number of partition (m) and divide Z into m exhaustive and mutually exclusive clusters $P = [p_1, p_2, \dots, p_m]$

with respect to a predefined criteria such that:

1. $p_i \neq \emptyset \quad i = 1, \dots, m;$
2. $p_i \cap p_l = \emptyset \quad i = 1, \dots, m, i \neq l;$
3. $\cup_{i=1}^m p_i = Z$

To achieve this objective with partitional algorithm, a modification to combinatorial particle swarm optimization (CPSO) [160] is proposed. The modification is necessary since the CPSO algorithm required the number of clusters to be fixed a priori. In what follows, particle swarm optimization, CPSO and its modification for the determination of the initial submodels are described.

Remark: Notice that the regression matrix for the whole data space Z is constructed before the partition take place, since the order of the model is assumed known. This is to keep the time dependency of the data and hence keep the structure of each data point. Otherwise, the time dependency of each data point would be lost if the input and output data are used.

3.3.1 Particle swarm optimization

Particle swarm optimization (PSO) [161] is a metaheuristic search algorithm, mimicking the movement of organisms in a bird flock or fish school. Due to its simple concept and quick convergence, PSO has attracted much attention and wide applications in various fields, including systems identification problem [e.g., 162, 163, 164, 165, 166, 13, 167]. PSO combines self and social experience for

directing search to an optimal solution. The position of individuals referred to as particles is influenced by: its best position ever attained called Pbest, the position of the best in the swarm called the Gbest, and the current velocity v_i that drives the particles. At each generation t , each particle i , adjusts its velocity v_{ij}^t and position x_{ij}^t for each dimension j by referring to its personal best, Pbest, and the global best, Gbest. The following equations are used by the original PSO for velocity and particle update:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1r_1(Pbest_{ij}^t - x_{ij}^t) + c_2r_2(Gbest^t - x_{ij}^t) \quad (3.7)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (3.8)$$

where v_{ij}^t and x_{ij}^t are the j^{th} element of the i^{th} velocity and particle vector respectively at generation t . $Gbest^t$ and $Pbest_{ij}^t$ are the global and personal best position of i^{th} particle during iterations 1 to t , respectively. w is the inertia weight that controls the impact of the previous velocities on the current velocity. r_1 and r_2 are uniformly distributed random variables in range $[0, 1]$, c_1 and c_2 are the acceleration constants.

3.3.2 CPSO based partition

CPSO [160] is an extension to the original PSO to be able to cope with clustering problem. It has similar procedure as the original PSO except that it differs in two characteristics: particle and velocity representation. In CPSO particles X_i are

encoded with label-based representation, where $X_i = [x_{i1}, x_{i2}, \dots, x_{iN}]$ provides integer numbers representing the cluster number of data points, such that $x_{ij} \in \{1, 2, \dots, m\}$ and m is the number of cluster. The velocity of each particle uses a dummy variable that permit transition from combinatorial to continuous state and vice versa. Thus, the velocity and particle are updated through the following equations :

$$v_{ij}^{t+1} = wv_{ij}^t + r_1c_1(-1 - y_{ij}^t) + r_2c_2(1 - y_{ij}^t) \quad (3.9)$$

where y_{ij} is a dummy variable defined by:

$$y_{ij}^t = \begin{cases} 1 & \text{if } x_{ij}^t = Gbest^t, \\ -1 & \text{if } x_{ij}^t = Pbest_{ij}^t, \\ -1 \text{ or } 1 \text{ randomly} & \text{if } x_{ij}^t = Gbest^t = Pbest_{ij}^t \\ 0 & \text{otherwise} \end{cases} \quad (3.10)$$

After velocity update, the position of each particle is updated through the dummy variable according to the following equations:

$$\lambda_{ij}^{t+1} = y_{ij}^t + v_{ij}^{t+1} \quad (3.11)$$

$$y_{ij}^{t+1} = \begin{cases} 1 & \text{if } \lambda_{ij}^{t+1} > \alpha, \\ -1 & \text{if } \lambda_{ij}^{t+1} < \alpha, \\ 0 & \text{otherwise} \end{cases} \quad (3.12)$$

$$x_{ij}^{t+1} = \begin{cases} Gbest & \text{if } y_{ij}^{t+1} = 1, \\ Pbest & \text{if } y_{ij}^{t+1} = -1, \\ \text{a random number} & \text{otherwise} \end{cases} \quad (3.13)$$

where α is determined by the user.

3.3.3 Modified CPSO (MCPSO) based partition

This section introduced the MCPSO and how it is used to determine the number of submodels and the initial partition. As mentioned earlier, the CPSO algorithm is modified since contrary to our case, the number of partitions in the algorithm is fixed a priori. Four features that are introduced and distinguish MCPSO from CPSO are discussed as follows:

1. *Particles encoding*: Similar to CPSO, MCPSO uses the label-based integer encoding to represent each particle. However, instead of assigning the same number of clusters to all particles, in MCPSO each particle evolve with its own number of clusters. Each particle position $X_i = [x_{i1}, x_{i2}, \dots, x_{iN}]$, characterized by N elements, where N is the number of data points, provides integer numbers representing the cluster number of each data point, such that $x_{ij} \in \{1, 2, \dots, m_i\}$ represents the cluster number of j^{th} data point in i^{th} particle and m_i is the number of clusters associated with i^{th} particle. m_i is assumed to lie in the range $[m_{min}, m_{max}]$, where m_{min} is 2 by default and m_{max} is manually specified by the user. The particle and velocity updates

follow that of CPSO (Equation 3.9 through 3.13).

2. *Avoiding empty clusters:* In the label-based representation, it is possible to generate a solution with empty cluster if the number of its clusters is smaller than the largest cluster number associated with the particle solution. To avoid this, new positions of particles are checked. At each generation, particles with empty cluster are corrected by changing the largest cluster number to the smallest unused cluster number.

3. *Fitness function:* The fitness criterion used in CPSO are the variance ratio criterion (VRC) and sum of square error (SSE) of the cluster. The SSE is not appropriate when the number of clusters is not known in advance. This is because the maximum number of clusters will always be favored as its value will decrease as the number of cluster increases. On the other hand, VRC has been used when the number of clusters is not known. However, in order to reflect the peculiarity of the problem at hand in MCPSO, a fitness function based on cluster regression error fused in minimum descriptive length (MDL) [168] framework is used. Given the data set Z defined in equation(3.6), the cluster regression error is defined by:

$$CRE = \sum_{i=1}^{m'_{max}} SE \quad (3.14)$$

where

$$SE = \left(\frac{1}{n_i} \sum_{j=1}^{n_i} (y_j - x_j \theta_i^T)^2 \right)^{\frac{1}{2}} \quad (3.15)$$

m'_{max} is the maximum number of clusters assigned to a solution, n_i is the number of data points in the i^{th} cluster and θ_i is the parameter of the linear model associated with the i^{th} cluster. This can be obtained using the least square technique as follows:

$$\theta_i = \left[\sum_{j=1}^{n_i} x_j x_j^T \right]^{-1} \left[\sum_{j=1}^{n_i} y_j x_j \right] \quad (3.16)$$

Finally, the fitness function is defined by:

$$fitness = (m'_{max} \log N + N \log(CRE^2))/2 \quad (3.17)$$

where N is the total number of data points. The smaller the fitness value, the better is the clustering solution.

4. *Avoiding small size data:* A situation may occur where the number of data points assigned to a cluster is too small. On one hand, if the number of data points n_i is less than the dimension of the data, $d = na + nb$, then the model obtained from the cluster will be singular. On the other hand, if $n_i \geq d$ but $n_i < td$, where td signifies a reasonable minimum number of data points (i.e 5% of data points), then the model obtained may not be well define. In order to avoid these two situations, when $n_i < 0.05N$ the SE in equation (3.15) is not calculated from the data. Rather, a penalty value of 1000 is assigned to the SE . This penalty value is used to discourage having small size of data points in a cluster. As such, it can be any value higher enough

than the SE value for acceptable number of data points.

After the encoding of the particles as discussed above, the execution of MCPSO to obtain number of clusters and the initial partition is done according to the following steps:

Step 1: Initialize particle position vector X and associated velocity V in the population randomly. For each particle i , the number of clusters m_i is randomly generated in the range $[m_{min}, m_{max}]$, then each data point is randomly assigned to one cluster.

Step 2: Evaluate the fitness function for each particle using equation (3.17)

Step 3: Compare the fitness value of each particle with its previous best solution ($Pbest$) fitness and update $Pbest$ with the current solution if it is better than the previous value ($Pbest$).

Step 4: Compare fitness value with the overall previous best ($Gbest$) fitness. Update $Gbest$ to the current solution if its fitness value is better than $Gbest$ fitness value.

Step 5: Update positions and velocities of particles using equation (3.9) to (3.13).

Step 6: Check for empty cluster in all particle solutions and correct if exist.

Step 7 Repeat Step 2 to Step 6 until the maximum number of iterations is completed.

3.3.4 Estimation of the initial submodels

Given a set of cluster representatives $\tilde{z} = \{\tilde{z}_i = (\tilde{x}_i, \tilde{y}_i), i = 1, \dots, m\}$ from the MCPSO algorithm, where m is the number of clusters, the next task is to estimate the initial submodels. For this purpose, a least square estimation is applied to the data set in each cluster to find the initial submodel. The coefficients vector θ_i for each submodel is computed through the formula:

$$\tilde{\theta}_i = (\tilde{\Phi}_i^T \tilde{\Phi}_i)^{-1} \tilde{\Phi}_i^T y_i \quad (3.18)$$

where $\tilde{\Phi}_i = [x_i(1), \dots, x_i(n_i)]^T$ and y_i are the regression matrix and output vector belonging to i^{th} cluster, respectively. n_i is the number data in the i^{th} partition.

In addition, the centers of the data are calculated by finding the mean of the data in each cluster produced by the previous stage. The center of each cluster is given as :

$$\tilde{c}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \tilde{x}_{ij} \quad i = 1, 2, \dots, m. \quad (3.19)$$

3.4 Stage 2: Obtain the final submodels

This stage involves a refinement to the submodels produced in the previous stage. In order to achieve this objective, a hybrid K-means criterion [169] is adopted. Given a data set $Z = [z_1, z_2, \dots, z_N]$ in \mathcal{R}^d , K-means algorithm group the data Z into k clusters p_i $i = 1, \dots, k$ such that an objective function is minimized. The K-means objective function is defined as the sum of square error between each

data point and the corresponding cluster center:

$$J_1(c) = \sum_{i=1}^m \sum_{z_j \in u_i} (z_j - c_i)^2 \quad (3.20)$$

where z_j and c_i are the data point and cluster center, respectively. The objective function is minimized using an alternating optimization procedure. It starts with an arbitrary k number of centers and assign each data point to the nearest center. The assignment of each data is define by a binary membership matrix \mathbb{U} , such that:

$$u_{ij} = \begin{cases} 1 & \text{if } (z_j - c_i)^2 \leq (z_j - c_k)^2, \quad i \neq k \\ 0 & \text{otherwise} \end{cases} \quad (3.21)$$

Next, each center is updated as the mean of all points assigned to it. These two steps are carried-out iteratively until a predefined termination criterion is met, which occurs when there is no change in the objective function.

In the same spirit of K-mean algorithm, the following linear regression loss function can also be formulated:

$$J_2(\theta) = \sum_{i=1}^m \sum_{\tilde{x}_j \in u_i} (y_j - \tilde{x}_j \theta_i^T)^2 \quad (3.22)$$

Thus, instead of minimizing the cluster error objective function, a linear regression objective function is minimized. Combining the two objectives J_1 and J_2 , the

hybrid K-means objective can be written as:

$$J(\theta, c) = \lambda J_1 + J_2 \quad (3.23)$$

where $\lambda \in [0, 1]$ is a constant term to be defined by the user to specify a relative weight of the objective function. This formulation allows not only partitioning of the data set but also associating a submodel to each partition. In addition, the partitions formed are guided toward linear regions. This fits perfectly into the problem definition, since the aim of partitioning the input space is to form linear submodel for each partition. Furthermore, the inclusion of J_1 will allow us to assign new data to a partition in a situation where the submodels need to be updated online.

The description of how this stage utilizes the hybrid K-means algorithm is shown in Table 3.1. It begins with using the previously estimated cluster centers and associated model parameters as initialization of the algorithm. This eliminates the burden of the determination of the number of clusters, diminishes the effect of initialization as well as increases the convergence rate of K-means algorithm . Line 2 starts a loop which repeats itself for as long as there is a significant change in the objective function. It begins by determining the membership matrix U by Equation (3.21), which assign each data point to a cluster. Line 4 estimates the parameter vector θ_i and the center c_i for each cluster. Next, from line 5 to 13, undefined clusters are detected and removed from subsequent update. An undefined cluster is characterized by singular cluster, that may result when the

size of the cluster falls below the number of regressor.

Line 8 to 13 remove a cluster from the pool if the number of data points in the cluster is less than the number of regressor. Also the number of cluster is reduced by 1. The break statement in line 12 ensures only one cluster is removed at each iteration when undefined cluster is detected. This is to allow other undefined clusters, if exist, to readjust during the next iteration and probably able to circumvent undefined status. Line 15 computes the objective function J according to equation (3.23) while line 16 increments the number of iteration. Next the loop goes back to line 3 to repeat the procedure.

Once the algorithm is completed, the final parameters θ of each submodel are obtained along with their associated centroid c_i . The submodels are now ready for interpolation to obtain the final global model that will represent the system under consideration.

3.5 Simulation Examples

The effectiveness of the proposed partition method is demonstrated in this section. Five simulation examples were carried out. Since it is assumed that the number of parameters of submodels is not known, two and four parameters submodel structures are examined with the proposed approach to illustrate its flexibility on the number of parameters selected for the submodels. The assumed two and four

Table 3.1: Stage 2: estimating final submodels

1:	Initialize $c_i = \tilde{c}_i, \theta_i = \tilde{\theta}_i, i = 1, \dots, m, l = 1$
2:	repeat
3:	Assign data to a cluster such that
	$u_{ij} = \begin{cases} 1 & \text{if } (y_j - \tilde{x}_j \theta_i^T)^2 + \lambda(\tilde{x}_j - c_i)^2 \\ & \leq (y_j - \tilde{x}_j \theta_k^T)^2 + \lambda(\tilde{x}_j - c_k)^2, \quad i \neq k \\ 0 & \text{otherwise} \end{cases}$
4:	compute center θ_i and c_i
	$\theta_i = \left[\sum_{x_j \in u_i} x_j x_j^T \right]^{-1} \left[\sum_{x_j \in u_i} x_j y_j \right], \quad i = 1, \dots, m$
	$c_i = \frac{1}{ U_i } \sum_{x_j \in u_i} x_j, \quad i = 1, \dots, m, \text{ and } U_i = \sum_{j=1}^{n_i} u_{ij}$
5:	Remove undefined cluster as follows
6:	for $i = 1 \dots m$ do
7:	if $n_i < na + nb$ then
8:	$c_i = \emptyset$
9:	$\theta_i = \emptyset$
10:	$m = m - 1$
11:	break
12:	end if
13:	end for
14:	Compute J^l
15:	$l = l + 1$
16:	until $\ J^{l-1} - J^l\ \leq \epsilon$

parameters submodel structures are respectively given by:

$$y_i(k) = a_{i1}y(k-1) + b_{i1}u(k-1) \quad (3.24)$$

$$y_i(k) = a_{i1}y_i(k-1) + a_{i2}y_i(k-2) + b_{i1}u(k-1) + b_{i2}u(k-2) \quad (3.25)$$

where $a_{i1}, a_{i2}, b_{i1}, b_{i2}$ are the i^{th} submodel scalar parameters to be identified by stage 1 and stage 2 above. The two structures in equation (3.24) and equation (3.25) subsequently refer to as 2-parameters and 4-parameters structure respectively. Except stated otherwise, the parameter settings of MCPSO used are given in Table 3.2. Also $\lambda = 0.01$ is selected in stage 2 throughout the simulations.

In order to form the multi-model representation of system we need to estimate the validity of the submodels, which is not the goal of this chapter. Therefore, it is sufficient for us to use some of the validity estimation in the literature: simple residue, reinforced residue, bayessian and quadratic methods, mentioned in chapter 2 to test the results of our simulation. Subsequently in chapter 4 we shall design another suitable validity estimation algorithm to be used for the same system for easy comparison.

The obtained multi-model is evaluated based on the validation data using the mean square error (MSE), percentage model fitness (PMF) and variance-accounted-for (VAF) performance measures:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y(i) - \hat{y}(i)) \quad (3.26)$$

$$PMF = \max\left(1 - \frac{\|y - \hat{y}\|}{\|y - \text{mean}(y)\|}, 0\right) \times 100 \quad (3.27)$$

$$VAF = \max\left(100 \times \left(1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)}\right), 0\right) \quad (3.28)$$

where y is the real system output, \hat{y} is the multi-model estimated output, $\|\cdot\|$ denotes norm and $\text{var}(\cdot)$ denotes the variance. All simulations are performed

using MATLAB 2012b on a 2.4 GHZ i3 64-bits Windows machine with 4 G RAM.

Table 3.2: MCPSO parameter settings

parameters	values
Swarm size	20
Max. Iterations	2000
w, α	0.4, 0.35
c_1, c_2	2, 2
m_{min}, m_{max}	2, 20

3.5.1 Example 1

In the first example, a discrete-time system from [109] is considered. The system is described by

$$y(k) = a_1(k)y(k-1) + a_2(k)y(k-2) + b_1(k)u(k-1) + b_2u(k-2)$$

The variation laws of different parameters of the process as shown in figure 3.2 is given by

$$a_1(k) = 0.04\sin(0.035k) - 0.8$$

$$a_2(k) = 0.005\sin(0.03k) + 0.1$$

$$b_1(k) = 0.02\sin(0.03k) + 0.5$$

$$b_2(k) = 0.01\sin(0.035k) + 0.2$$

The system was excited with uniform random signal $u(k)$ on the range $[-1, 1]$. 600 data points were generated for the two submodel structures above. The

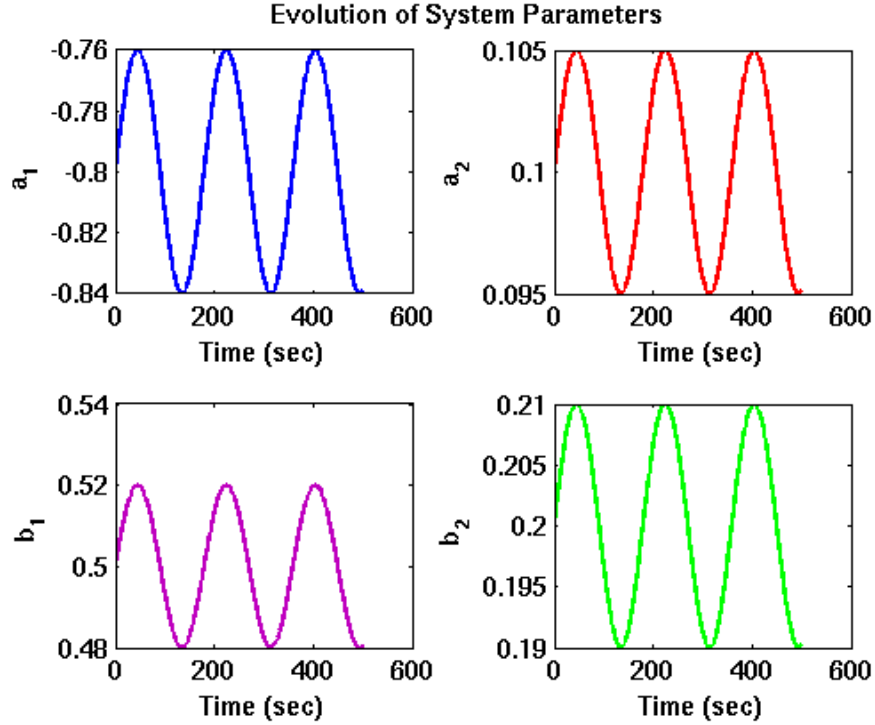


Figure 3.2: output blended multi-model identification structure

initial submodels and centers obtained in stage 1 is as shown in Table 3.3 while Table 3.4 shows the final submodels and centers obtained in stage 2. It can be observed from the Table 3.3 that four and two submodels were identified in stage 1 for 2-parameters and 4-parameters structures, respectively. The convergence paths of the objective function in the developed MPSO are shown in Figure 3.3. The figures show faster convergence with 4-parameters structures than with 2-parameters structures.

In the validation stage, a different input signal ($u(k) = 1 + \sin(0.06k)$) was injected into the real systems and the identified submodels. The submodels were interpolated with the validity estimation methods mentioned previously to form the multi-model output. The real system's output and the multi-model output for

Table 3.3: Results of Stage 1(initial submodels)

Initial submodels	2-parameters structure		4-parameters structure			
	a_1	b_1	a_1	a_2	b_1	b_2
1	0.80021	0.46683	0.90904	-0.17929	0.50286	0.13472
2	0.79705	0.50171	1.041831	-0.2762	0.50508	0.05015
3	0.81370	0.42337	-	-	-	-
4	0.75039	0.61168	-	-	-	-
Initial centers						
1	0.11698	0.05276	0.1198	0.11971	0.05152	0.05091
2	0.11998	0.05119	0.11543	-0.11181	0.04611	-0.05522
3	0.11966	0.052531	-	-	-	-
4	0.121169	0.04929	-	-	-	-

Table 3.4: Results of Stage 2 (final submodels)

Final submodels	2-parameters structure		4-parameters structure			
	a_1	b_1	a_1	a_2	b_1	b_2
1	0.79812	0.48362	0.93806	-0.19068	0.51385	0.10845
2	0.82082	0.55976	0.80392	-0.12192	0.52162	0.18125
3	0.75896	0.42027	-	-	-	-
4	0.64245	0.84146	-	-	-	-
Final centers						
1	0.13079	0.07222	0.14561	0.14268	0.05090	0.06226
2	0.12940	0.04096	0.09846	0.10036	0.05154	0.04208
3	0.10844	0.05039	-	-	-	-
4	0.09359	0.03171	-	-	-	-

the two assumed submodels' structures are compared using the MSE and VAF.

The multi-model identification results for the different validity estimation are shown in Table 3.5 and Figures 3.4 to 3.5. As can be observed, the proposed multi-model can well approximate the real system with either 2-parameters or 4-parameters structures. Simple and reinforced residue methods perform better than both Bayesian and Quadratic criterion methods.

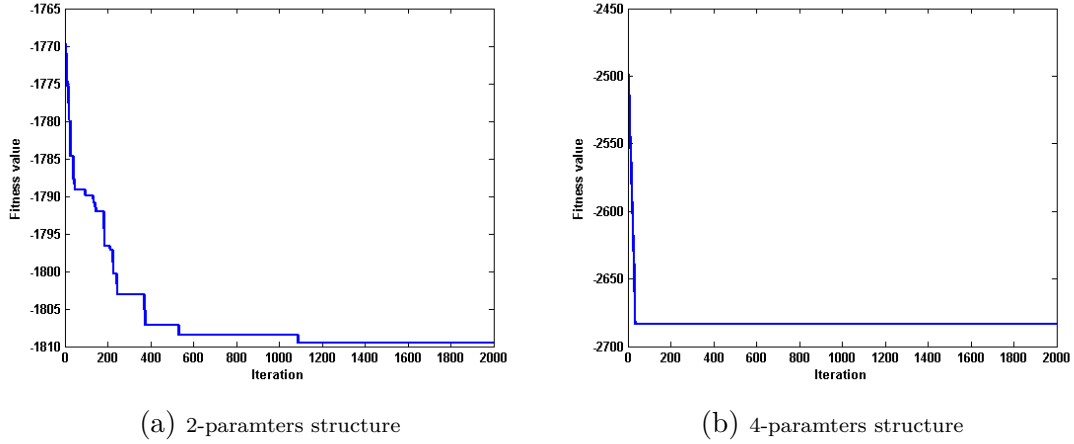


Figure 3.3: MCPSO objective function convergence plot

Table 3.5: Validation performance test on common validity estimations

Validity Estimation	2-paramters structure			4-paramters structure		
	MSE	PMF(%)	VAF(%)	MSE	PMF(%)	VAF(%)
simple Residue	0.0403	87.87	98.54	0.0277	89.91	99.00
Reinforced residue	0.0444	87.23	98.40	0.0277	89.91	99.00
Bayessian	0.0751	83.40	97.35	0.0576	85.46	97.89
Qaudratic	0.0765	83.24	97.35	0.0576	85.46	97.891

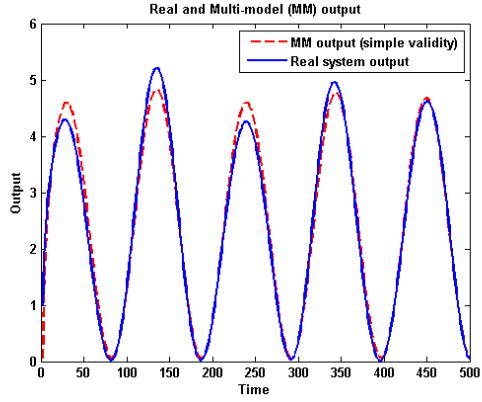
3.5.2 Example 2

A nonlinear dynamical system taken from [43] is considered for multi-model identification:

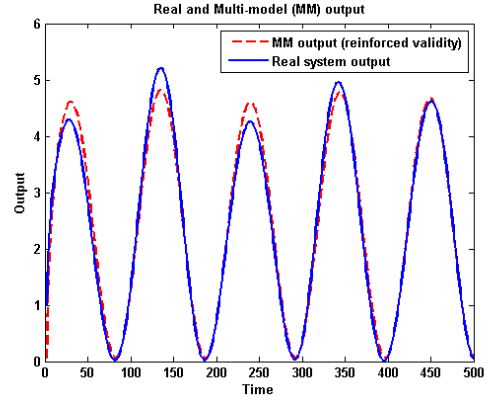
$$y(k+1) = (0.6 - 0.1a(k))y(k) + a(k)u(k) \quad (3.29)$$

$$a(k) = \frac{0.6 - 0.06y(k)}{1 + 0.2y(k)} \quad (3.30)$$

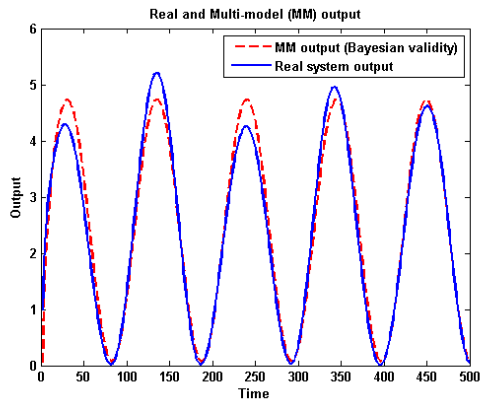
The identification was carried out using the proposed approach with data set of 600 samples of uniform random signal within the range of $[-0.9, 0.9]$. Tables 3.6 and 3.7 show the initial and final submodels parameters with their associated



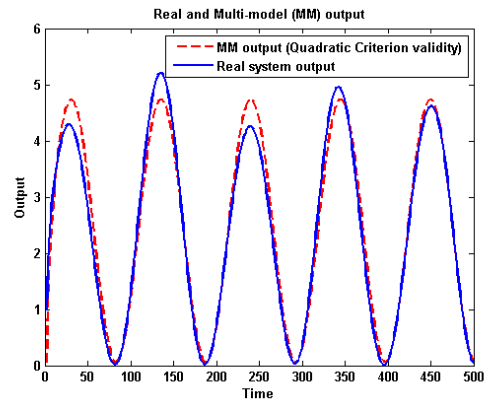
(a) 2-paramters structure with simple residue



(b) 2-paramters structure with reinforced residue



(c) 2-paramters structure with Bayesian



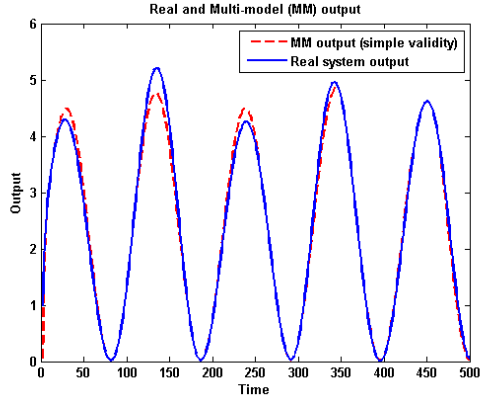
(d) 2-paramters structure with Quadratic

Figure 3.4: Multi-model identification outputs using validation data

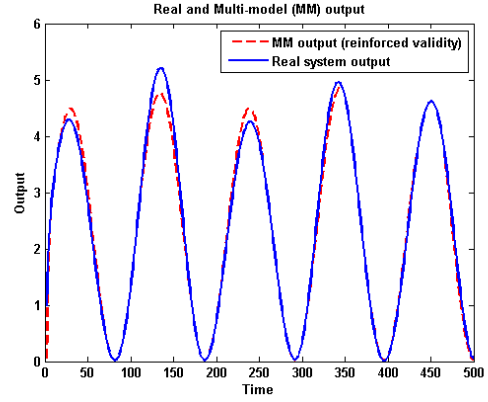
centers. The proposed multi-model partition identified two submodels for both 2-paramters and 4-paramters structures. Figure 3.6 shows the convergence of the objective function in the developed MPSO. In this case both structures have similar convergence.

To form the multi-model representation with the previously mentioned validity estimation, a validation data under the following input was used:

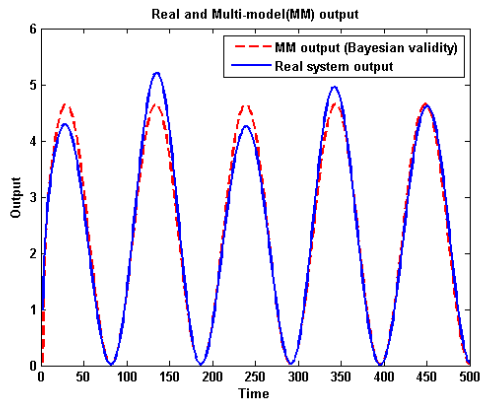
$$u(k) = \sin\left(\frac{2\pi}{25}k\right)$$



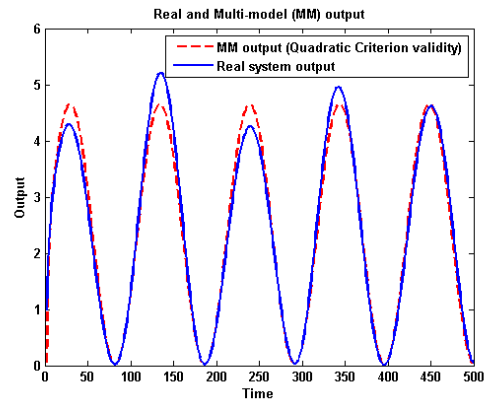
(a) 4-paramters structure with simple residue



(b) 4-paramters structure with reinforced residue



(c) 4-paramters structure with Bayesian



(d) 4-paramters structure with Quadratic

Figure 3.5: Multi-model identification outputs using validation data

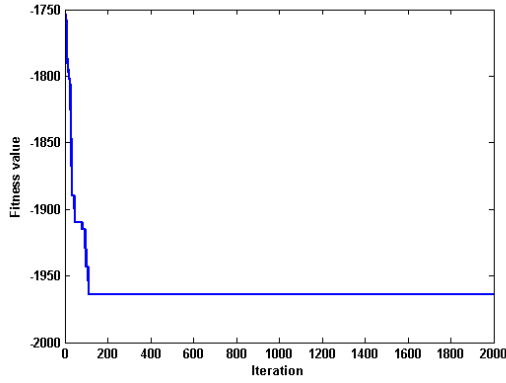
Table 3.6: Results of Stage 1 (initial submodels)

Initial submodels	2-paramters structure		4-paramters structure			
	a_1	b_1	a_1	a_2	b_1	b_2
1	0.54650	0.60674	0.47613	0.03870	0.60722	0.04359
2	0.56813	0.58940	0.61421	-0.04158	0.57869	-0.040053
Initial centers						
1	-0.01412	-0.00080	-0.01639	-0.01316	-0.01488	-0.01593
2	0.07727	-0.16727	0.12038	0.05947	0.10028	0.13306

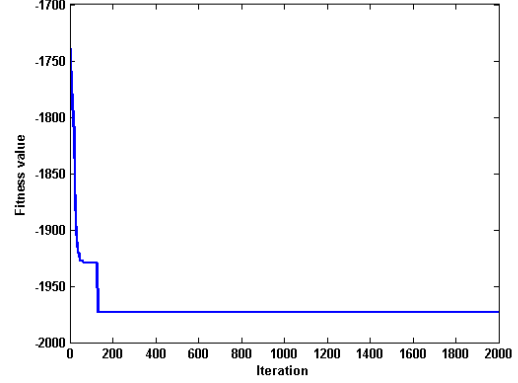
Figures 3.7 and 3.8 show the output of the multi-model representation compared with the real system output using the validation data. Table 3.8 shows the values of the performance measures. One can observe that all the methods have close

Table 3.7: Results of Stage 2 (final submodels)

	2-parameters structure		4-parameters structure			
	a_1	b_1	a_1	a_2	b_1	b_2
Final submodels						
1	0.48004	0.60620	0.46668	0.04022	0.64528	0.04504
2	0.59707	0.60308	0.49261	0.03074	0.56777	0.03306
Final centers						
1	0.01067	0.45786	-0.34414	-0.09591	-0.00413	-0.48350
2	-0.02951	-0.46991	0.29115	0.06810	-0.01360	0.41844



(a) 2-paramters structure



(b) 4-paramters structure

Figure 3.6: MCP SO objective function convergence plot

performance in this case.

3.5.3 Example 3

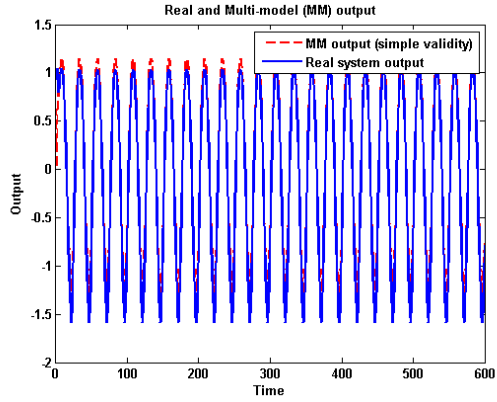
In the third example, the following highly nonlinear dynamical system is considered for identification:

$$y(k) = (y(k-1)/(1 + y(k-1)^2)) + u(k-1)^3; \quad (3.31)$$

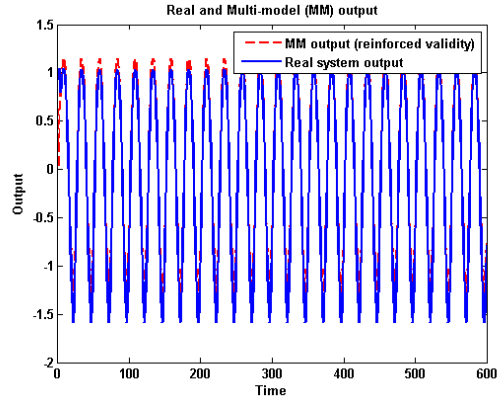
It is a benchmark system proposed in [170]. The system was excited by uniformly distributed random signal in the interval $[-1, 1]$. The identification was carried

Table 3.8: Validation performance test on common validity estimations

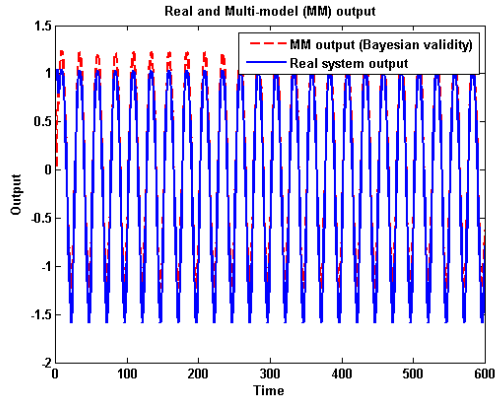
Validity Estimation	2-parameters structure			4-parameters structure		
	MSE	PMF(%)	VAF(%)	MSE	PMF(%)	VAF(%)
simple Residue	0.0221	83.89	98.39	0.0313	80.84	98.19
Reinforced residue	0.0221	83.89	98.39	0.0313	80.84	98.19
Bayessian	0.0357	79.55	98.18	0.0354	79.64	98.21
Qaudratic	0.0249	82.91	98.43	0.0303	81.17	98.26



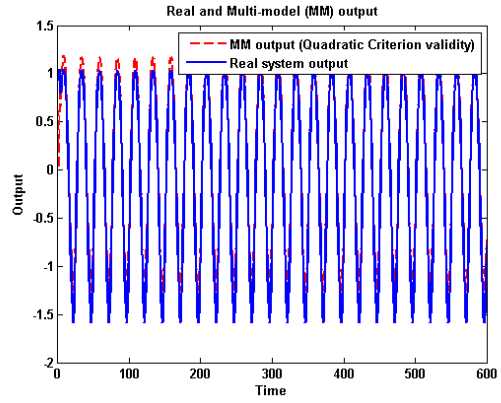
(a) 2-paramters structure with simple residue



(b) 2-parameter structure with Reinforced residue



(c) 2-parameter structure with Bayesian



(d) 2-parameter structure with Quadratic

Figure 3.7: Multi-model identification outputs using validation data

out with data set of 800 samples.

Using the proposed multi-model partition method, four and two submodels were identified for 2-parameter and 4-parameter structure respectively. The convergence of the objective function in the developed MPSO for both structures

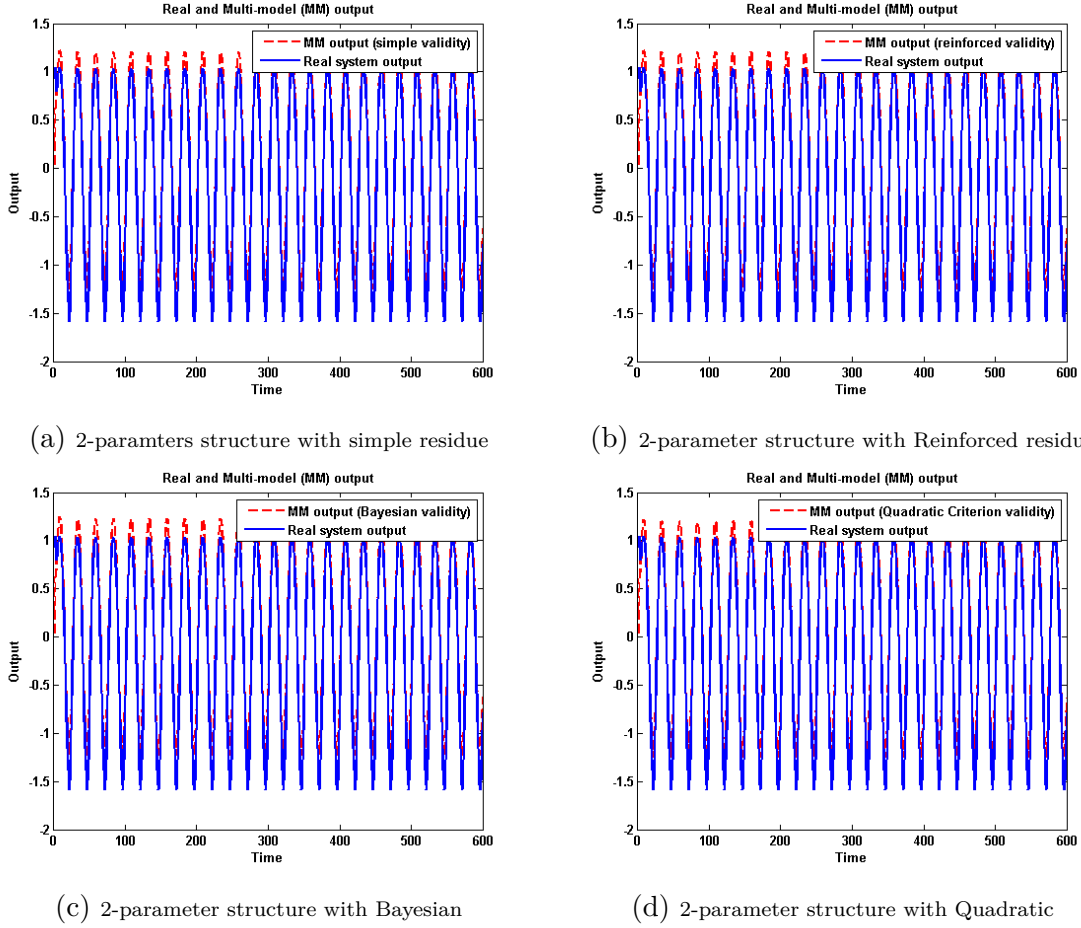


Figure 3.8: Multi-model identification outputs using validation data

are shown in Figure 3.9. The initial and final submodels' parameters with their associated centers are shown in Tables 3.9 and 3.10 respectively.

Validation of the multi-model identification was done with second data set of 500 samples generated by an input signal given by:

$$u(k) = \sin\left(\frac{2\pi}{25}k\right) + 0.2\sin\left(\frac{2\pi}{10}k\right)$$

Simulation results obtained using different validity estimation for the validation data set are shown in Figures 3.10 and 3.11. The figures show that the

Table 3.9: Results of Stage 1 (initial submodels)

Initial submodels	2-parameters structure		4-parameters structure			
	a_1	b_1	a_1	a_2	b_1	b_2
1	0.04262	2.38398	-0.08706	0.04869	2.38490	0.38678
2	0.08510	2.38842	-0.05318	0.03350	1.82057	0.21705
3	0.01367	2.18292	-	-	-	-
4	0.05480	2.03856	-	-	-	-
Initial centers						
1	0.14603	0.03785	0.11443	0.11181	0.03658	0.03908
2	0.05709	0.07277	0.28165	0.40862	0.12559	0.09584
3	0.15296	-0.01998	-	-	-	-
4	-0.084650	0.06599	-	-	-	-

Table 3.10: Results of Stage 2 (final submodels)

Final submodels	2-parameters structure		4-parameters structure			
	a_1	b_1	a_1	a_2	b_1	b_2
1	-0.0045	2.75430	-0.09400	0.07937	3.06291	0.40508
2	0.08208	3.65709	-0.09741	0.02515	1.19728	0.44751
3	-0.07079	1.82638	-	-	-	-
4	0.06738	0.74981	-	-	-	-
Final centers						
1	0.64153	0.13734	0.06135	-0.06274	0.10813	0.01715
2	-0.12759	0.06961	0.14812	0.20475	0.01336	0.05213
3	0.19983	0.03261	-	-	-	-
4	-0.03128	0.00691	-	-	-	-

estimated model outputs closely follow the system output for both submodels' structures. However, as we will show in the next chapter, the estimation can still be improved further with better validity estimation. In addition, it can be concluded from Table 3.11 that, generally, the 4-parameter structure shows the better performance.

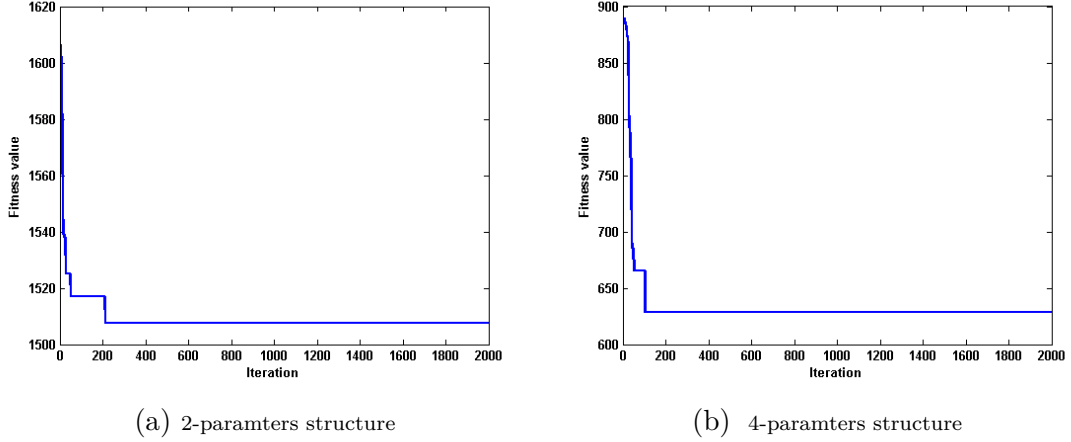


Figure 3.9: MCPSO objective function convergence plot

Table 3.11: Validation performance on common validity estimations

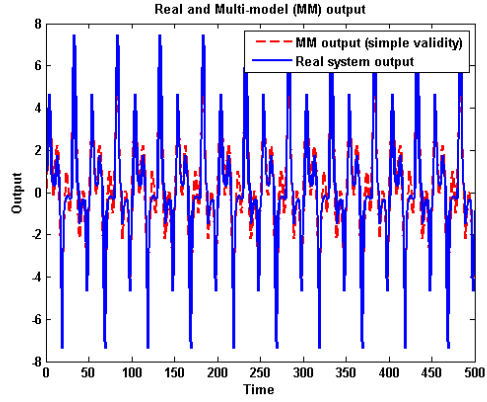
Validity Estimation	2-paramters structure			4-paramters structure		
	MSE	PMF(%)	VAF(%)	MSE	PMF(%)	VAF(%)
simple Residue	0.8224	65.75	88.28	0.4309	75.21	93.88
Reinforced residue	0.9480	63.23	86.49	0.4309	75.21	93.88
Bayessian	1.2033	58.58	82.85	1.1679	59.19	83.35
Qaudratic	1.1827	58.93	83.15	1.1657	59.23	83.38

3.5.4 Example 4

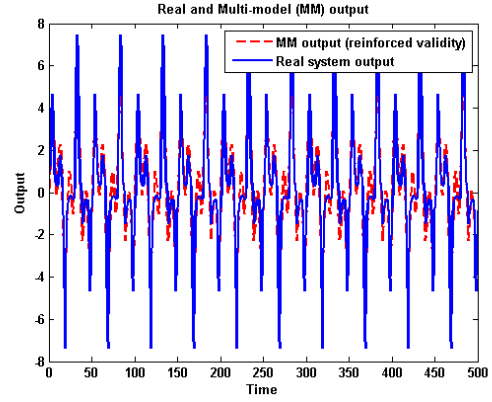
The next example considered for identification is another highly nonlinear dynamical system, also proposed in [170] as a benchmark system and has been used subsequently in [171, 126, 172, 43]. The system is described by

$$\begin{aligned}
 y(k+1) &= \frac{u(k)}{1 + y^2(k-1) + y^2(k-2)} \\
 &+ \frac{y(k)y(k-1)y(k-2)u(k-1)(y(k-2)-1)}{1 + y^2(k-1) + y^2(k-2)} \quad (3.32)
 \end{aligned}$$

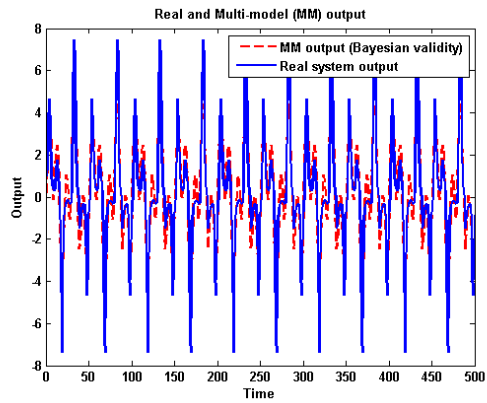
The system was excited by uniformly distributed random signal in the interval $[-1, 1]$. The identification was carried out with data set of 800 samples.



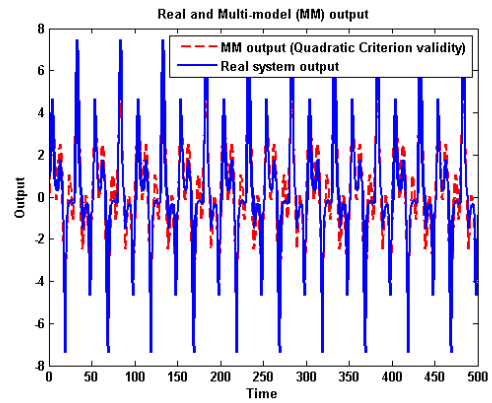
(a) 2-paramters structure with simple residue



(b) 2-parameter structure with Reinforced residue



(c) 2-parameter structure with Bayesian

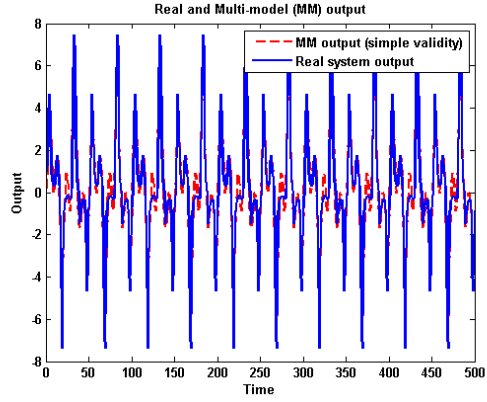


(d) 2-parameter structure with Quadratic

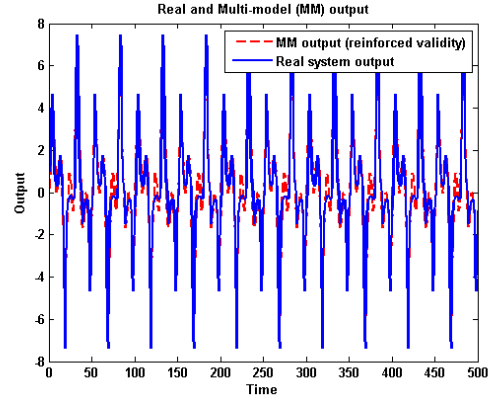
Figure 3.10: Outputs of multi-model identification using validation data

Using the proposed multi-model partition method, four and two submodels were identified for 2-parameter and 4-parameter structures respectively. The convergence of the objective function in the developed MPSO for both structures are shown in Figure 3.12. The initial and final submodels parameters with their associated centers are shown in Tables 3.12 and 3.13 respectively.

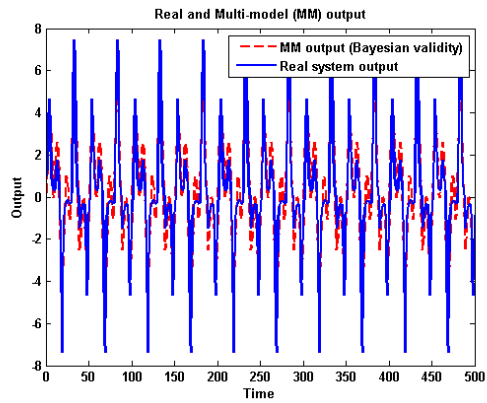
Validation of the multi-model identification was done with second data set of



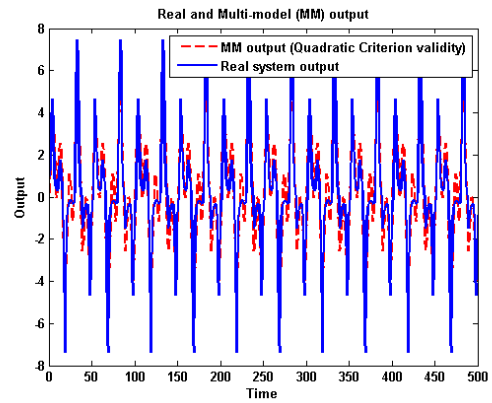
(a) 4-paramters structure with simple residue



(b) 4-parameter structure with Reinforced residue



(c) 4-parameter structure with Bayesian



(d) 4-parameter structure with Quadratic

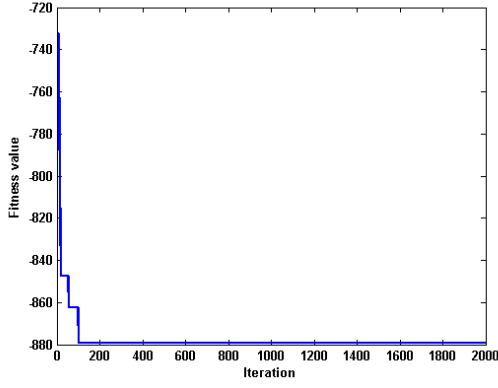
Figure 3.11: Outputs of multi-model identification using validation data

Table 3.12: Results of Stage 1 (initial submodels)

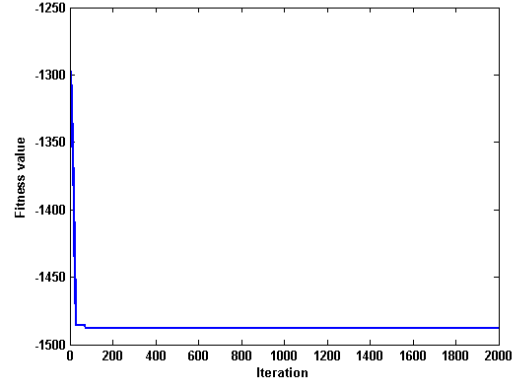
Initial submodels	2-parameters structure		4-parameters structure			
	a_1	b_1	a_1	a_2	b_1	b_2
1	-0.00935	0.74062	-0.11662	0.03551	0.74416	0.08480
2	-0.00839	0.72822	0.40609	0.01631	0.74891	-0.25511
3	-0.03470	0.79774	-	-	-	-
4	0.05112	0.80693	-	-	-	-
Initial centers						
1	-0.01530	-0.00774	-0.00143	-0.00626	-0.00842	0.00377
2	-0.00240	-0.00645	-0.14715	-0.06505	0.01370	-0.24046
3	-0.00191	0.00308	-	-	-	-
4	0.06674	-0.01483	-	-	-	-

Table 3.13: Results of Stage 2 (final submodels)

	2-parameters structure		4-parameters structure			
	a_1	b_1	a_1	a_2	b_1	b_2
Final submodels						
1	-0.01637	0.65325	-0.05385	0.03735	0.86452	0.02908
2	0.04630	0.47067	-0.09456	0.04253	0.59664	0.06760
3	-0.03507	0.87663	-	-	-	-
4	0.00081	0.87200	-	-	-	-
Final centers						
1	-0.19101	-0.05831	0.13269	0.02336	-0.00636	0.18525
2	0.20190	0.13531	-0.17305	-0.04705	-0.00841	-0.23354
3	0.09209	0.59731	-	-	-	-
4	0.02161	-0.59013	-	-	-	-



(a) 2-paramters structure



(b) 4-paramters structure

Figure 3.12: MCPSO objective function convergence plot

800 samples generated by an input signal given by:

$$u(k) = \begin{cases} \sin(\frac{2\pi}{250}k) & \text{if } k \leq 500 \\ 0.8\sin(\frac{2\pi}{250}k) + 0.2\sin(\frac{2\pi}{25}k) & \text{if } k > 500 \end{cases}$$

Simulation results obtained using different validity estimation for the validation data set are shown in Figures 3.13 and 3.14. The figures show that the estimated model outputs closely follow the system output for both submodels'

Table 3.14: Validation performance on common validity estimations

Validity Estimation	2-parameters structure			4-parameters structure		
	MSE	PMF(%)	VAF(%)	MSE	PMF(%)	VAF(%)
simple Residue	0.0037	87.52	99.00	0.0013	92.57	99.54
Reinforced residue	0.0045	86.16	98.78	0.0013	92.57	99.54
Bayessian	0.0066	83.35	98.22	0.0082	81.37	97.68
Qaudratic	0.0062	83.79	97.83	0.0110	78.47	97.91

structures. However, we shall show in the next chapter that the estimation can still be improve further with better validity estimation. In addition, it can be concluded from Table 3.14 that, generally, the 4-parameter structure shows better performance.

3.5.5 Example 5

In this example a benchmark continuous stirred tank reactor (CSTR) nonlinear chemical system is considered. The system is described by the following equations in which all variables are dimensionless [51, 173]:

$$\begin{aligned}
 \dot{x}_1 &= -x_1 + D_a \cdot (1 - x_1) \cdot \exp\left(\frac{x_2}{1 + x_2/\gamma}\right), \\
 \dot{x}_2 &= -x_2 + B \cdot D_a \cdot (1 - x_1) \cdot \exp\left(\frac{x_2}{1 + x_2/\gamma}\right) + \beta \cdot (u - x_2) \\
 y &= x_2
 \end{aligned} \tag{3.33}$$

where x_1 is the reagent conversion, x_2 is the reactor temperature and u is the coolant temperature. u and x_2 are the input and output of the system, respectively. The nominal values for the constants are $D_a = 0.072$, $\gamma = 20$, $B = 8$,

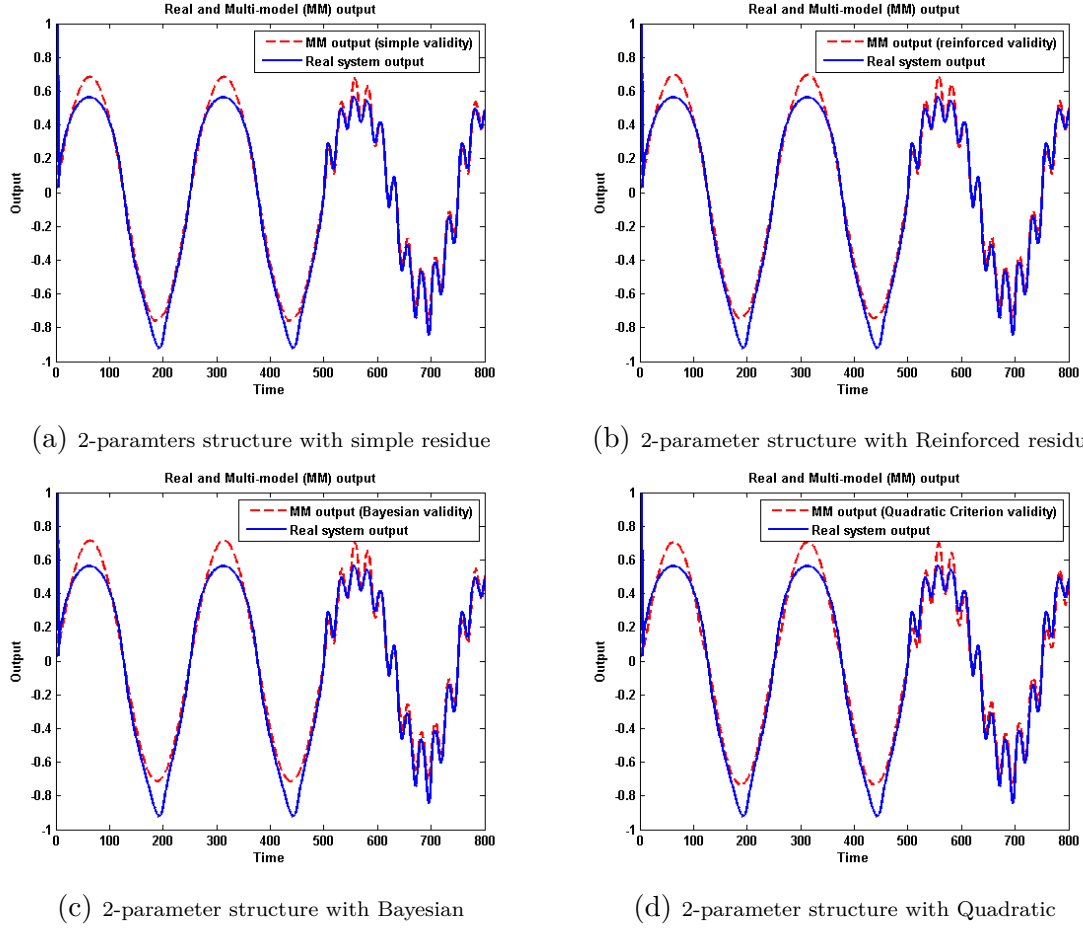


Figure 3.13: Outputs of multi-model identification using validation data

and $\beta = 0.3$. The operating range of the system is $\{y \in [0, 6]\}$. The system exhibits output multiplicity and according to [51] two stable and one unstable models can be obtained when the systems is linearized around three steady state points corresponding to $u = 0$.

To test the proposed multi-model partition method on previous validity estimations, a random white noise step signal between $[-1.5, 1.5]$ is used as input to the system. The system is simulated with a sampling time of 0.2 min, 700 pairs of input-output data were collected for the identification process and another 300 pairs for validation.

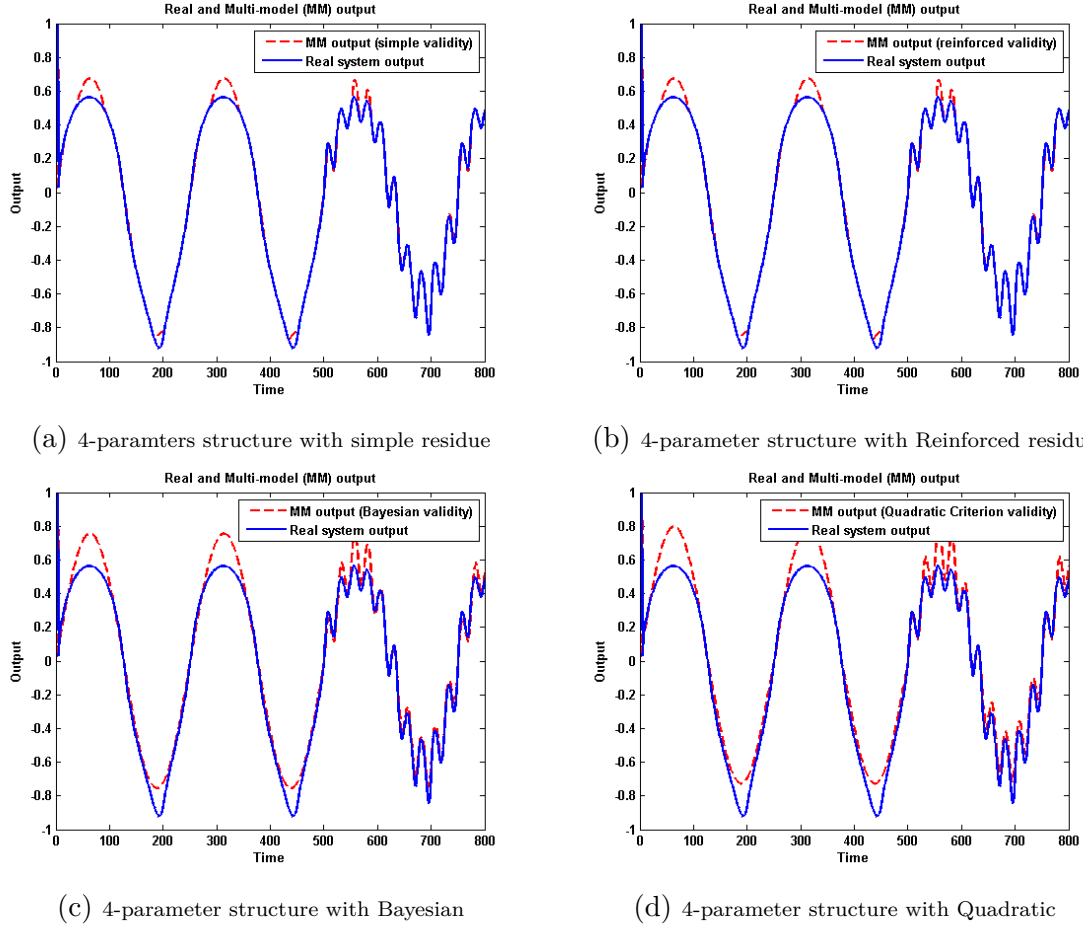


Figure 3.14: Outputs of multi-model identification using validation data

Based on the proposed method, three submodels were identified for 4-parameter structures only. For the 2-parameter structure, it was found that the MCPSO algorithm did not converge most of the time even after increasing the number of iteration to 10000. And when it converges, it does so with very poor result. Due to this, it is concluded that the 2-parameter structure is inadequate to estimate the system. The initial and final submodels parameters with their associated centers are shown in Tables 3.15 and 3.16 respectively. The convergence of the objective function in the developed MPSO is shown in Figure 3.15. Notice also that submodel 1 is unstable model for the 4-parameter structure.

Table 3.15: Results of Stage 1 (initial submodels)

	4-parameters structure			
	a_1	a_2	b_1	b_2
Initial submodels				
1	0.41435	0.28499	-0.13038	-0.45585
2	0.38424	0.15863	0.25568	0.25428
3	0.33236	0.48919	0.05200	-0.11514
Initial centers				
1	0.88647	0.97696	0.00436	-0.18552
2	1.02903	0.99540	0.03286	0.09124
3	0.99887	0.97750	0.04890	0.10564

Table 3.16: Results of Stage 2 (final submodels)

	4-parameters structure			
	a_1	a_2	b_1	b_2
Final submodels				
1	0.52332	0.60225	-0.79173	-1.45343
2	0.37017	0.05227	1.00263	1.22693
3	0.30684	0.37524	-0.15508	0.10070
Final centers				
1	0.84523	1.00344	0.02873	-0.07783
2	1.10918	0.90721	0.14359	0.18742
3	0.98968	1.03175	-0.05245	-0.00450

The validation results using the second data pairs are shown in Figure 3.16 and Table 3.17 for 4-parameter structures. It can be observed that only interpolation of the submodels with simple residue validity computation can estimate the system while the other three validity computations were unstable. This might be as a result of one unstable model in the identified submodels. Subsequently in the next chapter, it shall be shown that the estimation result can be improved.

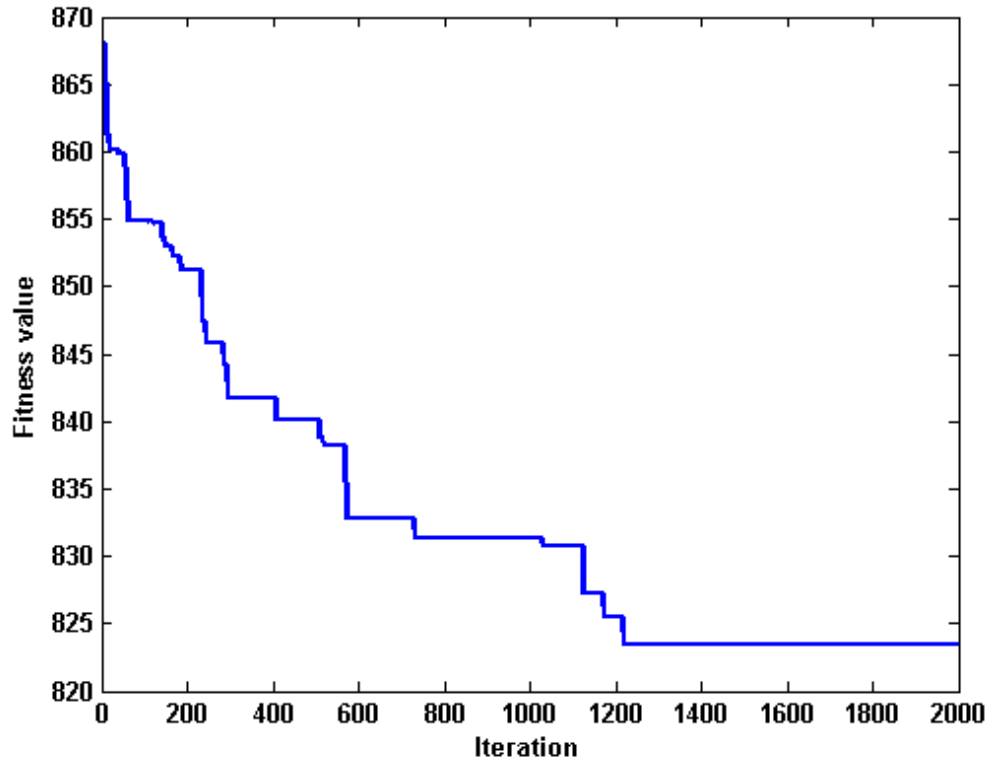


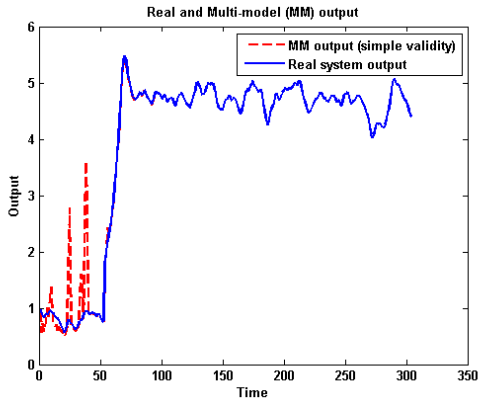
Figure 3.15: MCPSO objective function convergence plot for 4-paramters structure

3.6 Conclusion

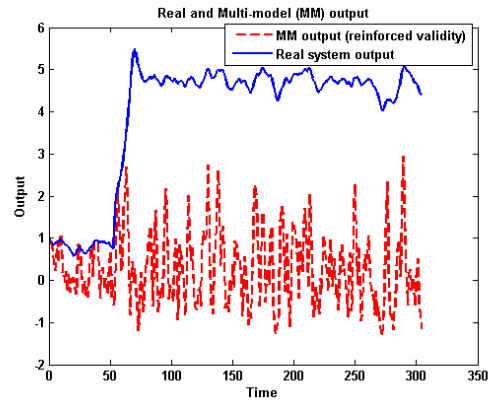
This chapter presents a novel meta-heuristic partition method for multi-model identification of nonlinear systems. In the proposed approach the number as well as the structure of the submodels are not known a priori. The proposed method consists of two stages. In the first stage, an initial estimate of the number of submodels and their parameters are obtained. The final submodels are obtained in the second stage. Four simulated nonlinear systems examples that had been studied previously in the literature are used to illustrate the performance of the method under different validity estimation methods in the literature for combining

Table 3.17: Validation performance test on common validity estimations

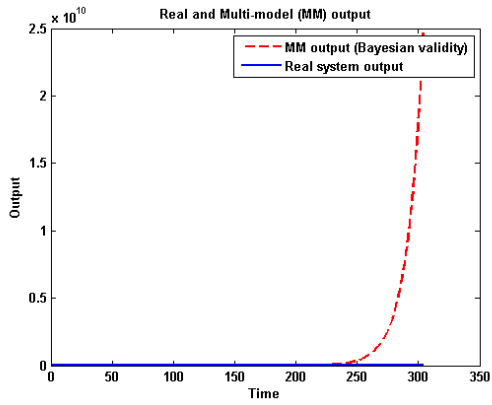
Validity Estimation	4-parameters structure		
	MSE	PAF(%)	VAF(%)
simple Residue	0.0570	84.23	97.53
Reinforced residue	17.0351	0	0
Bayessian	1.38E+19	0	0
Qaudratic	1.31E+19	0	0



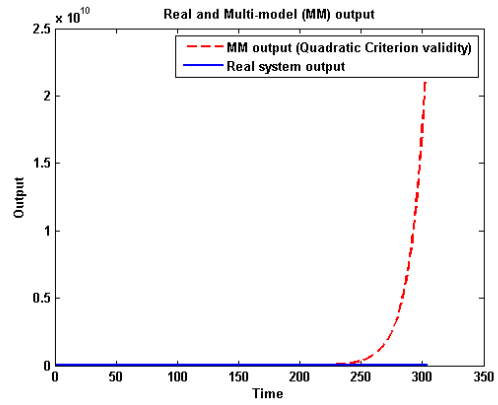
(a) 4-paramters structure with simple residue



(b) 4-parameter structure with Reinforced residue



(c) 4-parameter structure with Bayesian



(d) 4-parameter structure with Quadratic

Figure 3.16: Outputs of multi-model identification using validation data

the submodels generated. In the next chapter we shall introduce another validity estimation to improve on the performance obtained by these validity estimation methods.

CHAPTER 4

CONSTRAINED KALMAN FILTER FOR VALIDITY ESTIMATION

Another important challenge in multimodel framework is the computation of the validity associated with each sub-models. Validity computation is crucial to the correct identification of the system at hand in order to optimize its performance. In addition, it is a key decision making in multi-model-based fault diagnostic. Chapter 4 proposes a constrained Kalman Filter for the estimation of validity of submodel in multi-model framework. This is achieved by reformulating the multi-model output equation as an estimation problem. Simulation examples used in the previous chapter are utilized to illustrate the effectiveness of the proposed validity computation and compared to other commonly used ones.

4.1 Introduction

Multi-model frame is an appealing that composes of three steps: the first is the partitioning of the operating system into smaller regions based on a selected strategy. In the second step, both the structure and the parameters of the local model associated with each subregion are determined. Finally, the submodels are combined together by weight that defines the contribution of each local model to the real system. This signifies that a suitable validity estimation method needs to be added to the method for generating submodels developed in the chapter 2 to successfully form the multi-model representation. Although in the simulations given in chapter 2, some validity estimation methods were used to combined the submodels. Could these methods be the best for the partition method developed? The choice of the validity computation plays a crucial role in the accuracy of the multi-model identification approach [133, 45]. Due to this importance, various types of validity computations have been proposed in the literature.

One of the most commonly used validity computations is Gaussian function [24, 75, 88, 111, 124] due its smoothness property. However, determination of its center and width, which both affect the accuracy and interpretation of the identified model, is quite challenging. In [103, 142], sigmoid function is used as validity computation in axes-oblique partitioning algorithm. A nonlinear optimization technique is required to optimize its variables. Residue approach [143, 110, 109, 117] is another validity computation that relies on the computation of the distance between the current output of the system and that of the local

models. Simple and reinforced are the most commonly used of residue approaches. The two approaches, though simple, lack precision and are not recommended for use in complex and ill-defined systems [144]. Other validity based on residue approach such as minimization of quadratic criterion [144, 118] can only be used with clustering partition, Bayesian validity [146, 145] also lack precision, and Neural networks [119] and fuzzy logic [147] required special design for each submodel.

In this chapter, constrained Kalman filter (CKF) is developed for validity computation of output blended multi-model systems identification of nonlinear process systems. This is achieved by reformulating the multi-model output equation as an estimation problem. The method overcomes some of the drawback of the validity computations mentioned previously such as lack of precision, sensitivity to parameter selection, and restriction to partition strategy. Previous examples used in chapter 3 are re-investigated with the developed CKF interpolation algorithm and its performance is compared with other validity computation methods used in the chapter.

The rest of this chapter is organized as follows: section 4.2 describes the problem formulation. In section 4.3 Constrained Kalman Filter validity computation is developed to the problem at hand. In section 4.4, simulation results are provided to illustrate the performance of the proposed method. Finally a brief conclusion is given in section 4.5.

4.2 Problem Formulation

Consider the previous nonlinear system again:

$$y(k) = F(y(k-1), y(k-2), \dots, y(k-na), u(k-1), u(k-2), \dots, u(k-nb)) \quad (4.1)$$

where $u(k) \in \mathfrak{R}^{nb}$ is the input and $y(k) \in \mathfrak{R}^{na}$ is the output of the system. The integer nb and na are the time lag of the input and output respectively. The multi-model representation of the system can be describe by

$$y(k) = \sum_{i=1}^m f_i(x(k))\phi_i(k) \quad (4.2)$$

where m is the number of submodels, $f_i(\cdot)$ and $\phi_i(k)$ are the i^{th} submodel and validity function, respectively. The validity function describes the contribution of each submodel to the observed output and allows smooth transition between the local models when the system moves from one operating point to another. For easy interpretation, the validity function satisfies the convexity property [132, 159]:

$$\sum_{i=1}^m \phi_i(k) = 1 \quad \forall k \quad (4.3)$$

$$0 \leq \phi_i(k) \leq 1 \quad \forall k, \quad \forall i \in 1, \dots, m \quad (4.4)$$

Given a set of input-output data, the multimodel identification problem is to estimate (i) number of submodels (m), (ii) the parameters of each submodel (local model) ($f(\cdot)$) and (iii) the validity function (ϕ) for each submodel.

As described previously, in output blended multi-model representation of complex systems, the entire operating space of the system is partitioned into a number of operating region. For each operating region a local model is associated to depicts the behavior of the system within that region. A weighted sum of the output of the local models is then used to form the global output of the systems. In chapter 3, we have been able to find m , and $f(\cdot)$. In this chapter estimation of ϕ is our paramount concern. This will be achieved by constrained Kalman Filter (CKF).

4.3 Constrained Kalman Filter (CKF) for Validity Computation

4.3.1 Constrained Kalman Filter (CKF)

Consider a controllable and observable linear discrete time system of the form

$$\begin{aligned}x(k+1) &= F(k+1)x(k) + w(k) \\ y(k) &= H(k)x(k) + v(k)\end{aligned}\tag{4.5}$$

where $F(k+1)$ is the transition matrix, $x(k)$ is the state at time instant k , $H(k)$ is measurement matrix, and $y(k)$ is the measurement data at time k . $w(k)$ and $v(k)$ are respectively the process and measurement noise assumed to be white and Gaussian, with zero mean and covariance matrix Q and R . The state estimation problem is stated as using the entire observed data, consisting of the

vector $y(1), y(2), \dots, y(k)$ to find for each $k \geq 1$ the minimum mean square error estimate of the state $x(k)$ [174]. The Kalman Filter equations are given as follows:

$$\begin{aligned}\hat{x}^-(k) &= F(k)\hat{x}^-(k-1) \\ P^-(k) &= F(k)P^-(k-1)F^T(k) + Q(k-1) \\ \hat{x}(k) &= \hat{x}^-(k) + K(k)[y(k) - H(k)\hat{x}^-(k)] \\ K(k) &= P^-(k)H^T(k)[H(k)P^-(k)H^T(k) + R(k)]^{-1} \\ P(k) &= [I - K(k)H(k)]P^-(k)\end{aligned}\tag{4.6}$$

where the filter is initialized with $\hat{x}(0) = \mathbb{E}[x(0)]$ and $P(0) = \mathbb{E}[(x(0) - \hat{x}(0))^T(x(0) - \hat{x}(0))]$ and $\mathbb{E}[\dots]$ denotes the expectation operator.

Suppose there are linear constraints on the state of system such that given the system (4.5), the following constraints are given :

$$Ax(k) = a\tag{4.7}$$

$$Bx(k) \leq b\tag{4.8}$$

where A and B are known matrix of dimension $s \times n$, s is the number of constraints and n is the number of state variables. a and b are known vectors. That is, the state estimate, \hat{x} , is required to satisfies the equality constraint (4.7) and the inequality constraint (4.8).

Several approaches have been presented in the literature for incorporating the equality constraint to the Kalman filter equation (4.6). Model reduction approach

[175] reduces the equality constrained problem to unconstrained one by system model parameterization. A perfect measurement [176, 177] is another approach where the equality constraint, taken as zero measurement noise, is augmented with the measurement equation. Another popular approach is the projection of the unconstrained estimate of the filter on the constraint surface [178]. Yet another method is the modification of standard Kalman filter gain by projecting it to the constraint surface [179, 180]. Other approaches include, projected system representation, [181], and the use of descriptor system theory [182]. Although these methods are fundamentally different, they are mathematically equivalent [183].

Concerning the inequality constraint (4.8), some of the approaches for the equality constraint problem have been adapted. These include state estimate projection [184] and gain projection [185]. Probability Density Function (PDF) truncation [186] is another methods where the PDF of the constraint edges is truncated to update the filter. Yet another approach is to truncate the state estimate into the feasible state [187]. This is a simple approach that avoid the quadratic program problem of the state estimation and the complication that arises from PDF method [188]. A more detailed survey of these methods can be found in [188]. In what follows, a CKF approach is proposed for multi-model weights computation, using the projection and truncation methods for the equality and inequality constraints, respectively.

4.3.2 Constrained Kalman Filter (CKF) for Validity Estimation of local models

Kalman filter as a well known optimal state estimator can be reformulated as a parameter estimation problem (e.g. [174, 189]). Therefore, reformulating the validity in (2.2) as a parameter estimation problem can be solved by Kalman filter as follows:

Given the multi-model output in Equation (2.2)

$$y(k) = \sum_{i=1}^M f_i(\varphi_i(k), \Theta_i)\phi_i(k)$$

Taking the vector form of Equation (2.2) is written as:

$$y(k) = \bar{y}(k)\Phi(k) \tag{4.9}$$

where $\bar{y} = [y_1, y_2, \dots, y_M]$ is the known vector of local model outputs and $\Phi = [\phi_1, \phi_2, \dots, \phi_M]^T$ is the unknown vector of weights for the local models.

Casting the parameter estimation problem to state estimation problem, we have the following state estimation equation :

$$\Phi(k+1) = \Phi(k) \tag{4.10}$$

$$y(k) = \bar{y}(k)\Phi(k) + v(k)$$

where $\Phi(k)$ is the vector of unknown parameter (validity) to be estimated, and $v(k)$ is the measurement noise with covariance $Ev^2 = R(t)$, as in equation (4.5). In the case where $v(k)$ is a white and gaussian, the Kalman filter theory says that the posterior distribution of $\phi(k)$, given all the observation up to $k-1$, is gaussian mean value $\hat{\phi}(k)$ and covariance matrix P . One should note that an artificial noise $w(k)$, with variance $EW(k)w^T(k) = Q$ could be added to $\Phi(k+1)$ in the case of time-varying parameters and also to ensure persistence excitation and avoid ill conditioned numerical computation.

Furthermore, in order for the validity computation (Φ) to satisfy the partition of unity, the equality constraint in (2.3a) need to be added. Also, since at any time instant it is possible for any local model to fully contribute or not to contribute to the system's output, there is a need to impose an inequality constraint (2.3b). Therefore, these two constraints need to be included in the estimation of (Φ) to give the following full state estimation equation:

$$\begin{aligned}\Phi(k+1) &= \Phi(k) + w(k) \\ y(k) &= \bar{y}(k)\Phi(k) + v(k)\end{aligned}$$

such that (4.11)

$$\begin{aligned}\beta\Phi(k) &= 1 \\ 0 &\leq \phi_i(k) \leq 1\end{aligned}$$

where β is a row vector of $[1, 1, \dots, 1, 1]$. The problem is thus formulated as giving

a state equation in (4.10), minimize the minimum mean square error estimate of the state $\Phi(k)$.

$$\begin{aligned} & \underset{\Phi}{\text{minimize}} \quad \mathbb{E}[(\Phi(k) - \hat{\Phi}(k))^2] \\ & \text{such that} \end{aligned} \tag{4.12}$$

$$\beta\Phi(k) = 1$$

$$0 \leq \phi_i(k) \leq 1$$

Where \mathbb{E} is the expectation operation, β is a row vector of $[1, 1, \dots, 1, 1]$, Φ is the unknown parameter and $\hat{\Phi}$ is the estimated one.

The above problem can be solved in two steps. In the first step, the equality constraint is solved using the projection techniques [178, 190], where the unconstrained estimate $\hat{\Phi}(k)$ is projected onto the constraint space. The equality constrained optimization problem can be written as

$$\underset{\Phi}{\text{minimize}} \quad \mathbb{E}[(\Phi(k) - \hat{\Phi}(k))^T W (\Phi(k) - \hat{\Phi}(k))] \tag{4.13}$$

such that

$$\beta\Phi(k) = 1 \tag{4.14}$$

where β is a row vector of $[1, 1, \dots, 1, 1]$, and $\Phi(k)$ is $[\phi_1(k), \phi_2(k), \dots, \phi_M(k)]^T$

and W is a positive definite matrix. The solution to this problem is given as

$$\begin{aligned}
\hat{\Phi}^*(k) &= \hat{\Phi}(k) + K^*(k)[1 - \beta\hat{\Phi}(k)] \\
K^*(k) &= W^{-1}\beta^T[R + \beta W^{-1}\beta^T]^{-1} \\
P^*(k) &= [I - K^*(k)\beta]W^{-1} + Q
\end{aligned} \tag{4.15}$$

where $\hat{\Phi}$ is the unconstrained estimate, $\hat{\Phi}^*(k)$ is the updated equality constrained estimate that satisfy (2.3a) and W is a positive definite matrix weight. Setting $W = P^{-1}(k)$ in (4.15) results in minimum variance estimate and setting $W = I$ gives least square estimate of $\Phi(k)$ [188]. Both settings are implemented in this study. This implies that the unconstrained problem is first solved with standard solution of kalman filter after which the obtained unconstrained estimate, $\hat{\Phi}$ is used to update the constrained estimate in (4.15). Given observations $y(k), y(k-1), \dots, y(1)$ and local model outputs $\bar{y}(k), \bar{y}(k-1), \dots, \bar{y}(1)$ the optimal unconstrained estimate, $\hat{\phi}$, can be computed using the following Kalman filter equation

$$\begin{aligned}
\hat{\Phi}^-(k) &= \hat{\Phi}^-(k-1) \\
P^-(k) &= P^-(k-1) + Q(k-1) \\
\hat{\Phi}(k) &= \hat{\Phi}^-(k) + K(k)[y(k) - \mathbf{y}(k)\hat{\Phi}^-(k)] \\
K(k) &= P^-(k)\mathbf{y}^T(k)[\mathbf{y}(k)P^-(k)\mathbf{y}^T(k) + R(k)]^{-1} \\
P(k) &= [I - K(k)\mathbf{y}(k)]P^-(k)
\end{aligned} \tag{4.16}$$

Lastly, in the second step truncation and normalization [187] are adopted for the inequality constraints. The truncation is used to readjust each element of $\hat{\Phi}^*(k)$ in order not to violate the inequality constraint in (4.12) as follows.

$$\hat{\phi}_i^{**}(k) = 0 \text{ If } \hat{\phi}_i^*(k) < 0$$

Finally, $\hat{\phi}^{**}(k)$ is normalized since the truncation can violate the equality constraint in (2.3a) and to satisfy the other part of the inequality constraint.

$$\hat{\phi}_i^{***}(k) = \frac{\hat{\phi}_i^{**}(k)}{\sum_{i=1} \hat{\phi}_i^{**}(k)} \quad (4.17)$$

$\hat{\Phi}^{***}(k) = [\hat{\phi}_1^{***}(k), \dots, \hat{\phi}_M^{***}(k)]^T$ is the final estimated validity computation at time k . The summary of CKF algorithm is shown in Table 4.1.

4.3.3 Estimating Q and R in CKF algorithm

As it will be observed in the next section, the values of R and Q in the CKF algorithm can influence the estimation of the models' validity computation. This has been well-known drawback of Kalman filter as the process and measurement noise statistic are generally not known. Therefore R and Q are often considered as turning parameters. Since doing this manually can constitute a considerable burden, there is need to find a systematic way of estimating these parameters. Although several methods such as Bayesian [191], fuzzy logic [192], genetic algorithm [193], neural networks [194], self turning [195], autocovariance[196, 197, 198], etc,

Table 4.1: Validity estimation of submodels

<p>1: Initialize $\Phi = [\phi_1, \dots, \phi_m]$, P Require: Q, R 2: Compute unconstrained estimate $\hat{\Phi}$ of Φ</p> $\hat{\Phi}^-(k) = \hat{\Phi}^-(k-1)$ $P^-(k) = P^-(k-1) + Q(k-1)$ $\hat{\Phi}(k) = \hat{\Phi}^-(k) + K(k)[y(k) - \mathbf{y}(k)\hat{\Phi}^-(k)]$ $K(k) = P^-(k)\mathbf{y}^T(k)[\mathbf{y}(k)P^-(k)\mathbf{y}^T(k) + R(k)]^{-1}$ $P(k) = [I - K(k)\mathbf{y}(k)]P^-(k)$ <p>3: Compute equality constrained estimate, $\hat{\Phi}^*$ of Φ</p> $\hat{\Phi}^*(k) = \hat{\Phi}(k) + K^*(k)[1 - \beta\hat{\Phi}(k)]$ $K^*(k) = W^{-1}\beta^T[R + \beta W^{-1}\beta^T]^{-1}$ $P^*(k) = [I - K^*(k)\beta]W^{-1} + Q$ <p>4: Truncation of $\hat{\Phi}^*(k)$</p> $\hat{\phi}_i^{**}(k) = 0 \text{ If } \hat{\phi}_i^*(k) < 0$ <p>5: Finally normalized $\hat{\phi}^{**}(k)$</p> $\hat{\phi}_i^{***}(k) = \frac{\hat{\phi}_i^{**}(k)}{\sum_{i=1} \hat{\phi}_i^{**}(k)}$ <p>6: The final estimated validity computation at k is 7: $\hat{\Phi}^{***}(k) = [\hat{\phi}_1^{***}(k), \dots, \hat{\phi}_m^{***}(k)]^T$</p>
--

have been proposed in the literature for turning R and Q in relation to Kalman filter, there is still need to estimate these parameters in the context of constrained Kalman filter for validity estimation. Although estimating R is not critical as it can be chosen by taken the variance of the measurement or using the sensor characteristics, however, the tuning of the process noise covariance Q is considered to be critical.

Now, if Q is considered has the parameter that incorporate the modeling errors and uncertainties as well as noises affecting the process [199], then it value can be computed as a factor due to the submodel estimation and that due to the final output estimation during the interpolation.

As for the one due to the submodel estimation, since each submodel were estimated from the least square criterion (i.e. Equation 3.23), then it can be estimated from data by computing the empirical covariance from the classical result in least square theory [1]

$$q_{ai} = \left\| \frac{1}{n_i - (d + 1)} \sum_{j=1}^{n_i} (y_j - \hat{y}_j) \sum_{j=1}^{n_i} [\varphi_j \varphi_j^T]^{-1} \right\| \quad (4.18)$$

where φ_j are the vector of the regressors for i^{th} model and n_i is the number of data points used for estimating the i^{th} model. d is the number of parameters, y and \hat{y} are the actual and predicted output, respectively. The first part of estimating Q is therefore

$$Q_a = \begin{bmatrix} q_{a1} & 0 & \dots & 0 \\ 0 & \ddots & \dots & 0 \\ 0 & 0 & \dots & q_{am} \end{bmatrix} \quad (4.19)$$

Using the idea in [195], although with different rationale, the second part can be computed from the model error taken at every time instant k . From the

unconstrained Kalman filter solution in Equation 4.16 this would be

$$q_b(k) = \hat{\Phi}(k) - \hat{\Phi}^-(k) = K(k)[y(k) - \mathbf{y}(k)\hat{\Phi}^-(k)] \quad (4.20)$$

Translating into the constrained form

$$q_b(k) = \hat{\Phi}^{***}(k) - \hat{\Phi}^{***}(k-1) \quad (4.21)$$

Unlike in [195], each component of q_b represents lack of accuracy due to a sub-model. Hence, for all the submodels, a diagonal matrix, $Q_b(k) = \text{diag}[q_b(k)^2]$, is constructed.

Both Q_a and Q_b can be seen as the confidence associated with estimating the parameters and the validities of the models. Following the two components above, the time varying covariance matrix $Q(k)$ is estimated as

$$Q(k) = Q_a + Q_b(k) \quad (4.22)$$

4.4 Simulation Examples

The effectiveness of the CKF validity computation is demonstrated in this section. In the first subsection, the suitability of CKF algorithm as validity computation is tested with the two settings of $W = P^{-1}(k)$ and $W = I$. In the second subsection, the algorithm is tested on previously used examples in chapter 3 and compared with the commonly used validity computations used in that chapter.

In the implementation of the CKF algorithm, setting $W = P^{-1}(k)$ and setting $W = I$ are both utilized. All simulations are performed using MATLAB 2012b on a 2.4 GHZ i3 64-bits Windows machine with 4 G RAM.

4.4.1 Case 1: Suitability of CKF Algorithm as Validity Computation

Here the suitability of CKF algorithm is tested by considering an arbitrary non-linear systems. Note that the aim here is not to identify the system but to show by simulation that CKF is well suited as validity. In fact, this can be considered as a convergence test for the CKF algorithm. Therefore either static or dynamic, linear or nonlinear function can be used. The actuator dynamics input-output data [88] are used in this case.

The system output is segmented sequentially into three. Each segment represents the system output data for a particular time duration. The first segment is the system output for the duration 1 to 300, the second and the third segments are for duration 301 to 600 and 601 to 1024 respectively. The three segments are used to form three separate outputs such that random values are assigned to the time duration for which there are no original output value. For example, in the first segment random values are assigned to time duration 301 to 1024. Fig. 4.1 shows the real system output and the segmented outputs. These three outputs are then run simultaneously and combined together using the two CKF validity computation, setting $W = P^{-1}(k)$ and $W = I$. For acceptable performance it

is expected that the CKF combined output will equal that of the system output with minimum error. Furthermore, the profile of CKF should reach unity only within the time duration for which each segment has the real system output value. This will indicate that CKF is ignoring the random signal of a particular segment output and converging to the true output of the system from another segment.

The simulation results shown in Fig. 4.2 indicates that the two settings of proposed CKF algorithm can adequately be used for validity computation as the validity values of each segment in Fig. 4.2c and Fig. 4.2d reach unity only within the time duration corresponding to that of original system output and tending to zero outside the duration. This characterized the suitability of the CKF algorithm as validity computation.

However, it is observed that the two settings are not of the same accuracy. It is found that setting $W = I$ took 9 seconds each to converge to the true output in the second and third segments, while setting $W = P^{-1}(k)$ took 8 seconds each. In addition, we can observed from Fig. 4.2b that setting $W = P^{-1}(k)$ does not response quickly to change in the output compared to setting $W = I$, causing more error at the point of switching. This is due to the way the segments are generated as the random number added to the segments is far away from the real output. Also, the validity profiles (Figures4.2c and 4.2d) show that setting $W = I$ is less sensitive to initialization of the validity values.

Furthermore, to determine the effect of Q and R on both CKF settings, different values of Q and R are used. It is observed that the setting $W = P^{-1}(k)$

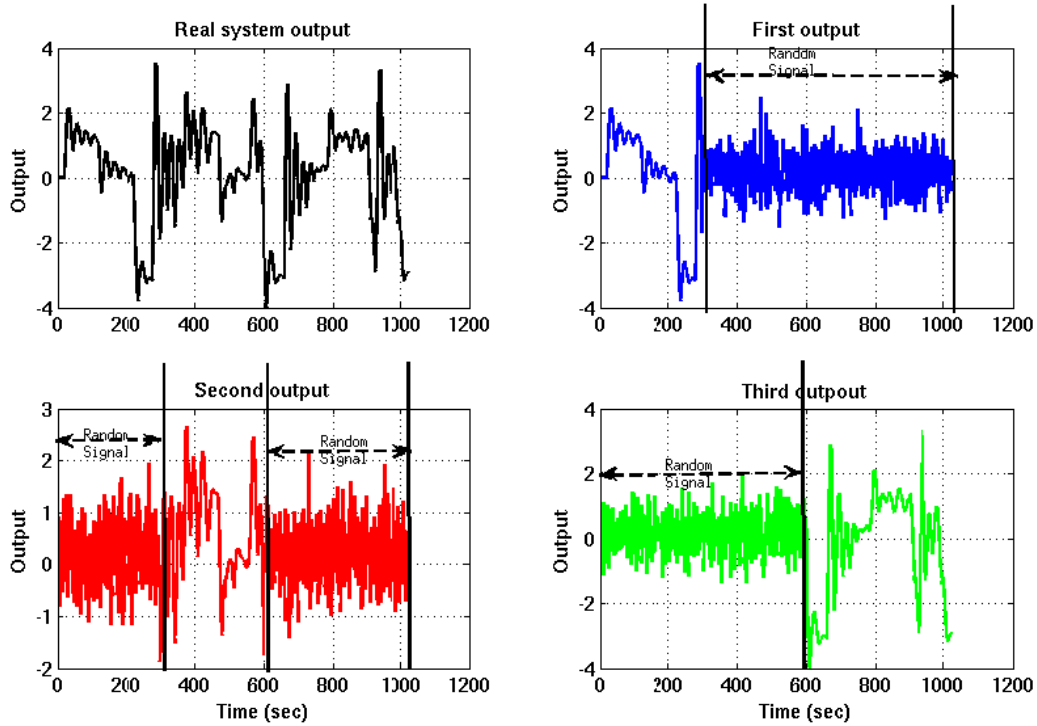
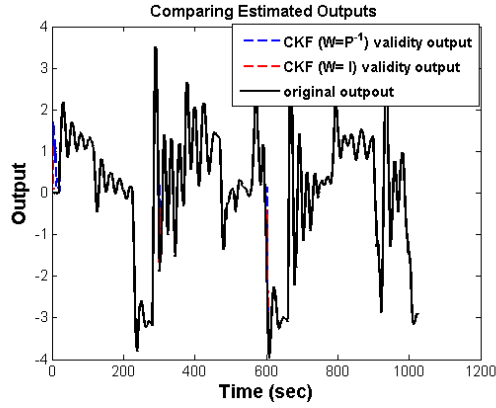


Figure 4.1: Actuator Output and segmented outputs

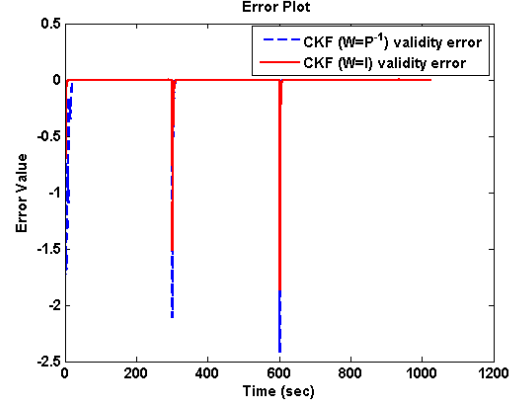
became numerically unstable when Q is zero and also slightly sensitive to choice R . Setting $W = I$ is however remain numerically stable with $Q = 0$ and less sensitive to the value R .

In addition, another system output shown in Figures4.3 [109] using similar segmentation procedure is considered. The first segment in this case is the system's output for duration 1 to 160. The second and the third segments are respectively duration 161 to 320 and 321 to 500.

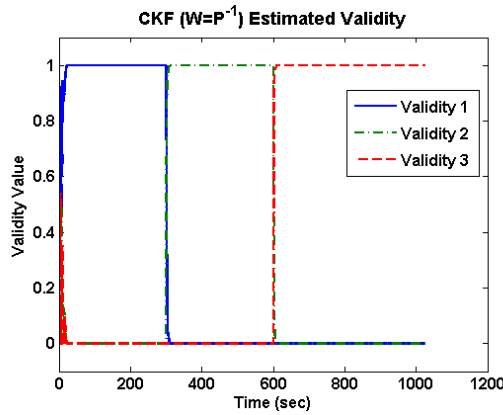
The result is as shown in Figure4.4. One can observe that setting $W = I$ took 7 seconds and 19 seconds to converge to the true output in the second and third segments respectively, while setting $W = P^{-1}(k)$ took 4 seconds each to converge in both segments. This actually confirmed faster convergence of setting $W =$



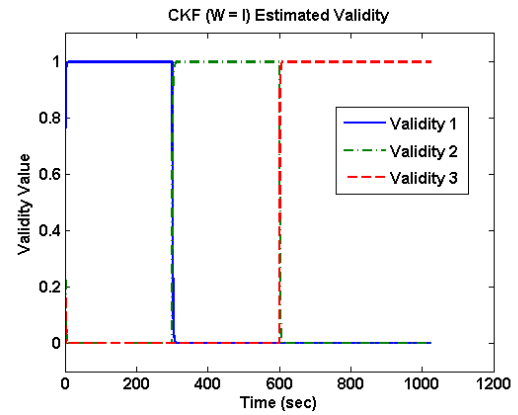
(a) The combined segment outputs with CKF



(b) The combined segment Error with CKF



(c) CKF Profile setting $W = P^{-1}(k)$



(d) CKF Profile setting $W = I$

Figure 4.2: Estimated combined output and validity profiles

$P^{-1}(k)$ as observed in the previous case. Furthermore, it is observed that there are less error at the point of switching, which suggests closer segment outputs, making the setting $W = P^{-1}(k)$ to give better output than setting $W = I$. Although the former is still less sensitive to initialization as in the previous simulation.

To summarize, the CKF algorithm has shown to be well suitable as validity computation. It is generally observed that setting $W = P^{-1}(k)$ converge faster than setting $W = I$ but sensitive to initialization, the value of Q and R . Furthermore, setting $W = P^{-1}(k)$ would be more appropriate when the output of the sub-models are closer.

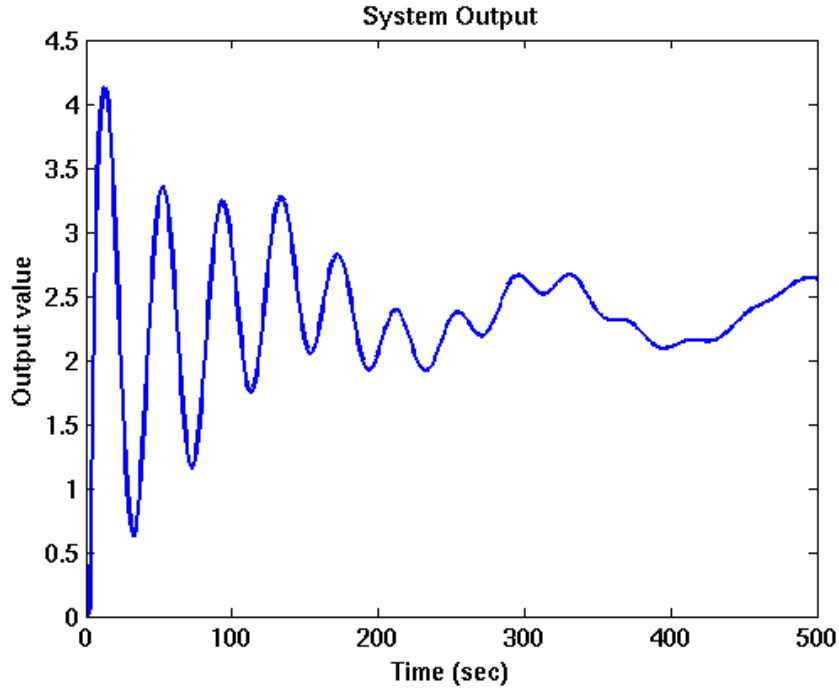


Figure 4.3: System Output

4.4.2 Case 2: Multi-model Identification

In this subsection, the developed CKF validity estimation is used for interpolation of the submodels developed for examples in the chapter 3. The output blended multi-model framework with CKF validity computation is shown in Figure 4.5. The obtained multi-model is evaluated based on the validation data using the mean square error (MSE), percentage model fitness (PMF) and variance-accounted-for (VAF) performance measures as described in chapter 3. The results obtained using the CKF algorithm is compared with other validity estimations used in chapter 3.

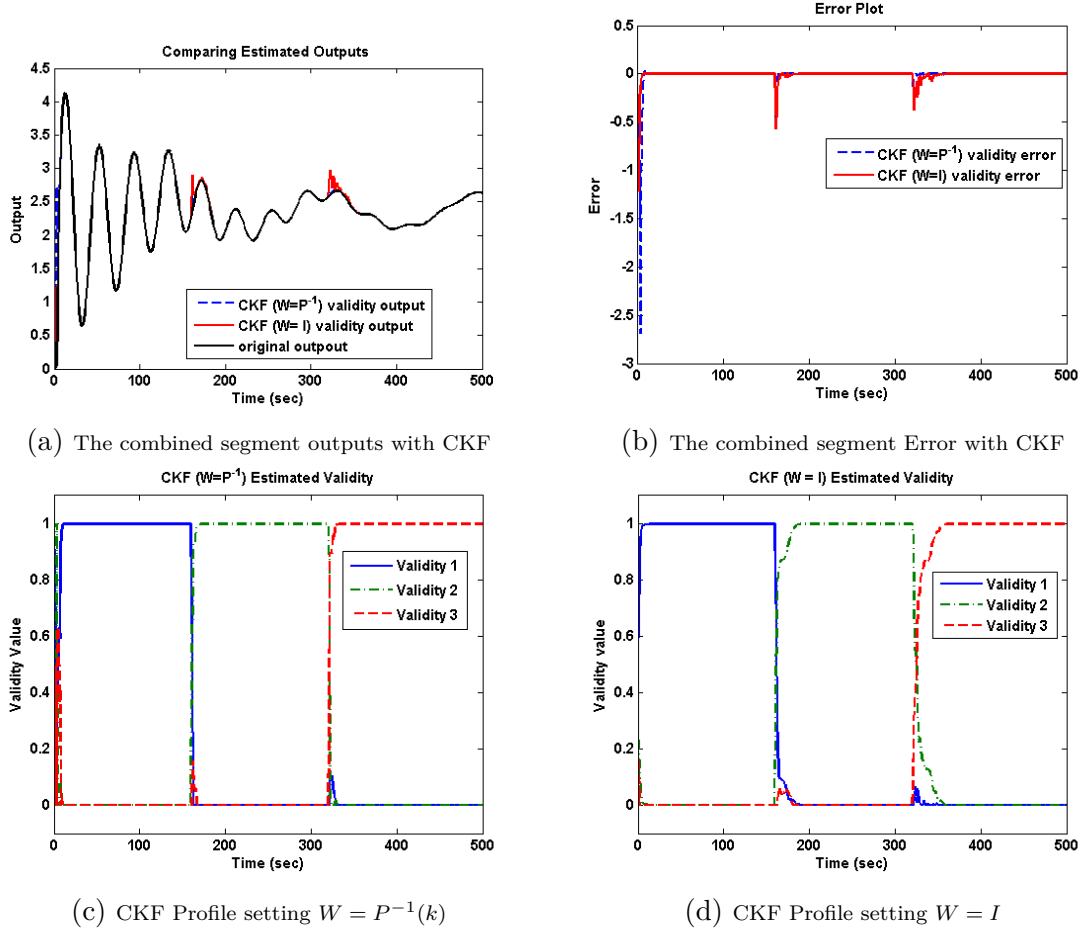


Figure 4.4: Estimated combined output and validity profiles

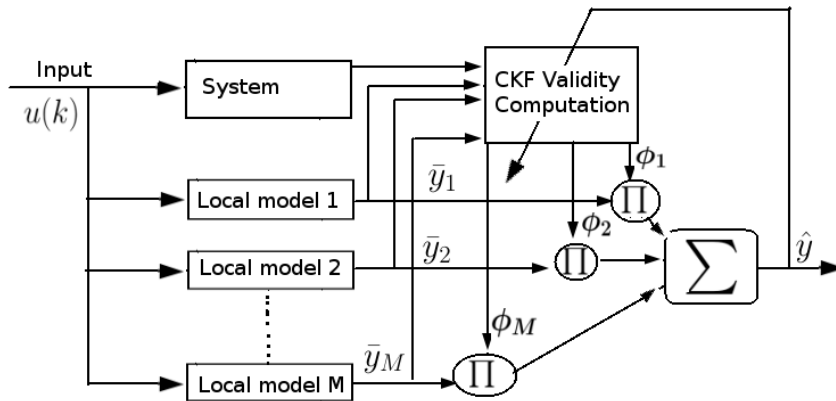


Figure 4.5: CKF based validity computation for output blended multi-model framework

Example 1

Consider a discrete-time system from [109] described by

$$y(k) = a_1(k)y(k-1) + a_2(k)y(k-2) + b_1(k)u(k-1) + b_2(k)u(k-2)$$

The variation laws of different parameters of the process is given by

$$a_1(k) = 0.04\sin(0.035k) + 0.8$$

$$a_2(k) = 0.005\sin(0.03k) + 0.1$$

$$b_1(k) = 0.02\sin(0.03k) + 0.5$$

$$b_2(k) = 0.01\sin(0.035k) + 0.2$$

The submodels representation of the system was identified in section 3.5.1 for 2-parameter and 4-parameter structures with four and two submodels, respectively.

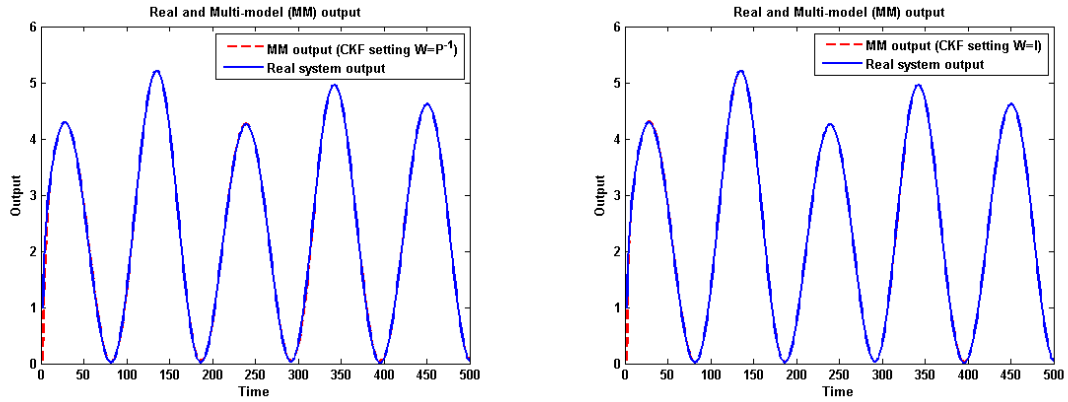
Now using the same validation data in section 3.5.1, the submodels were interpolated with CKF validity estimation as described in section 4.1 to form the multi-model output. The real system's output and the multi-model output for the two assumed submodels' structures are compared using the MSE, PMF, and VAF.

The multi-model identification results are shown in Figures 4.6 to 4.7. As can be observed, the proposed multi-model can well approximate the real system with either 2-parameters or 4-parameters structures using both CKF settings. However, the 2-parameters structure shows better performance than the 4-parameters structure. Also, the CKF with setting $W = I$ has better performance values in 2-parameters submodel structure and similar performance in 4-parameters submodel structure. Table 4.2 shows the comparison of CKF algorithm with the validity estimations used previously. We can conclude from the table, especially in the MSE and PMF columns, that interpolation of the submodels with the CKF

Table 4.2: Performance Measures Comparison of Different Validity Computations

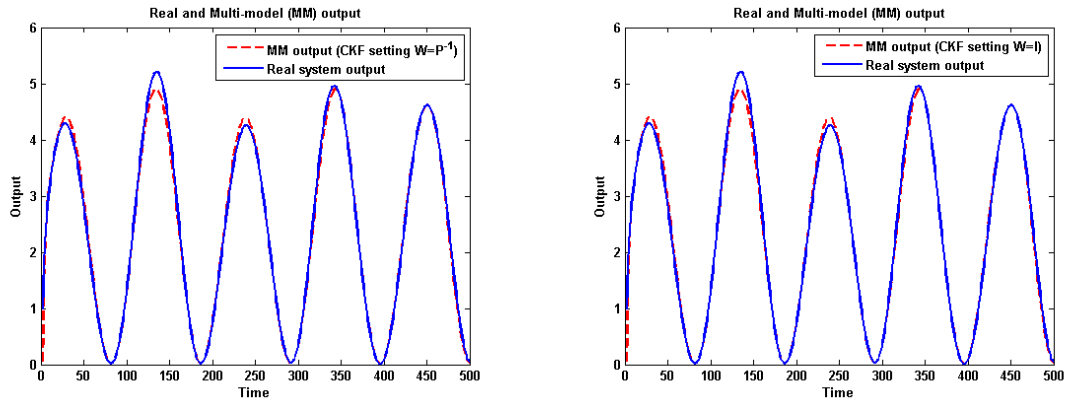
Validity Estimation	2-parameters structure			4-parameters structure		
	MSE	PMF(%)	VAF(%)	MSE	PMF(%)	VAF(%)
simple Residue	0.0403	87.84	98.54	0.0277	89.91	99.00
Reinforced residue	0.0444	87.23	98.39	0.0277	89.91	99.00
Bayessian	0.0751	83.40	97.35	0.0576	85.46	97.89
Qaudratic	0.0765	83.24	97.35	0.0576	85.46	97.89
CKF($W = P^{-1}(k)$)	0.0075	94.74	99.73	0.0150	92.57	99.46
CKF($W = I$)	0.0053	95.58	99.81	0.0151	92.57	99.46

algorithm gave the best results compared to other methods. Furthermore, it can be pointed out that our methods achieved fewer parameters (8) in comparison to the multi-model approach adopted in [109] (12 parameters).



(a) 2-paramters structure with CKF $W = P^{-1}(k)$

(b) 2-paramters structure with CKF $W = I$



(c) 4-paramters structure with CKF $W = P^{-1}(k)$

(d) 4-paramters structure with CKF $W = I$

Figure 4.6: Multi-model identification outputs using validation data

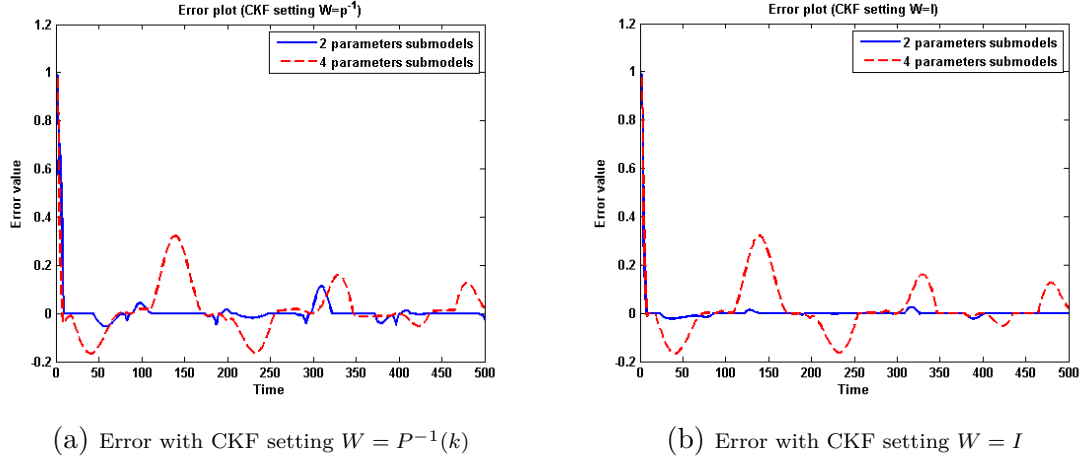


Figure 4.7: Multi-model identification error using validation data

Example 2

Next, consider a nonlinear dynamical system taken from [43]:

$$y(k+1) = (0.6 - 0.1a(k))y(k) + a(k)u(k) \quad (4.23)$$

$$a(k) = \frac{0.6 - 0.06y(k)}{1 + 0.2y(k)} \quad (4.24)$$

This system was identified with two submodels for both 2-parameters and 4-parameters structures in section 3.5.2. To verify the accuracy of the proposed CKF validity, these submodels were interpolated with the CKF validity using during validation using the same validation data in section 3.5.2.

Figures 4.8 and 4.9 show the output of the multi-model and the error obtained, respectively, using the CKF algorithm for interpolation of the submodels with the validation data. One can observe that both submodels' structures can effectively approximate the real system. In addition, both CKF settings show the same performance. Table 4.3 shows the values of the performance measures for different

Table 4.3: Performance Measures Comparison of Different Validity Computations

Validity Estimation	2-parameters structure			4-parameters structure		
	MSE	PMF(%)	VAF(%)	MSE	PMF(%)	VAF(%)
simple Residue	0.0221	83.89	98.39	0.0313	80.84	98.19
Reinforced residue	0.0221	83.89	98.39	0.0313	80.84	98.19
Bayessian	0.0357	79.55	98.18	0.0354	79.64	98.21
Qaudratic	0.0249	82.91	98.42	0.0303	81.17	98.26
CKF($W = P^{-1}(k)$)	0.0124	87.97	99.02	0.0225	83.77	98.81
CKF($W = I$)	0.0124	87.96	99.02	0.0227	83.68	98.82

validity methods including the CKF algorithm. Just as observed in the previous example, The CKF algorithm outperforms other methods.

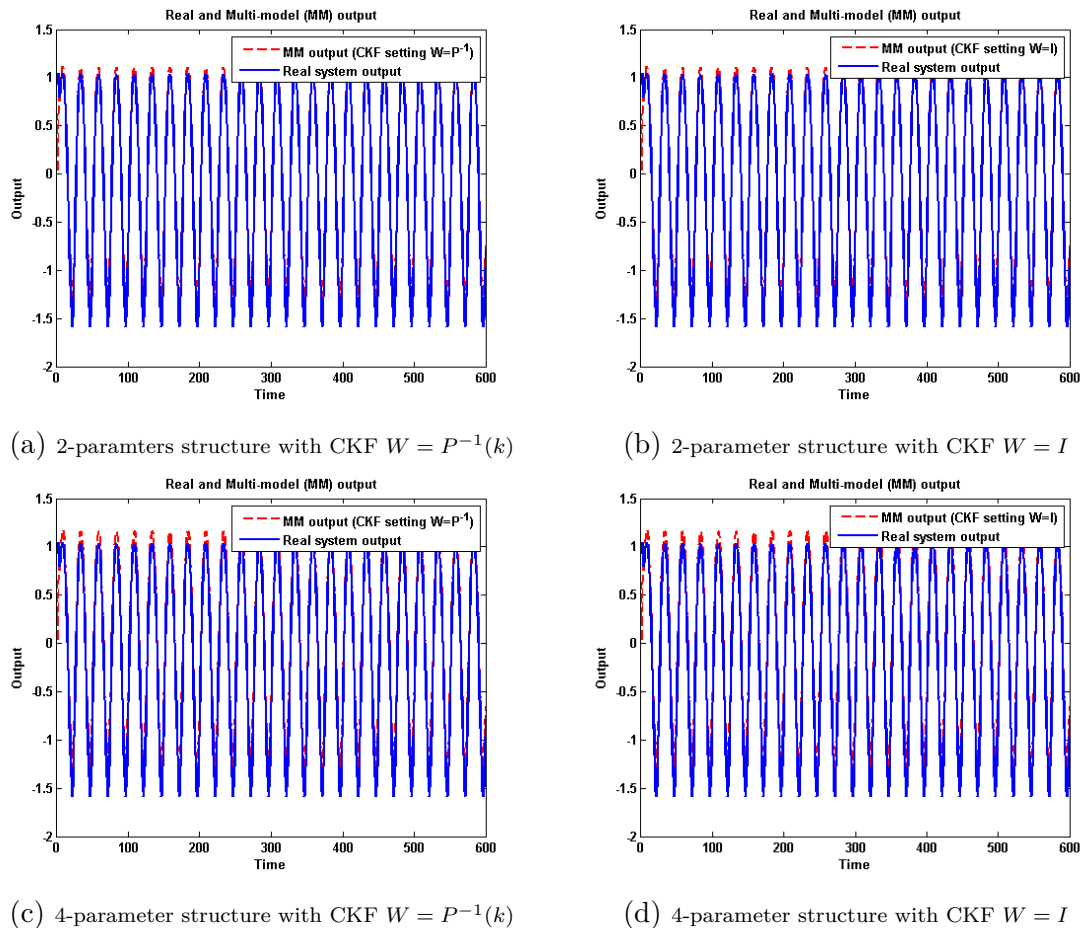
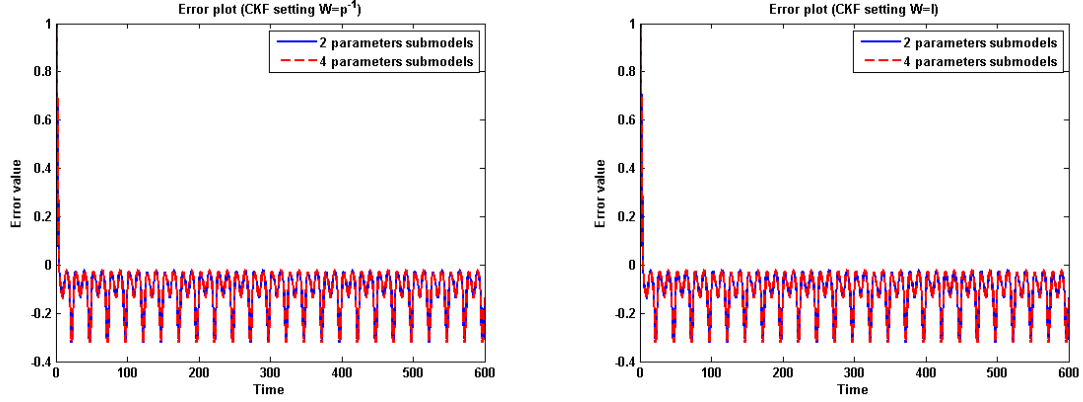


Figure 4.8: Multi-model identification outputs using validation data



(a) Error with CKF setting $W = P^{-1}(k)$

(b) Error with CKF setting $W = I$

Figure 4.9: Multi-model identification error using validation data

Example 3

In the third example, the following nonlinear dynamical system is considered for identification:

$$y(k) = (y(k-1)/(1 + y(k-1)^2)) + u(k-1)^3 \quad (4.25)$$

Using the proposed approach in chapter 3, four and two submodels were identified for 2-parameter and 4-parameter structures respectively. Based on these submodels, The CKF algorithm was used to interpolated the submodels during validation stage using the same validation data in section 3.5.3

Simulation results obtained from Figures 4.10 and 4.11 show that the model outputs closely agree with the system output for both submodels' structures using the CKF algorithm. Table 4.4 shows the performance measures of CKF compared with other validity methods used previously. In general, it can be seen that the CKF validity clearly outperforms other methods.

Table 4.4: Performance Measures Comparison of Different Validity Computations

Validity Estimation	2-parameters structure			4-parameters structure		
	MSE	PMF(%)	VAF(%)	MSE	PMF(%)	VAF(%)
simple Residue	0.8224	65.75	88.28	0.4309	75.21	93.88
Reinforced residue	0.9480	63.23	86.49	0.4309	75.21	93.88
Bayessian	1.2033	58.58	82.85	1.1679	59.19	83.35
Qaudratic	1.1827	58.93	83.15	1.1657	59.23	83.38
$W = P^{-1}(k)$	0.1400	85.87	98.00	0.1954	83.31	97.23
$W = I$	0.1001	88.05	98.58	0.1969	83.24	97.21

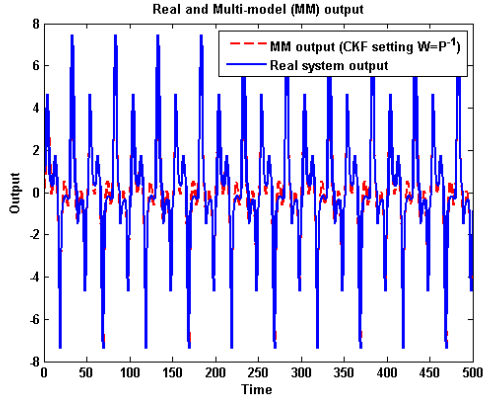
Example 4

Next, the following highly nonlinear dynamical system is considered for identification:

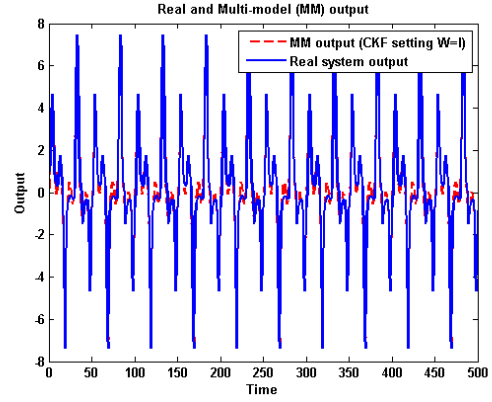
$$\begin{aligned}
 y(k+1) = & \frac{u(k)}{1 + y^2(k-1) + y^2(k-2)} \\
 & + \frac{y(k)y(k-1)y(k-2)u(k-1)(y(k-2) - 1)}{1 + y^2(k-1) + y^2(k-2)} \quad (4.26)
 \end{aligned}$$

Using the proposed approach in chapter 3, four and two submodels were identified for 2-parameter and 4-parameter structures respectively. Based on these submodels, The CKF algorithm was used to interpolated the submodels during validation stage using the same validation data in section 3.5.4

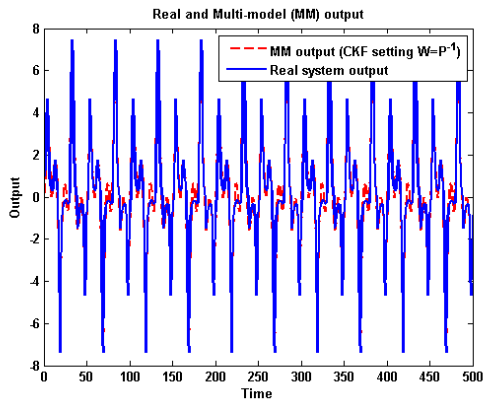
Simulation results obtained as depicted in Figures 4.12 and 4.13 show that the model outputs closely agree with the system output for both submodels' structures using the CKF algorithm. Table 4.5 shows the performance measures of CKF compared with other validity methods used previously. It can be seen that the simple and reinforced validity methods are slightly better than the CKF validity



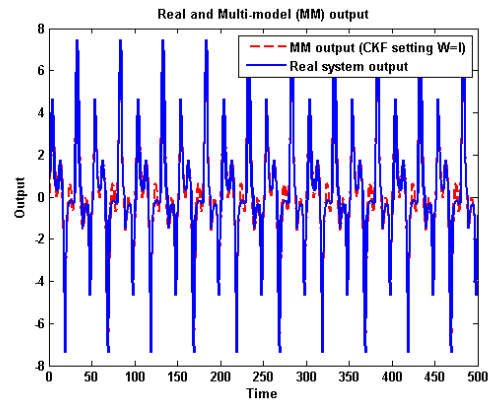
(a) 2-parameter structure with CKF $W = P^{-1}(k)$



(b) 2-parameter structure with CKF $W = I$



(c) 4-parameter structure with CKF $W = P^{-1}(k)$



(d) 4-parameter structure with CKF $W = I$

Figure 4.10: Outputs of multi-model identification using validation data

with settings $W = P^{-1}(k)$ in the 2-parameter structure and the same performance in the 4-parameter structure.

In addition, it can be concluded from the table that the 2-parameter structure with CKF setting $W = I$ shows the best performance. As mentioned earlier, this system has been identified with other multi-model approach in the literature. Using the same validation data set, [126] achieved MSE of 0.0002 and VAF of 99.9% with four third-order models (24 parameters), [172] achieved MSE of 0.112 and VAF of 97.9% with with 10 BPWA functions (306 parameters) and [43] achieved MSE of 0.00067 and VAF of 99.7% with 16 parameters. It is quite noteworthy to

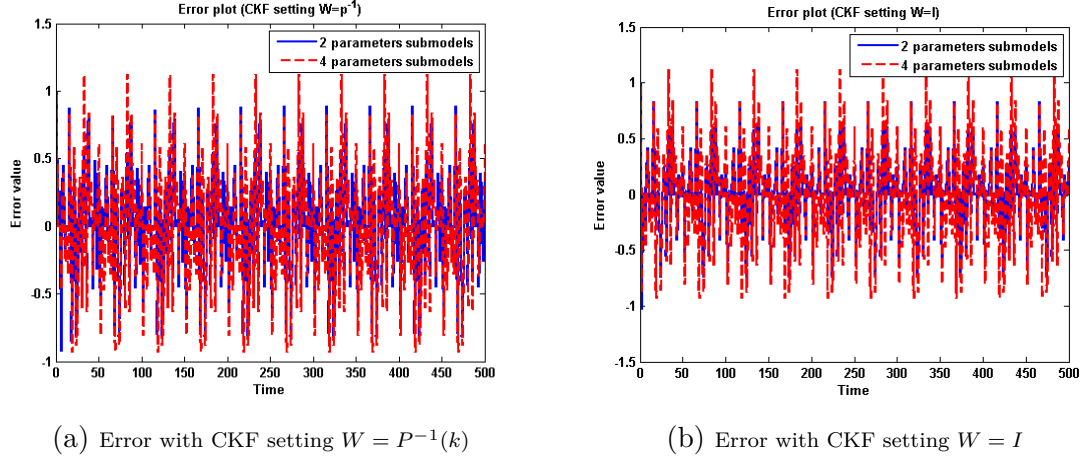


Figure 4.11: Multi-model identification error using validation data

Table 4.5: Validation performance on common validity estimations

Validity Estimation	2-parameters structure			4-parameters structure		
	MSE	PMF (%)	VAF(%)	MSE	PMF(%)	VAF (%)
simple Residue	0.0037	87.52	99.00	0.0013	92.57	99.54
Reinforced residue	0.0045	86.16	98.78	0.0013	92.57	99.54
Bayessian	0.0066	83.35	98.22	0.0082	81.37	97.68
Qaudratic	0.0062	83.79	97.83	0.0110	78.47	97.91
$W = P^{-1}(k)$	0.0054	84.90	98.19	0.0012	92.84	99.54
$W = I$	0.0001	97.37	99.93	0.0004	95.45	99.83

point out that the proposed multi-model approach does yield close performances to other techniques in the literature but with fewer parameters (8).

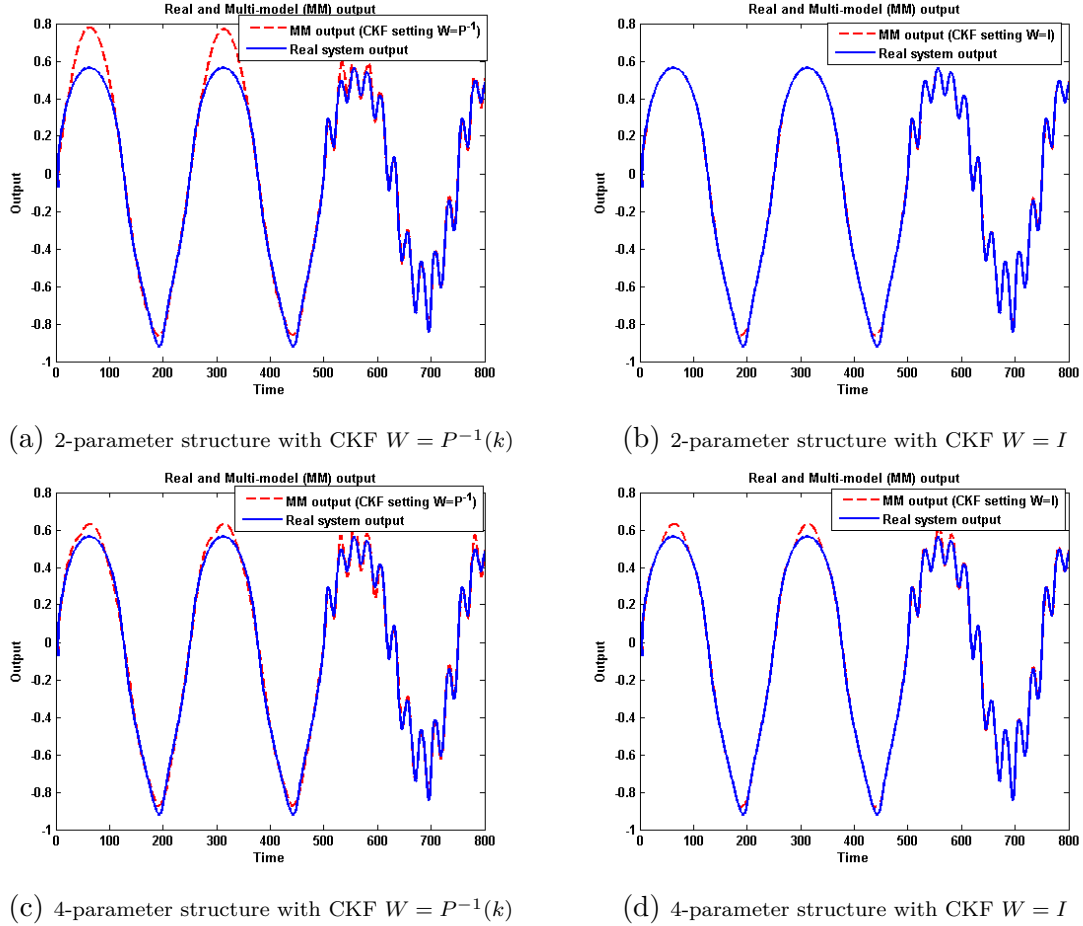


Figure 4.12: Outputs of multi-model identification using validation data

Example 5

The last example is the continuous stirred tank reactor (CSTR) nonlinear chemical system described by in section 3.5.5. The input-output relationship of the system was identified in section 3.5.5 only for 4-parameter structure with three submodels. We also need to remember that submodel 1 is unstable model.

The CKF validity algorithm was used to interpolate the submodels as done previously for other validity methods used in section 3.5.5. The results of the simulation are shown in Figure 4.14. It can be observed that the estimated output matched the real output using the CKF algorithm.

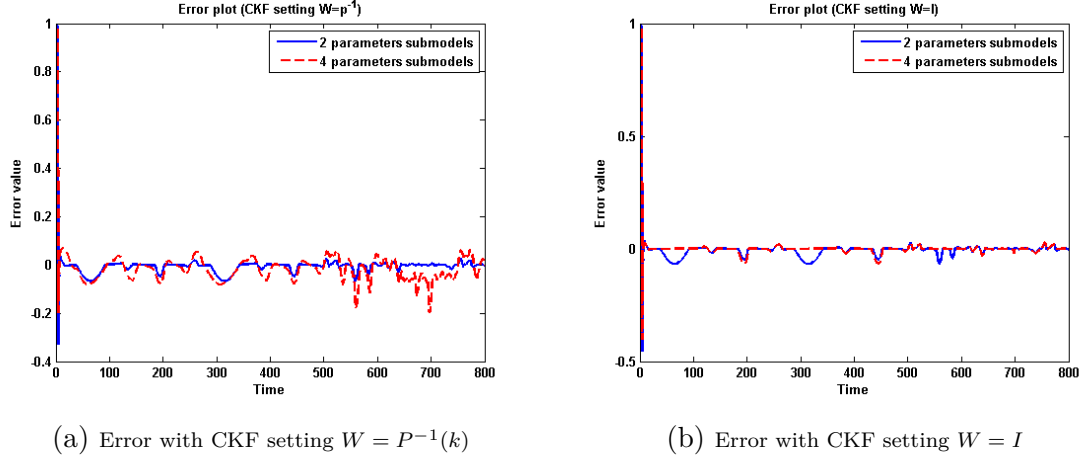
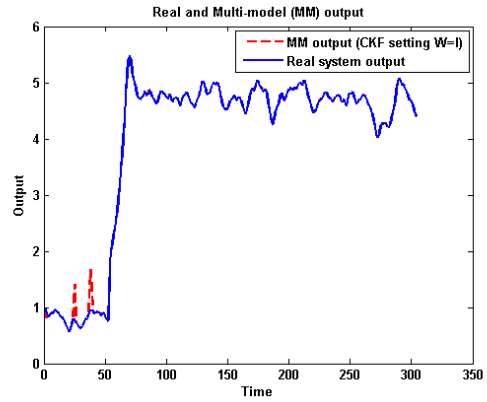
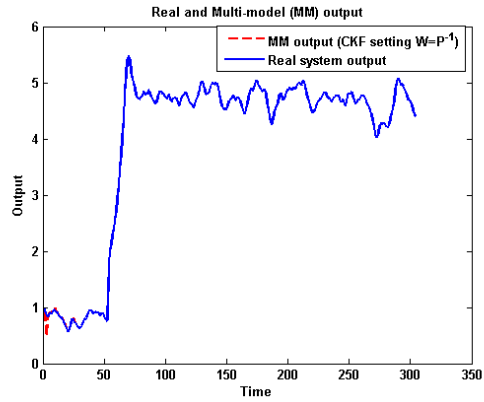


Figure 4.13: Multi-model identification error using validation data

Table 4.6: Performance Measures Comparison of Different Validity Computations

Validity Estimation	4-parameters structure		
	MSE	PAF (%)	VAF (%)
simple Residue	0.0570	84.2310	97.54
Reinforced residue	17.0351	0	0
Bayesian	1.38E+19	0	0
Qaudratic	1.31E+19	0	0
$W = P^{-1}(k)$	0.0005	98.52	99.98
$W = I$	0.0037	96.00	99.84

Table 4.6 shows the comparison of CKF validity with other validity methods. The table revealed the clear outstanding performance of CKF validity estimation despite the presence of unstable model in the submodels.



(a) 4-parameter structure with CKF $W = P^{-1}(k)$

(b) 4-parameter structure with CKF $W = I$

Figure 4.14: Outputs of multi-model identification using validation data

4.5 Conclusion

This chapter focused on validity computation of submodels which is an important issue of the multi-model technique for interpolation of submodels. In this study, a constrained Kalman filter (CKF) validity computation is proposed for multi-model design. The method overcomes some of the drawback of commonly used validity computations for output blended multi-model such as sensitivity to parameter selection, and restriction to partition strategy. The proposed CKF has been implemented and tested for multi-model systems identification using the proposed submodels identification method developed in chapter three. Simulation results show that CKF is of good performance and better than other commonly used validity such as simple residue, reinforced residue, quadratic criterion, and Bayesian validity computations.

CHAPTER 5

OPTIMIZATION OF THE NUMBER OF SUBMODELS' PARAMETERS

Chapter 5 extends the algorithms presented in chapter 3. Here instead of optimizing only the parameters and number of submodels, the order of the submodels are also included. This allows partitioning the operating space and generate a parsimonious number of submodels without prior knowledge. Simulation examples are given to illustrate the effectiveness of the proposed algorithms.

5.1 Introduction

As mentioned earlier, multi-model design involves three steps. The first is the partitioning of the operating system into smaller regions based on a selected strategy.

In the second step, both the structure and the parameters of the local model associated with each subregion are determined. Finally, the local models are combined together using weighting function that defines the contribution of each local model to the real nonlinear system. In chapter 3, it was mentioned that the partitioning process involves optimization of number of submodels and their parameters. Therefore, a modified combinatorial particle swarm optimization (MCPSO) and hybrid K-means algorithms are proposed to obtain the partitions and their associated submodels, only with prior knowledge of the submodels' order or submodels' number of parameter. The knowledge of the submodels' order is to allow the us to incorporate the system's knowledge into the algorithms as well as to provides learning efficiency in the algorithm. However, having prior knowledge of the order of the submodels might be difficult, since the knowledge based on the nonlinear system may not generally translate to the submodels. Hence, there is need to incorporate estimation of submodels' order into our previously presented partition algorithm.

Under this consideration, the main goal of this chapter is to extends the MCPSO algorithm presented in section 3.3 to include estimation of submodels' order (number of parameters). Previous benchmark dynamic systems are used to illustrate the effectiveness of the extension.

The rest of this chapter is organized as follows: section 5.2 describes problem formulation. In section 5.3, the extension of the first stage of the previously proposed partition approach is discussed, and simulation examples are provided

in section 5.4 to demonstrate the effectiveness of the proposed method. Finally a brief conclusion is given in section 5.5.

5.2 Problem Formulation

Again, we consider multi-model system of the form

$$y(k) = \sum_{i=1}^m f_i(x(k))\phi_i(k) \quad (5.1)$$

where $f_i(\cdot)$ and $\phi_i(k)$ are the i^{th} submodel and validity function, respectively and m is the number of submodels. The validity function satisfies the convexity property [159, 132]:

$$\sum_{i=1}^m \phi_i(k) = 1, \quad \forall k \quad (5.2)$$

$$0 \leq \phi_i(k) \leq 1, \quad \forall k, \quad \forall i \in 1, \dots, m \quad (5.3)$$

The system (5.1) can be used to approximate a nonlinear system

$$y(k) = F(y(k-1), y(k-2), \dots, y(k-na), u(k-1), u(k-2), \dots, u(k-nb)) \quad (5.4)$$

where $u(k) \in \mathfrak{R}^{nb}$ is the input and $y(k) \in \mathfrak{R}^{na}$ is the output of the system. The integer nb and na are the time lag of the input and output respectively.

For identification, the first goal is to obtain the representative submodels from a given set of input-output data, by estimating (i) number of submodels (m), and

(ii) the parameters of each submodel (local model) ($f(\cdot)$).

As noted in chapter 3, the function $f(x_i(k))$ is chosen as linear models described by

$$f_i(x(k)) = x_i(k)\theta_i^T(k) \quad (5.5)$$

where θ_i is the vector of parameters of i^{th} submodel which can be estimated from the data pairs:

$$z(k) = \{x(k), y(k) : k = 1, \dots, N\} \quad (5.6)$$

where $x(k) = [y(k-1), y(k-2), \dots, y(k-na), u(k-1), u(k-2), \dots, u(k-nb)]$ is the regressor vector. An affine $x(k)$ is also possible, such that $x(k) = [1, y(k-1), y(k-2), \dots, y(k-na), u(k-1), u(k-2), \dots, u(k-nb)]$.

However, In some cases when na and nb are not known a priori, then the regressor vector $x(k)$ can not be formed. In that case na and nb need to be estimated to be able to form the regressor vector $x(k)$ for each submodel. Therefore, Obtaining the submodels representative for the multi-model identification of complex nonlinear systems (5.4) is to determine (i) number of submodels (m), (ii) the time lag of the input (nb) and output(na) and (iii) the parameters of each submodel (local model) ($f(\cdot)$).

Indeed this problem is similar to the one solved in chapter 3, however with the inclusion of the number of parameters in θ (i.e. $nb + na$). Therefore, the two stage algorithms proposed in chapter 3 shall be extended to solve this problem as follows:

1. *Obtain the number of submodels, the number of parameters and the initial submodel.* This stage involves application of modified combinatorial particle swarm optimization (MCPSO) to obtain m number of partitions, the time lag of the input (nb) and output (na) and the representative data sets for each partition. The m number of clusters obtained from MCPSO is then used to estimate the initial submodels, each with $nb + na$ number of parameters, and initial cluster centers.
2. *Obtain the final submodels.* In this stage hybrid Kmeans criterion is applied to the result of the previous stage to refine the submodels, which can be presented for interpolation. Since this second stage is exactly the same as the one presented in chapter 3, it will not be repeated here.

5.3 Obtain the number of submodels, number of parameters of submodels and the initial partition

The aim of this stage is to determine the number of partitions and to evolve a partition representing a possible grouping of the given data set. That is, given a data set $Z = [z_1, z_2, \dots, z_N]^T$ in \mathcal{R}^d , i.e. N points each with d dimension, we need to simultaneously find the number of partition (m), number of parameters ($na + nb$) and divide Z into m exhaustive and mutually exclusive clusters $P = [p_1, p_2, \dots, p_m]$ with respect to a predefined criteria such that:

1. $p_i \neq \emptyset \quad i = 1, \dots, m;$
2. $p_i \cap p_l = \emptyset \quad i = 1, \dots, m, i \neq l;$
3. $\cup_{i=1}^m p_i = Z$

Remark: Note that z in this case is not as in Equation (5.6). Instead, z is the input- output data pair given by

$$z(k) = \{u(k), y(k) : k = 1, \dots, N\} \quad (5.7)$$

where $u(k)$ and $y(k)$ are the input and output of the nonlinear system.

To achieve this objective the modified CPSO (MCPSO) based partitional algorithm presented in chapter 3 is extended as described in section 5.3.1

5.3.1 Obtain the initial partition using MCPSO

The four features of MCPSO are extended as follows:

1. *Particles encoding:* Each particle position $X_i = [x_{i1}, x_{i2}, \dots, x_{iN}, x_{iN+1}]$, characterized by $N + 1$ elements, where N is the number of data points. Its N elements provide integer numbers representing the cluster number of each data point and $N + 1$ element provide the number of parameters for each submodel, such that $x_{ij} \in \{1, 2, \dots, m_i\}, j = 1, \dots, N$, represents the cluster number of j^{th} data point in i^{th} particle and m_i is the number of clusters associated with i^{th} particle. Also, $x_{iq} \in \{1, 2, \dots, p_i\}, q = N + 1$ represents the $na = nb$ for all submodels in i^{th} particle . m_i and p_i are assumed to lie

in the range $[m_{mim}, m_{max}]$ and $[p_{mim}, p_{max}]$ respectively. m_{min} and p_{mim} are 2 by default and m_{max} and p_{max} are manually specified by the user.

2. *Avoiding empty clusters:* To avoid empty cluster as stated in the previous chapter, new positions of particles are checked. At each generation, particles with empty cluster are corrected by changing the largest cluster number to the smallest unused cluster number. Note that for each particle X_i , dimension $x_{i1}, x_{i2}, \dots, x_{iN}, x_{iN}$ only has to be consider for this operation.
3. *Fitness function:* The fitness criterion used in the MCPSO algorithm of chapter 3 is still used except that it has to reflect the given data in Equation(5.7) and estimation of na and nb . Therefore, given the data set Z defined in Equation(5.7), the cluster regression error is defined by:

$$CRE = \sum_{i=1}^{m'_{max}} SE \quad (5.8)$$

Where

$$SE = \left(\frac{1}{n_i} \sum_{j=1}^{n_i} (y_j - x_j^i \theta_i^T)^2 \right)^{\frac{1}{2}} \quad (5.9)$$

m'_{max} is the maximum number of clusters assigned to a solution, n_i is the number of data points in the i^{th} cluster and θ_i is the parameter of the linear model associated with the i^{th} cluster. This can be obtained using the least square technique as follows:

$$\theta_i = \left[\sum_{j=1}^{n_i} x_j^i x_j^{iT} \right]^{-1} \left[\sum_{j=1}^{n_i} y_j x_j^i \right] \quad (5.10)$$

Where $x^i = [y(k-1), y(k-2), \dots, y(k-na), u(k-1), u(k-2), \dots, u(k-nb)]$ is the regressor vector associated with the i^{th} cluster. we take $na = nb = p'_i$. p'_i is the last element of X_i , representing the order of the submodels.

Remark: Note that at this point the order of the model for each particle is now available from the last element of X_i (i.e. p'_i). Hence the regression matrix for the whole data point N should be constructed before the partition take place, to keep the time dependency of the data. Otherwise, the time dependency of each data point would be lost if the input and output data in (5.7) are partitioned before the construction of the regressor vector.

Finally, the fitness function is defined by:

$$fitness = (W \log N + N \log(CRE^2))/2 \quad (5.11)$$

where $W = m'_{max} + p'_i$ and N is the total number of data points. The smaller is the fitness value, the better is the clustering solution.

4. *Avoiding small size data:* This is the exactly the same as the one chapter 3 and need not to be repeated here.

After the encoding of the particles as discussed above, the execution of MCPSO to obtain number of clusters, number of parameters and the initial partition is done according to the following steps:

- Step 1: Initialize particle position vector X and associated velocity V in the population randomly. For each particle i , the number of clusters m_i and the

number of parameters () are randomly generated in the range $[m_{min}, m_{max}]$ and $[p_{min}, p_{max}]$, respectively. Then each data point is randomly assigned to one cluster.

Step 2: Use $p_i = na = nb$ to construct the regression matrix for the whole data points and evaluate the fitness function for each particle according to Equation (5.11).

Step 3: Compare the fitness value of each particle with its previous best solution (P_{best}) fitness and update P_{best} with the current solution if it is better than the previous value (P_{best}).

Step 4: Compare fitness value with the overall previous best (G_{best}) fitness. Update G_{best} to the current solution if its fitness value is better than G_{best} fitness value.

Step 5: Update positions and velocities of particles using Equation (3.9) to (3.13).

Step 6: Check for empty cluster in all particle solutions and correct if exist.

Step 7 Repeat Step 2 to Step 6 until the maximum number of iterations is completed.

5.3.2 Estimation of the initial submodels

Once the MCPSO search process is terminated, the regression matrix for the whole data ($x(k) = [y(k-1), y(k-2), \dots, y(k-na), u(k-1), u(k-2), \dots, u(k-nb)]$, $k = 1 \dots N$) is first constructed, to preserve the time dependency, using

the last element($p_i = na = nb$) of the global solution during the search process. Also by using the global solution from the MCPSO search process, each cluster representatives $\tilde{z} = \{\tilde{z}_i = (\tilde{x}_i, \tilde{y}_i), i = 1, \dots, m\}$ is obtained, where m is the number of clusters. Subsequently, we estimate the initial submodels for each cluster using the least square algorithm, as done in chapter 3. The coefficients vector θ_i for each submodel is computed through the formula:

$$\tilde{\theta}_i = (\tilde{\Phi}_i^T \tilde{\Phi}_i)^{-1} \tilde{\Phi}_i^T y_i \quad (5.12)$$

where $\tilde{\Phi}_i = [x_i(1), \dots, x_i(n_i)]^T$ and y_i are the regression matrix and output vector belonging to i^{th} cluster, respectively. n_i is the number of data points in the i^{th} partition.

In addition, the centers of the data are calculated by finding the mean of the data in each cluster produced by the previous stage. The center of each cluster is given as :

$$\tilde{c}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \tilde{x}_{ij} \quad i = 1, 2, \dots, m. \quad (5.13)$$

Once the estimation of the initial submodel and their associated centers has been obtained, the final parameters θ of each submodel are obtained along with their associated centroid c_i using the procedure in section 3.4. Finally, the submodels are now ready for interpolation to obtain the final global model that will represent the nonlinear system under consideration.

5.4 Simulation Examples

The effectiveness of the proposed partition method without prior knowledge of number of parameters for submodels is demonstrated in this section. Two simulation examples were carried out. Except stated otherwise, the parameter settings of MCPSO used are given in Table 5.1. Also $\lambda = 0.01$ is selected in stage 2 throughout the simulations. The CKF validity computation presented in chapter 4 is used for the interpolation of the submodels, to form the multi-model representation of nonlinear system.

The obtained multi-model is evaluated based on the validation data using the mean square error (MSE), percentage model fitness (PMF) and variance-accounted-for (VAF) performance measures:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y(i) - \hat{y}(i)) \quad (5.14)$$

$$PMF = \max\left(1 - \frac{\|y - \hat{y}\|}{\|y - \text{mean}(y)\|}, 0\right) \times 100 \quad (5.15)$$

$$VAF = \max\left(100 \times \left(1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)}\right), 0\right) \quad (5.16)$$

Where y is the real system output \hat{y} is the multi-model estimated output, $\|\cdot\|$ denotes norm and $\text{var}(\cdot)$ denotes the variance. All simulations are performed using MATLAB 2012b on a 2.4 GHZ i3 64-bits Windows machine with 4 G RAM.

Table 5.1: MCPSO parameter settings

parameters	values
Swarm size	20
Max. Iterations	5000
w, α	0.4, 0.35
c_1, c_2	2, 2
m_{min}, m_{max}	2, 20
p_{min}, p_{max}	2, 7

5.4.1 Example 1

The first example considered is the previously presented highly nonlinear dynamical system describe by

$$y(k) = \frac{y(k-1)}{1 + y(k-1)^2} + u(k-1)^3; \quad (5.17)$$

The system was excited by uniformly distributed random signal in the interval $[-1, 1]$. The identification was carried out with data set of 800 samples.

With the proposed extension we are able to obtain two submodels with four parameters each in the learning process. The convergence of the objective function in the extended MCPSO is shown in Figure 5.1. The initial and final submodels parameters with their associated centers are shown in Table 5.2 and 5.3 respectively.

Validation of the multi-model identification was done with second data set of 500 samples generated by an input signal given by:

$$u(k) = \sin\left(\frac{2\pi}{25}k\right) + \sin\left(\frac{2\pi}{10}k\right)$$

Table 5.2: Results of Stage 1 (initial submodels)

Initial submodels	a_1	a_2	b_1	b_2
1	-0.10673	0.04935	2.62918	0.42943
2	-0.05355	0.03074	1.92930	0.30293
Initial centers				
1	0.04696	-0.01651	-0.01008	0.02466
2	0.21073	0.29268	0.10032	0.06193

Table 5.3: Results of Stage 2 (final submodels)

Final submodels	a_1	a_2	b_1	b_2
1	-0.09183	0.06339	3.09449	0.41960
2	-0.09659	0.02118	1.24222	0.45112
Final centers				
1	0.28352	-0.04442	0.13273	0.09495
2	0.06090	0.19251	0.00573	0.02149

Simulation results obtained using the CKF validity estimation for the validation data set are shown in Figures 5.2 and 5.3. The figures show that the estimated model outputs closely follows the system output for both submodels' structures. The result of CKF validity computation for setting $W = I$ is the same as that obtained in example 3 of chapter 4 with the 4-parameter structure, which has the same number of submodels and parameters. However, It would have been expected that the result of this extension algorithm should be 4 and 2 for number of submodels and parameters respectively (2-paramter structure), since this gave better performance in example three of chapter 4. This confirmed our statement previoulsy that given the user the opportunity to pre-select the number

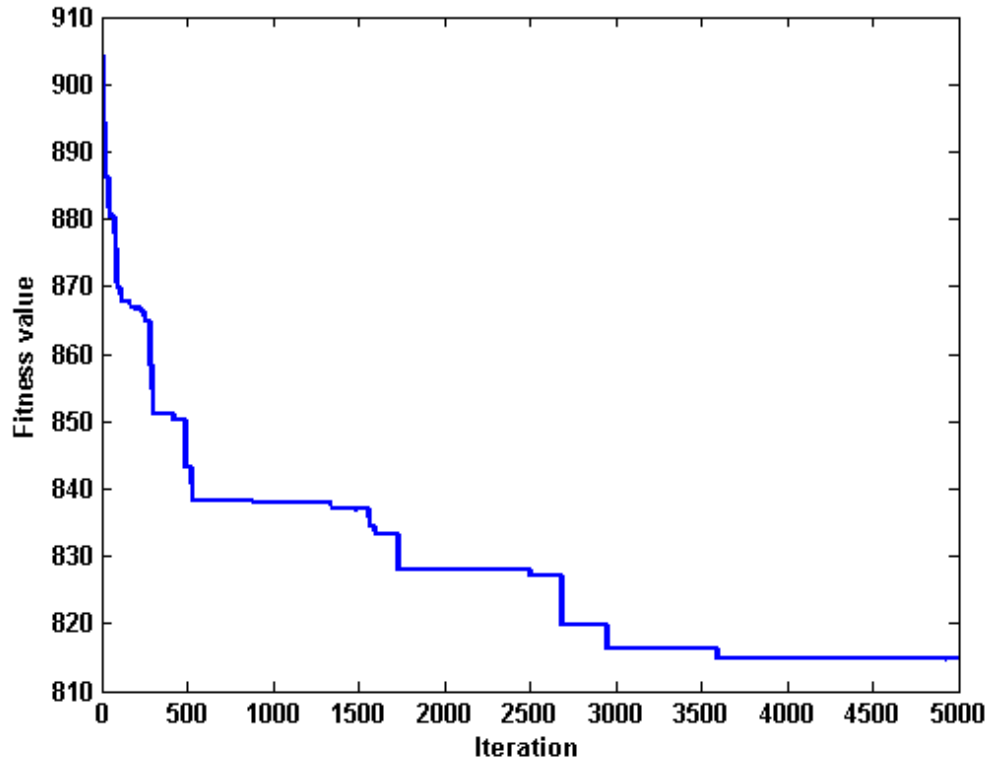
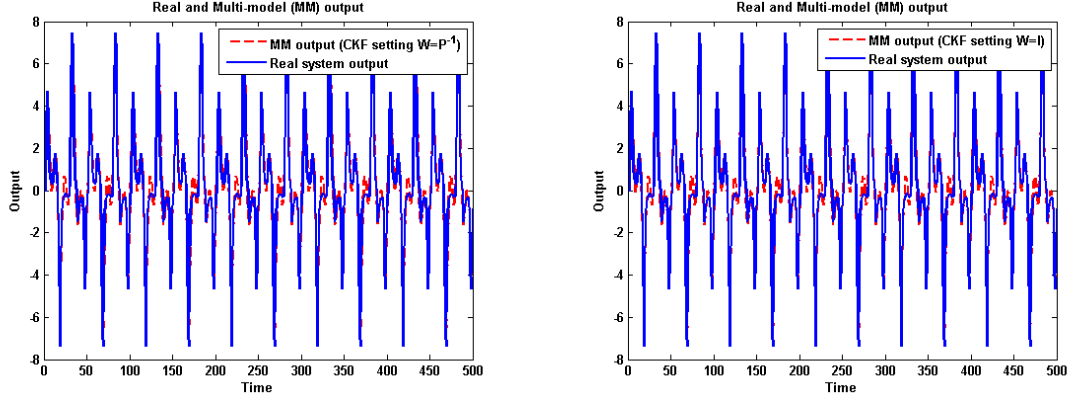


Figure 5.1: MCPSO objective function convergence plot

of parameters would provides learning efficiency in the MCPSO algorithm.

Table 5.4: Validation performance for system 1

CKF settings	MSE	PMF (%)	VAF(%)
$W = P^{-1}(k)$	0.3432	77.86	95.12
$W = I$	0.2046	82.90	97.10



(a) Using CKF $W = P^{-1}(k)$ validity computation (b) Using CKF $W = I$ validity computation

Figure 5.2: Outputs of multi-model identification using validation data for system 1

5.4.2 Example 2

The next example considered for identification is another highly nonlinear dynamical system described by

$$\begin{aligned}
 y(k+1) &= \frac{u(k)}{1 + y^2(k-1) + y^2(k-2)} \\
 &+ \frac{y(k)y(k-1)y(k-2)u(k-1)(y(k-2) - 1)}{1 + y^2(k-1) + y^2(k-2)} \quad (5.18)
 \end{aligned}$$

The system was excited by uniformly distributed random signal in the interval $[-1, 1]$. The identification was carried out with data set of 800 samples as done previously.

Using the proposed extension, two submodels were identified with four parameters each. The convergence of the objective function in the extended MCPSO is shown in Figure 5.4. The initial and final submodels parameters with their associated centers are shown in Table 5.5 and 5.6 respectively.

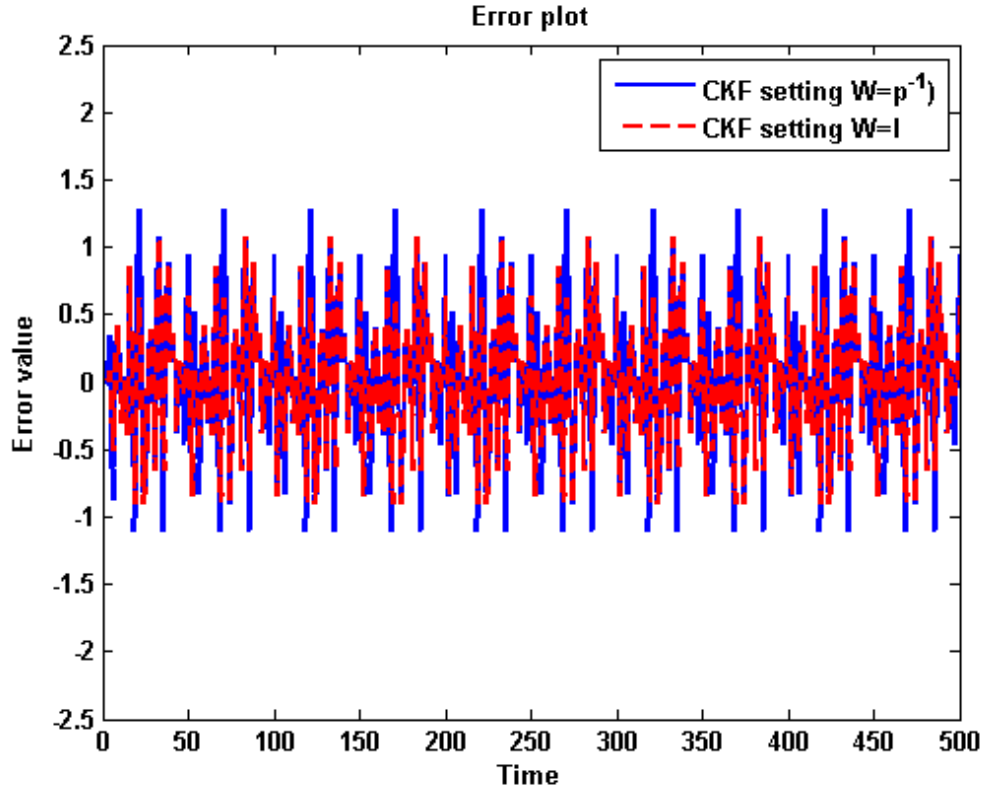


Figure 5.3: Multi-model identification error using validation data for system 1

Table 5.5: Results of Stage 1 (initial submodels)

Initial submodels	a_1	a_2	b_1	b_2
1	-0.27894	0.05235	0.70949	0.20132
2	0.13377	0.00174	0.80020	-0.10636
Initial centers				
1	-0.03936	0.00353	0.00595	-0.05240
2	0.03067	-0.02556	-0.02435	0.04807

Validation of the multi-model identification was done with second data set of

Table 5.6: Results of Stage 2 (final submodels)

Final submodels	a_1	a_2	b_1	b_2
1	-0.10342	0.04138	0.59581	0.07318
2	-0.05368	0.03737	0.86455	0.02884
Final centers				
1	-0.16968	-0.04723	-0.00707	-0.22956
2	0.12770	0.02303	-0.00751	0.17897

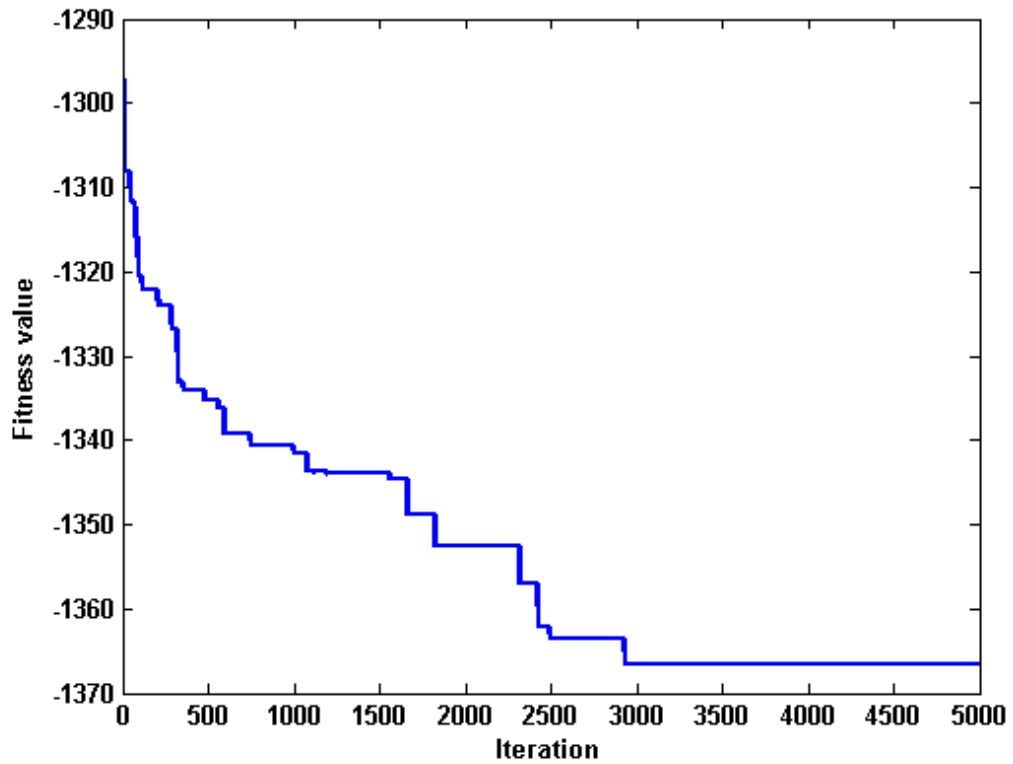


Figure 5.4: MCP SO objective function convergence plot

800 samples generated by an input signal given by:

$$u(k) = \begin{cases} \sin\left(\frac{2\pi}{250}k\right) & \text{if } k \leq 500 \\ 0.8\sin\left(\frac{2\pi}{250}k\right) + 0.2\sin\left(\frac{2\pi}{25}k\right) & \text{if } k > 500 \end{cases}$$

Table 5.7: Validation performance on common validity estimations

CKF settings	MSE	PMF (%)	VAF(%)
$W = P^{-1}(k)$	0.0012	93.00	99.56
$W = I$	0.0004	95.72	99.84

Simulation results obtained using CKF validity estimation for the validation data set are shown in Figures 5.5 and 5.6. The figures show that the estimated model outputs follows the nonlinear system output. Also, it can be observed from Table 5.7, that the results obtained are almost the same as that obtained in section 4.4.2 with 4-parameter structure. Similarly as observed in the example 1, it is expected that the number of submodels and parameters would be 4 and 2 respectively (2-parameter structure), since this gave better performance in chapter 4. This did not happen, as the learning efficiency has been reduced with inclusion of estimation of number of parameters.

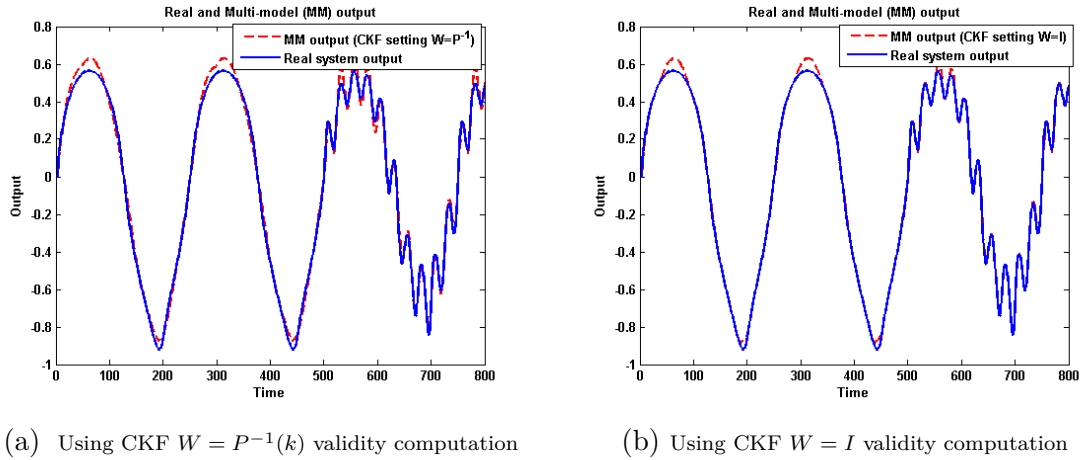


Figure 5.5: Outputs of multi-model identification using validation data

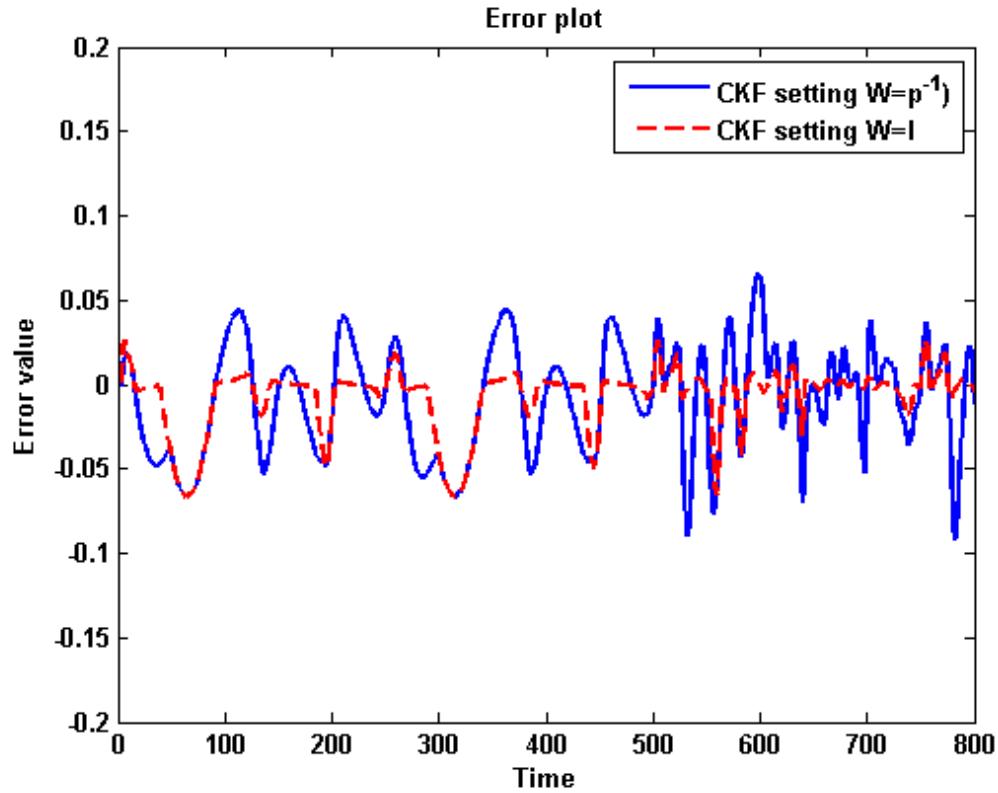


Figure 5.6: Multi-model identification error using validation data

5.5 Conclusion

Chapter 5 presents an extension of the meta-heuristic partition method for multi-model identification of nonlinear systems presented in chapter 3. Unlike in chapter 3, it is assumed that the number of parameters of the submodels are not known. Hence, the first stage of the procedure presented in chapter 3 is modified to include estimation of number of parameters for the submodels. Simulations example carried out showed the capability of the extension.

CHAPTER 6

APPLICATION IN CONTROL AND FAULT DIAGNOSTICS

Chapter 6 focused on the applications of the proposed multi-model framework developed in the previous chapters. Two important application areas namely, control and fault diagnosis, are investigated on the multi-model framework. In the first, some of the systems identified in the previous chapter are evaluated for reference tracking control, using weighted one-step ahead control algorithm. In the second, the suitability of the CKF algorithm under multi-model framework is tested for fault detection and isolation on a three-tank system.

6.1 Multi-model control of nonlinear systems

Increase in the research of modeling and control of nonlinear systems have been steady over several years, due to the inherent nonlinearity and wide operating conditions in virtually all the industrial processes. However, nonlinear controls

present substantial demand in terms of design and implementation compared to linear controls. Therefore control engineers usually opt for linear controllers to control a linear approximation of the nonlinear system, in order to benefit from its easy implementation and rich linear control methodologies.

Unfortunately in practice, a linear controller for a nonlinear system may exhibit performance limitation especially when applied to the entire operating conditions of the system. One promising remedy, which continued to gain acceptance, to overcome this problem is the multi-model control. In this approach linear controllers are designed for different sub-operating regions of the nonlinear system. There can either be switching among the controllers or interpolation of the controllers. In switching multi-model control, a supervisor is designed to select an appropriate controller at every instant based on the state of the system. This scheme has been used in [200, 201, 202, 203, 155]. One concern about the switching scheme is the stability of the system especially during the transient period [200, 204].

Interpolation of multiple controllers is another possible scheme where the output or parameters of multiple controllers are weighed and summed up according to a rule to form the final control. Several contributions to the interpolation scheme can be found in the literature. In [139], PI controllers are designed as local controllers based on operating point linearization of the nonlinear system and their output are weighted as a function of defined closed-loop gap metric. PID local controllers are also reported in [148, 149, 150] with Gaussian weighting function

for interpolation of the outputs of the controllers. In [55, 56, 53, 54] the local controller utilized loop shaping H_∞ control and their output are weighted with a trapezoidal function. In [141] an optimal robust controller is designed with gap metric weighting function. Internal model multiple controllers is utilized in [34] with Gaussian weighting function for interpolation of the outputs. In [151], RST controller are reported with residue weighting both for outputs and parameters interpolation.

Model predictive control (MPC) is another control methodology that has become popular with the multi-model framework. Usually two methods are adopted in the interpolation scheme. In the first, multiple linear models are interpolated to predict the output of the nonlinear system that will be used in the control optimization stage [205, 152, 146, 124, 153, 154]. In the second, the multiple MPC are designed and interpolated to form the global control of the nonlinear system [206, 207]. Interpolation of multiple MPC is less attractive due to computational load on the optimization algorithms for all the controllers, depending on the system complexity. Different interpolation rules have also been applied in the multi-model MPC. Gaussian function is applied in [205, 207, 124, 153, 152] while Bayesian and triangular function rules are utilized respectively in [146, 154].

In this section, a model-based controllers will be derived to control nonlinear systems for reference tracking. The multiple models are generated using the heuristic and meta-heuristic approach as proposed in the previous chapter. In the study two multi-model controls configurations are investigated with weighted

one-step-ahead controller. The first configuration is the fusion of controller outputs, while in the second the parameters of the models are fused together to generate the final control signal. In both configurations, the weights needed for interpolation are generated online using the CKF validity proposed in chapter 4. Simulation studies on previously identified systems verified the efficacy of both control algorithms.

6.1.1 Weighted One-Step-Ahead Controller

Given an input-output linear model expressed in the following form:

$$y(k+1|k) = a_0y(k) + a_1y(k-1) + \dots + a_nay(k-na) \quad (6.1)$$

$$+ b_0u(k) + b_1u(k-1) + \dots + b_nbu(k-nb) \quad (6.2)$$

To ensure that the output of the model, $y(k+1)$, tracks the a reference signal $r(k+1)$, we can minimizing the square of the difference between the output of the model and that of the reference at every instant with respect to $u(k)$. Therefore, minimizing the following cost function with respect to $u(k)$:

$$J = (r(k+1) - y(k+1))^2 + \lambda u^2(k) \quad (6.3)$$

yields a weighted one-step-ahead control law as follows

$$u(k) = \frac{b_0}{b_0^2 + \lambda} \begin{bmatrix} r(k+1) \\ -a_0y(k) - a_1y(k-1) - \dots - a_{n_a}y(k-n_a) \\ -b_1u(k-1) - \dots - b_{n_b}u(k-n_b) \end{bmatrix} \quad (6.4)$$

The λ in the cost function is a user tuning parameter to achieve a balance between the control magnitude and tracking accuracy.

6.1.2 Fusion of Weighted One-Step-Ahead Controllers

In the fusion of controllers, the outputs of multiple controllers are weighted based on the contribution of each model, to obtain the final control signal of the nonlinear system. Although this configuration allows different control algorithms to be designed for each model that represents the system, same control algorithms are used in this thesis.

Given a nonlinear system

$$y(k) = F(y(k-1), \dots, y(k-n_a), u(k-1), \dots, u(k-n_b)) \quad (6.5)$$

The control problem is to ensure that $y(k+1)$, tracks the a reference signal $r(k+1)$. Therefore, as depicted in Figure 6.1, given the submodels that are obtained from the nonlinear system, a weighted one-step-ahead controller is designed based on Equation (6.4) for each submodel and weighted to form the final control signal to excite the nonlinear system. That is, given the cost function for each submodel

as

$$J_i = (r(k+1) - y_i(k+1))^2 + \lambda u^2(k) \quad (6.6)$$

The weighted one-step-ahead controller for each submodel can be written as:

$$u_i(k) = \frac{b_{i0}}{b_{i0}^2 + \lambda} \begin{bmatrix} r(k+1) \\ -a_{i0}y_i(k) - a_{i1}y_i(k-1) - \dots - a_{in_a}y_i(k-n_a) \\ -b_{i1}u(k-1) - \dots - b_{in_b}u(k-n_b) \end{bmatrix} \quad (6.7)$$

Finally, the final control signal for the nonlinear system is obtained as :

$$u(k) = \sum_{i=1}^m u_i(k)\phi_i(k) \quad (6.8)$$

where $\phi_i(k)$ is the validity of each model obtained from the CKF algorithm, developed in Chapter 4.

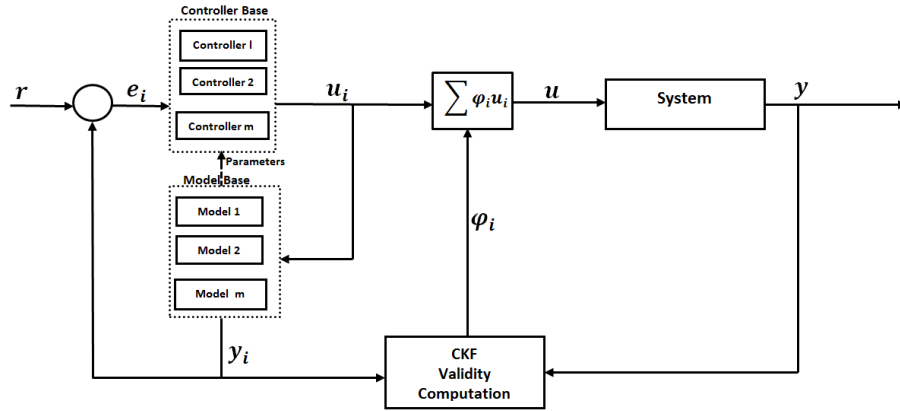


Figure 6.1: Fused controller multi-model control configuration

6.1.3 Fusion of Model Parameters for Weighted One-Step-Ahead Controller

In this configuration, rather than designing multiple controllers and merging them, a single controller is designed by considering the weighted multi-model output representation of the system. Hence the cost function in Equation (6.3) is rewritten as:

$$J = (r(k+1) - \sum_i^m y_i(k+1)\phi_i(k+1))^2 + \lambda u^2(k) \quad (6.9)$$

As shown in Chapter 2, the global output of multi-model representation is given as

$$y(k) = \sum_{i=1}^m y_i(k)\phi_i(k), \quad (6.10)$$

where m is the number of submodels, $y_i(\cdot)$ and $\phi_i(k)$ are the i^{th} submodel's output and validity, respectively. Note that the output of the each submodel can be written as

$$y_i(k+1|k) = \sum_{j=0}^{n_a} a_{ij}y_i(k-j) + \sum_{j=0}^{n_b} b_{ij}u(k-j) \quad (6.11)$$

Hence, Equation (6.10) becomes

$$y(k+1) = \sum_{i=1}^m \phi_i(k+1) \left[\sum_{j=0}^{n_a} a_{ij}y_i(k-j) + \sum_{j=0}^{n_b} b_{ij}u(k-j) \right] \quad (6.12)$$

Factoring out the $u(k)$ term yields

$$y(k+1) = \sum_{i=1}^m \sum_{j=0}^{n_a} \phi_i a_{ij} y_i(k-j) + \sum_{i=1}^m \sum_{l=1}^{n_b} \phi_i b_{il} u(k-l) + \sum_{i=1}^m \phi_i b_{i0} u(k) \quad (6.13)$$

where the $k + 1$ index of ϕ_i has been dropped for easy notation.

Substituting Equation (6.13) into the cost function in Equation (6.9), and minimizing with respect to $u(k)$ yields a weight one-step-ahead controller as follows:

$$u(k) = \frac{\sum_{i=1}^m \phi_i b_{i0}}{(\sum_{i=1}^m \phi_i b_{i0})^2 + \lambda} \begin{bmatrix} r(k+1) \\ -\sum_{i=1}^m \sum_{j=0}^{n_a} \phi_i a_{ij} y_i(k-j) \\ -\sum_{i=1}^m \sum_{l=1}^{n_b} \phi_i b_{il} u(k-l) \end{bmatrix} \quad (6.14)$$

Figure 6.2 gives the schematic representation of the fusion of model parameters for weighted one-step-ahead controller.

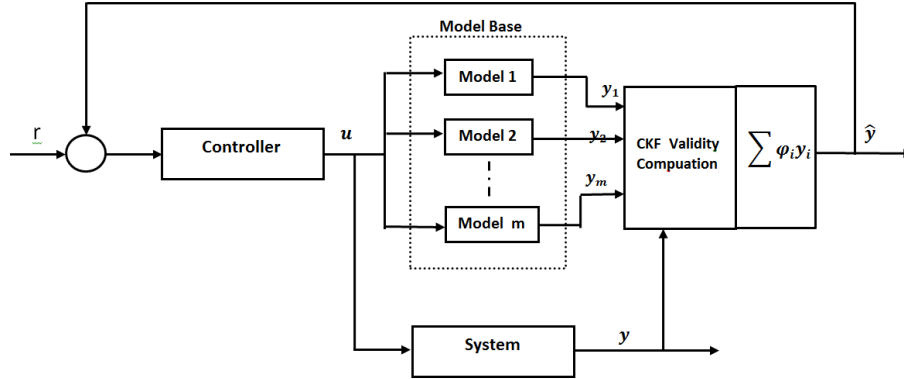


Figure 6.2: Fused parameters multi-model control configuration

6.1.4 Simulation Examples

Two controller designed in sections 6.1.2 and 6.1.3, subsequently refer to as Type A and Type B respectively, were evaluated on some of the benchmark systems presented in the chapters 4. The CKF settings $W = P^{-1}(k)$ and $W = I$ are refer to as CKFP and CKFW respectively. The results of the controllers were assessed by MSE, PMF and VAF.

Example 1

Consider the following discrete time varying system described by :

$$y(k) = a_1(k)y(k-1) + a_2(k)y(k-2) + b_1(k)u(k-1) + b_2u(k-2)$$

where the parameters variation laws are:

$$a_1(k) = 0.04\sin(0.035k) + 0.8$$

$$a_2(k) = 0.005\sin(0.03k) + 0.1$$

$$b_1(k) = 0.02\sin(0.03k) + 0.5$$

$$b_2(k) = 0.01\sin(0.035k) + 0.2$$

The above system was identified in section 4.4.2 with 2-parameter and 4-parameter structures, with good accuracy, using four and two submodels, respectively. Now, the time varying system is controlled using the Type A and Type B controllers while tracking the reference signal

$$r(k) = 0.5 \sin(0.2k) + 0.7 \sin(0.04k)$$

The performance of the two controllers are shown in Table 6.2, Figure 6.3 and Figure 6.4. The control signals are shown in Figure 6.5 and Figure 6.6 respectively for Type A and Type B controllers. The controller tuning parameters are shown in Table 6.1. The R and Q parameters of the CKF algorithm was estimated

using the proposed method in section 4.3.3. Although, both controllers were able to track the signal, some little differences can be noticed. We can observe from Table 6.2 that there is slightly better performance when the CKFW is used. Also, the 2-parameter is better in performance than 4-parameter structure and lastly, the Type B controller is slightly better in performance than Type A despite it is a single controller.

Furthermore, the effect of R and Q parameters in the CKF algorithm was investigated by randomly optimizing the two parameters. The results is as shown in Table 6.3. It can be observed that there is a slight improvement in the results of both controllers when CKFP validity computation is used, due to change in the value of Q . Recall that the parameter Q is added to avoid numerical instability, particularly for the CKFP and can be omitted for CKFW. Therefore, the estimation of R and Q using the method presented in session 4.3.3 may be sufficient, at least for the CKFW.

Table 6.1: Controllers' parameters (λ)

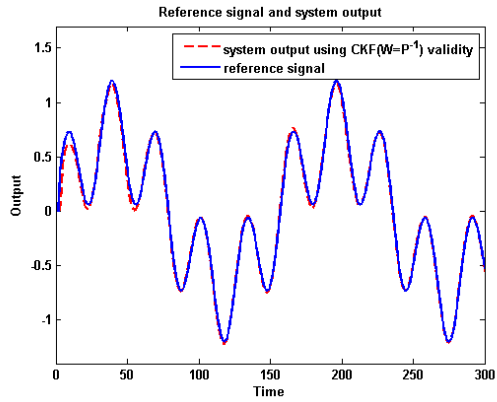
CKF setting	2-parameters structure	4-parameters structure
Type A controller		
CKFP	0.03	0.009
CKFW	0.03	0.0009
Type B controller		
CKFP	0.04	0.002
CKFW	0.004	0.002

Table 6.2: Performance of controllers

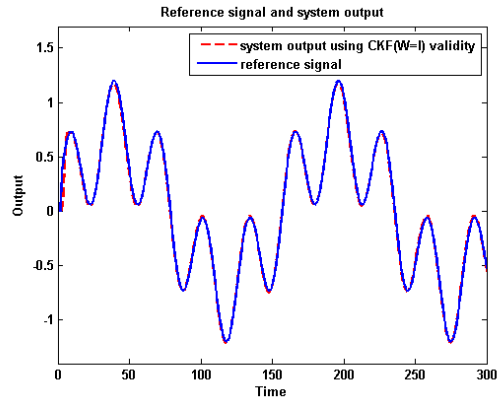
CKF setting	2-parameters structure			4-parameters structure		
	MSE	PMF(%)	VAF(%)	MSE	PMF(%)	VAF(%)
Type A controller						
CKFP	0.0021	92.47	99.49	0.0020	92.578	99.52
CKFW	0.0014	93.83	99.64	0.0037	89.96	99.28
Type B controller						
CKFP	0.0036	90.16	99.17	0.0015	93.64	99.76
CKFW	0.0002	97.92	99.96	0.0012	94.25	99.80

Table 6.3: Performance of controllers for optimized R and Q in CKF

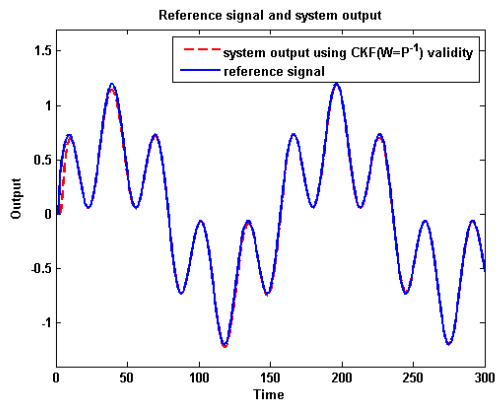
CKF setting	2-parameters structure			4-parameters structure		
	MSE	PMF(%)	VAF(%)	MSE	PMF(%)	VAF(%)
Type A controller						
CKFP	0.0021	92.47	99.49	0.0020	92.60	99.52
CKFW	0.0014	93.83	99.64	0.0037	90.02	99.28
Type B controller						
CKFP	0.0011	94.45	99.71	0.0011	94.52	99.80
CKFW	0.0002	97.94	99.96	0.0011	94.52	99.80



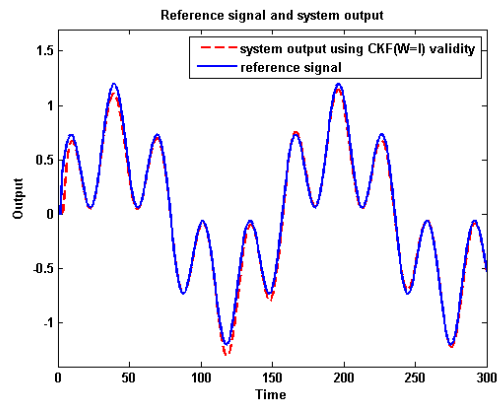
(a) control using 2-paramters structure with CKFP



(b) control using 2-paramters structure with CKFW

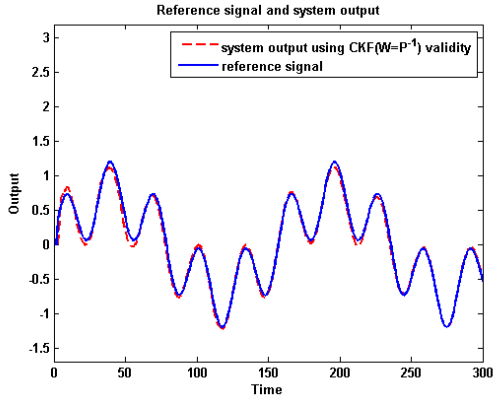


(c) control using 4-paramters structure with CKFP

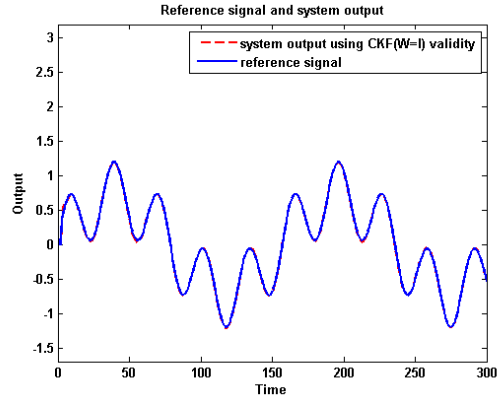


(d) control using 4-paramters structure with CKFW

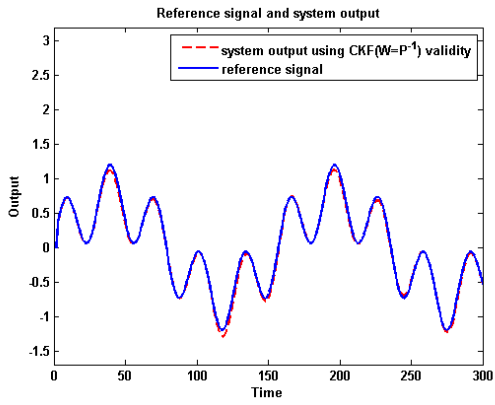
Figure 6.3: Control of time varying system using Type A controller



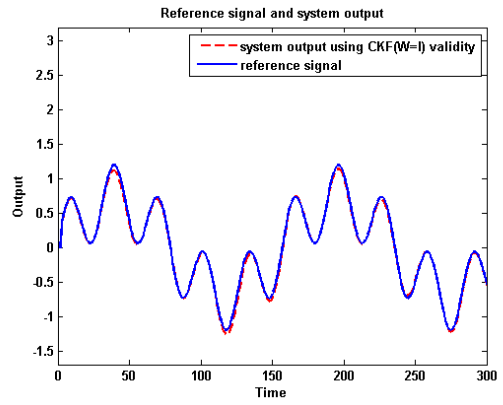
(a) control using 2-paramters structure with CKFP



(b) control using 2-paramters structure with CKFW

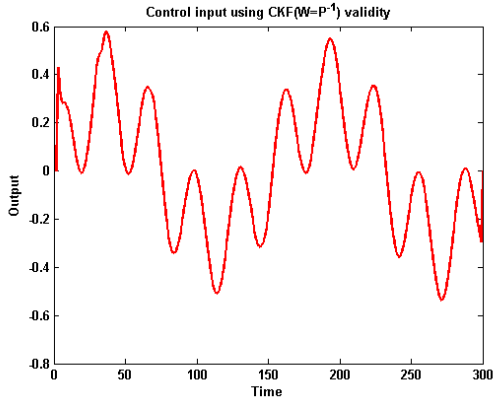


(c) control using 4-paramters structure with CKFP

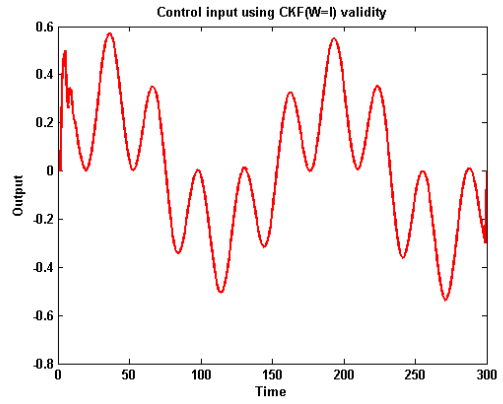


(d) control using 4-paramters structure with CKFW

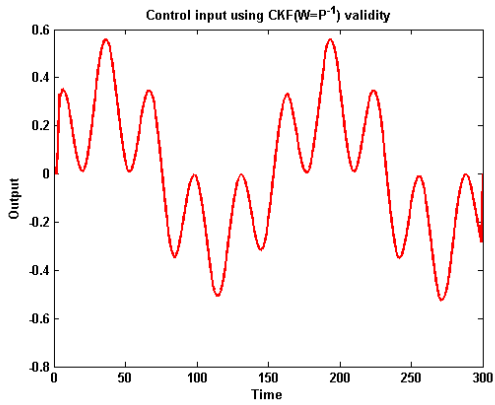
Figure 6.4: Control of time varying system using Type B controller



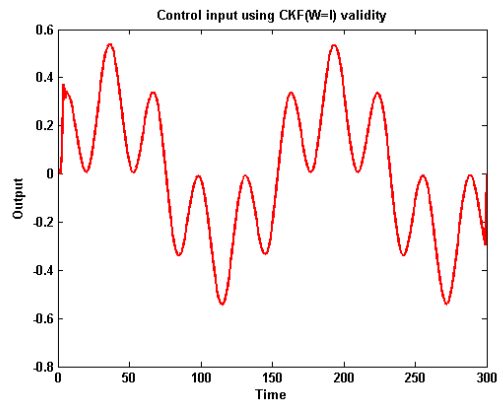
(a) using 2-paramters (CKFP)



(b) using 2-paramters (CKFW)

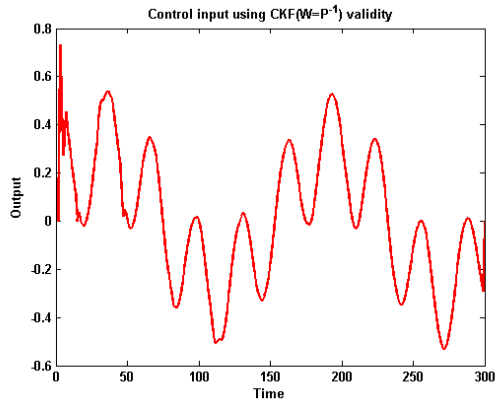


(c) using 4-paramters (CKFP)

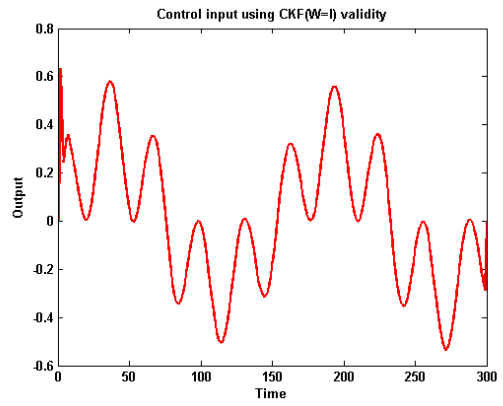


(d) using 4-paramters (CKFW)

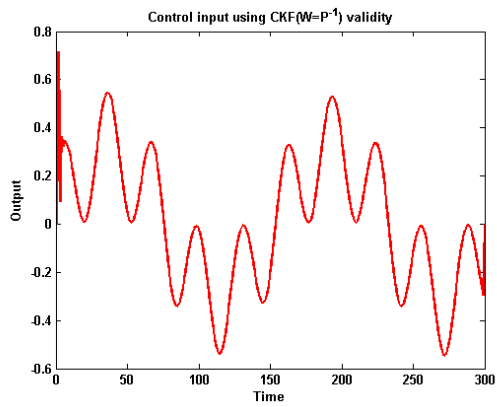
Figure 6.5: Control inputs for Type A controller



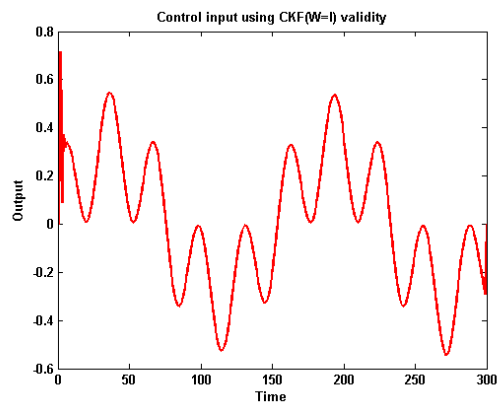
(a) using 2-paramters (CKFP)



(b) using 2-paramters (CKFW)



(c) using 4-paramters (CKFP)



(d) using 4-paramters (CKFW)

Figure 6.6: Control inputs for Type B controller

Example 2

Considered the highly nonlinear benchmark system [170] described by

$$y(k+1) = \frac{u(k)}{1 + y^2(k-1) + y^2(k-2)} + \frac{y(k)y(k-1)y(k-2)u(k-1)(y(k-2) - 1)}{1 + y^2(k-1) + y^2(k-2)} \quad (6.15)$$

This system, with reasonably accuracy, was identified in section 4.4.2 with 2-parameter and 4-parameter structures using four and two submodels, respectively.

It is required that the system track the following reference signal:

$$r(k) = 0.4 \sin\left(\frac{2\pi}{10}k\right) + 0.2 \sin\left(\frac{2\pi}{25}k\right)$$

Table 6.5, Figure 6.7 and Figure 6.8 show the results of using the Type A and Type B controllers for the tracking problem respectively. Figure 6.9 and Figure 6.10 show the control inputs. Judging from the table and figures, both controllers can be said to have reasonably good tracking performance. The trend noticed in the Example 1 can also be observed here. Similarly, from Table 6.6 further optimization of Q and R can be seen to slightly improved the performance of the CKFP especially in the Type B controller.

Table 6.4: Controllers' parameters (λ)

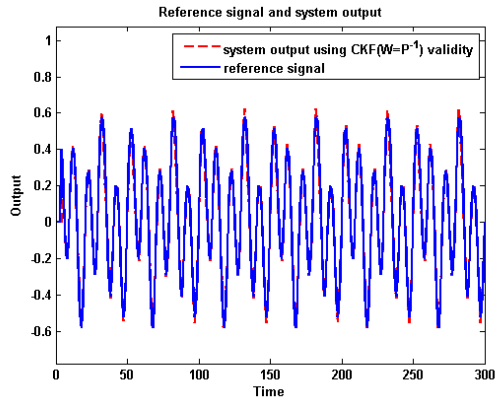
CKF setting	2-parameters structure	4-parameters structure
Type A controller		
CKFP	0.15	0.06
CKFW	0.1	0.06
Type B controller		
CKFP	0.08	0.055
CKFW	0.07	0.065

Table 6.5: Performance of controllers

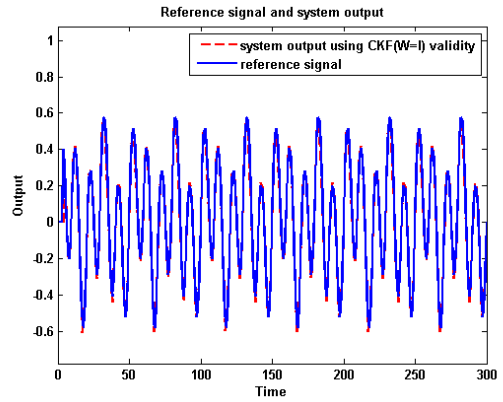
CKF setting	2-parameters structure			4-parameters structure		
	MSE	PMF(%)	VAF(%)	MSE	PMF(%)	VAF(%)
Type A controller						
CKFP	0.0016	87.27	98.46	0.0018	86.31	98.21
CKFW	0.0014	88.04	98.64	0.0019	86.23	98.21
Type B controller						
CKFP	0.0018	86.54	98.28	0.0013	88.47	98.73
CKFW	0.0011	89.60	98.97	0.0013	88.37	98.72

Table 6.6: Performance of controllers for optimized R and Q in CKF

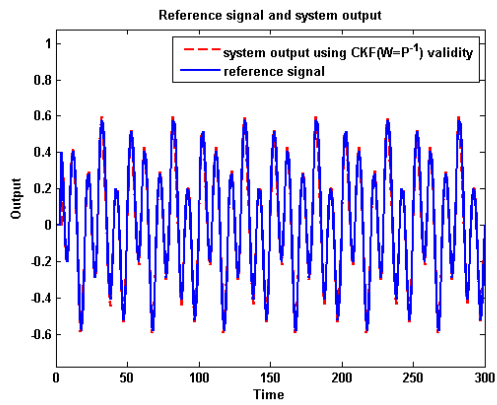
CKF setting	2-parameters structure			4-parameters structure		
	MSE	PMF(%)	VAF(%)	MSE	PMF(%)	VAF(%)
Type A controller						
CKFP	0.0021	92.47	99.49	0.0014	87.91	98.60
CKFW	0.0014	93.83	99.64	0.0014	87.94	98.61
Type B controller						
CKFP	0.0016	87.04	98.41	0.0011	89.20	98.88
CKFW	0.0011	89.60	98.97	0.0012	88.88	98.80



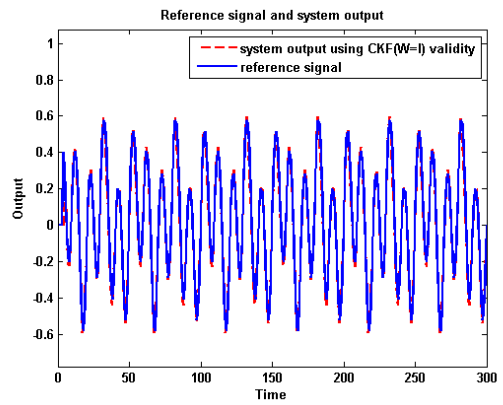
(a) control using 2-paramters structure with CKFP



(b) control using 2-paramters structure with CKFW

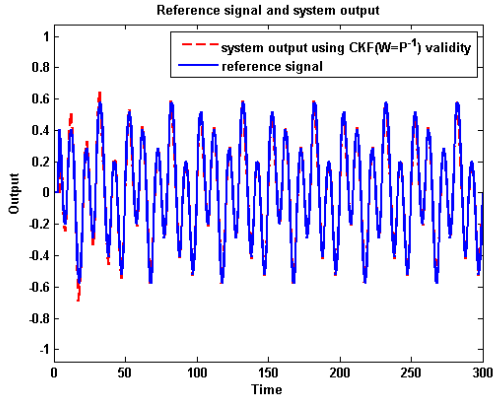


(c) control using 4-paramters structure with CKFP

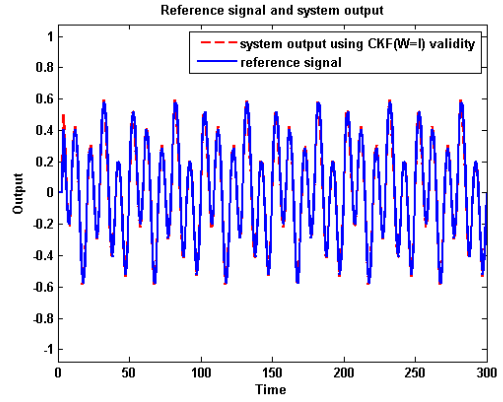


(d) control using 4-paramters structure with CKFW

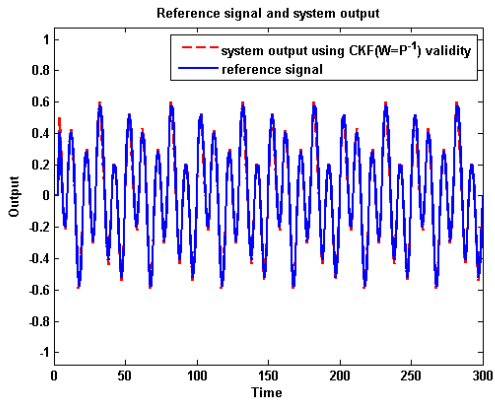
Figure 6.7: Control of time varying system using Type A controller



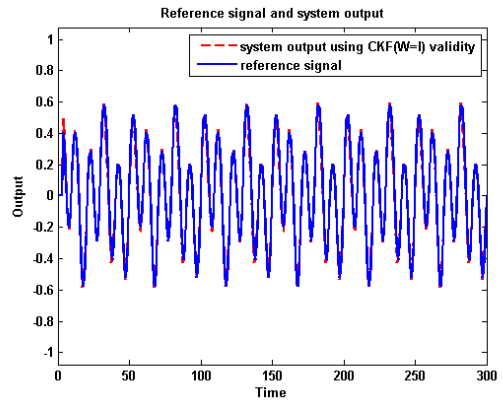
(a) control using 2-paramters structure with CKFP



(b) control using 2-paramters structure with CKFW

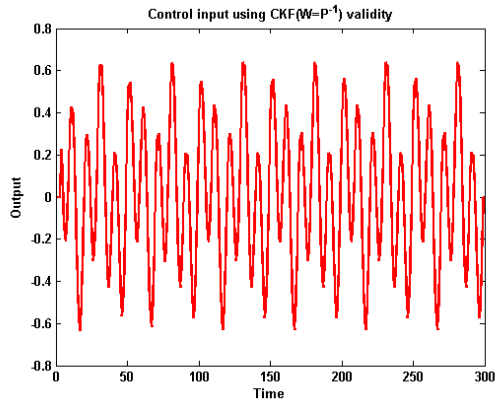


(c) control using 4-paramters structure with CKFP

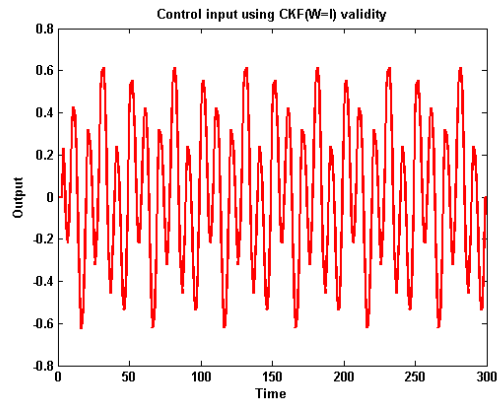


(d) control using 4-paramters structure with CKFW

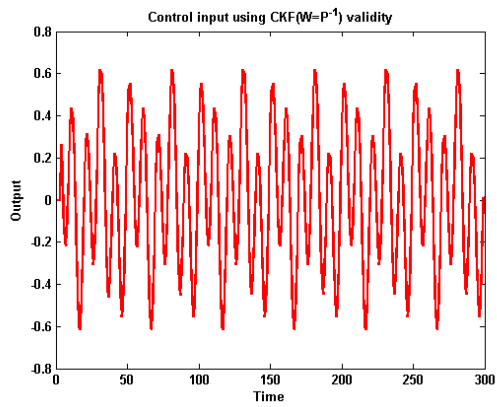
Figure 6.8: Control of time varying system using Type B controller



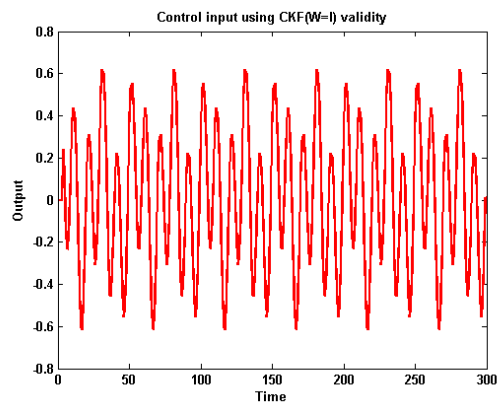
(a) using 2-paramters (CKFP)



(b) using 2-paramters (CKFW)

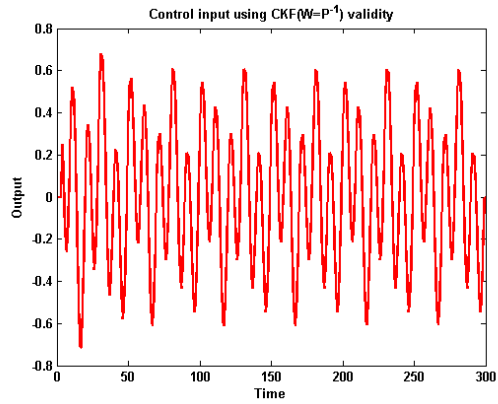


(c) using 4-paramters (CKFP)

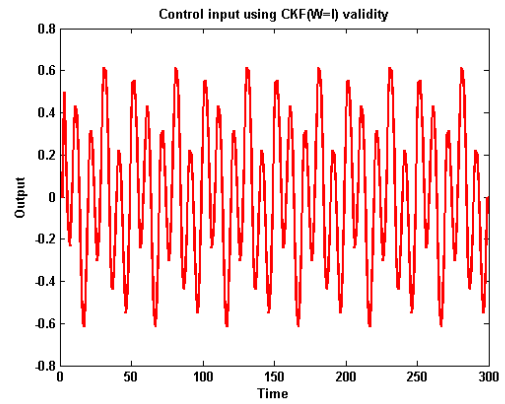


(d) using 4-paramters (CKFW)

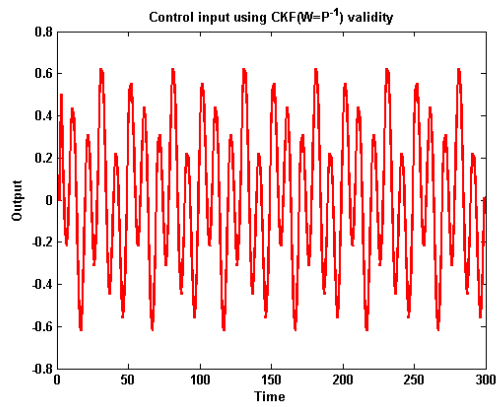
Figure 6.9: Control inputs for Type A controller



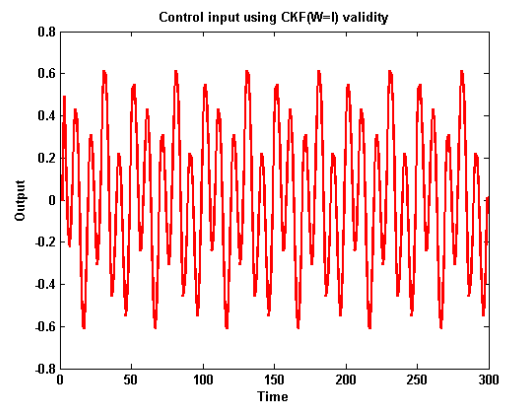
(a) using 2-paramters (CKFP)



(b) using 2-paramters (CKFW)



(c) using 4-paramters (CKFP)



(d) using 4-paramters (CKFW)

Figure 6.10: Control inputs for Type B controller

6.2 Fault detection and Isolation

Knowledge based multi-model method has been employed for fault diagnosis [36, 37, 38]. This approach to fault diagnosis uses a model bank composed of dynamical models of the systems in normal and fault situations, and the integration of these models (see Figure 6.11). According to multi-model concept, the state of the system is determined by running the model bank online and computing the validity of the models. In this way, fault isolation is achieved by representing the individual faults using different models. In what follows, multi-model fault detection and isolation is carried out using the CKF algorithm to compute the validity of the dynamical models of a system in normal and fault modes. Comparative study between the CKF algorithm and two other commonly use validity computations are also carried out.

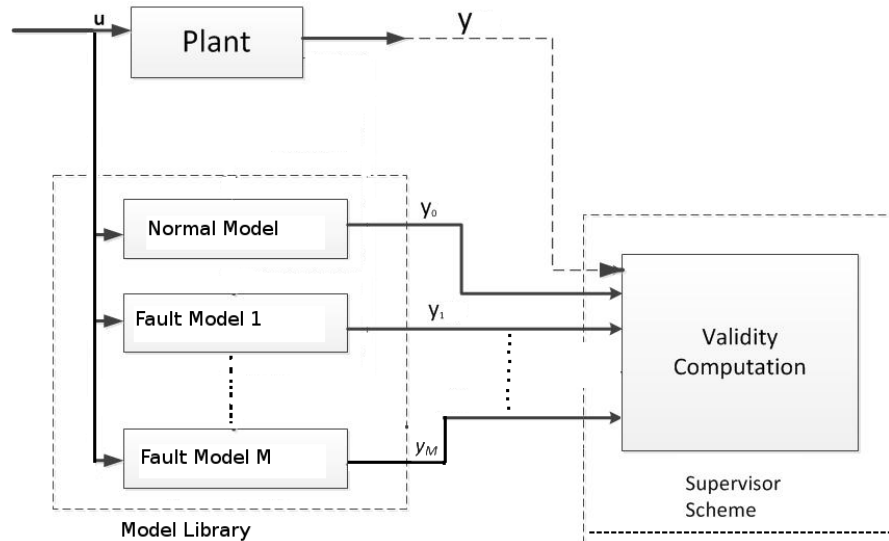


Figure 6.11: Multi-model Fault Diagnosis

Consider a three-tank system shown in Figure 6.12. The mass balance equation

of the system ([208, see]) is represented by

$$\begin{aligned}
S_1 \frac{dh_1(t)}{dt} &= q_1(t) - q_{12}(t) , \\
S_2 \frac{dh_2(t)}{dt} &= q_{12}(t) - q_{23}(t) - q_{10}(t) , \\
S_3 \frac{dh_3(t)}{dt} &= q_2(t) + q_{23}(t) - q_{20}(t) ,
\end{aligned} \tag{6.16}$$

where $h_i(t), i = 1, 2, 3$, is the liquid level in tank i , $S_i, i = 1, 2, 3$, is the cross section of tank i , g is the constant of gravity, and $q_{ij}(t)$ is the flow rate from tank i to j , $i, j = 1, 2, 3$. The flow rate according to Torricelli's rule is given by

$$q_{ij}(t) = \mu_i s_p \text{sign}(h_i(t) - h_j(t)) \sqrt{2g|h_i(t) - h_j(t)|} ,$$

where $\mu_i, i = 1, \dots, 4$ is the output flow coefficient which represents the state of the valve. It has a value of unity when the valve is open and zero otherwise. s_p is the section of all valves, and $q_i(t), i = 1, 2$, is the input flow rate controlled by two pumps. q_{10} and q_{20} represents the outflow rate given by

$$q_{i0}(t) = \mu_i s_p \sqrt{2gh_i(t)} \quad i = 1, 2.$$

The normal mode of this system is to keep constant quantities of water for two consumer's mass through valve v_2 and v_3 . That is, while the two pumps continuously pump in water through q_1 and q_2 , all the four valves are opened. The system output measurement is $y(t) = h_3$. Six failure modes can occur in this

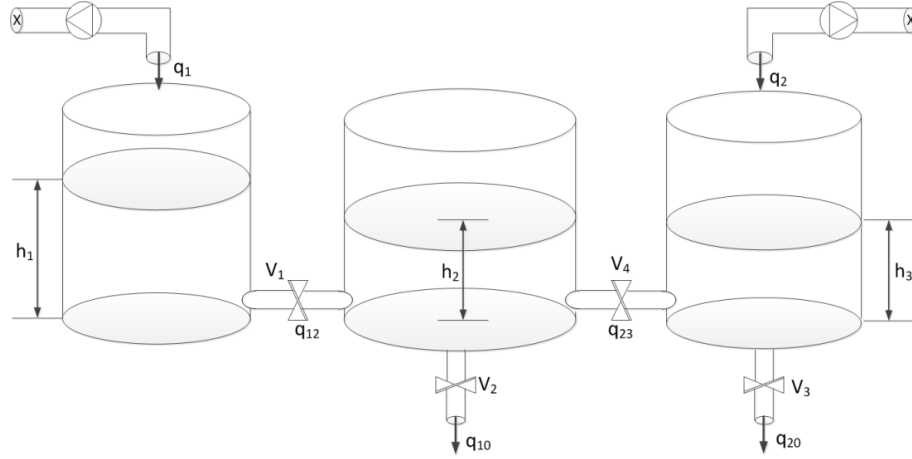


Figure 6.12: The Three-Tanks System

system; two from the failure of the two pumps and four from each of the valves . However, two possible failures are considered in this simulation; failure in the valve v_2 and/or v_3 . This implies that at the point of failure the valve is closed (i.e. $\mu_i = 0, i = 2, 3$). The system is discretized at sampling time of 5s with the parameters; $s_p = 0.00065m^2$, $S_1 = S_3 = 0.0491m^2$, $S_2 = 0.0616m^2$, $q_1 = 0.000154m^3s^{-1}$, $q_1 = 0.00025m^3s^{-1}$, $g = 9.81Nm^2$, $h(0) = [0.40116 \ 0.10249 \ 0.05124]$.

In the simulation, model bank consisting of models of normal system mode and the two fault mode are compared with the simulated system output to determine not only the existence of a fault but also the kind of fault. Figure 6.13 shows the schematic diagram of a multi-model fault detection and isolation. Different validity computations namely, simple residue, reinforced residue and the CKF algorithms are examined to detect possible fault in the system. The fault diagnosis simulation for the two faults are given below.

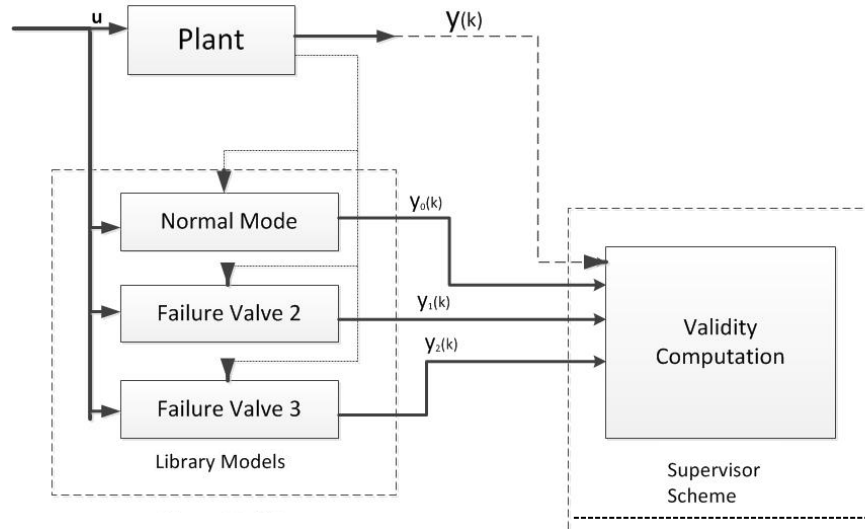


Figure 6.13: Multi-model Fault Detection of Three-Tanks System

Normal mode

In this case the system is allowed to run in normal mode without any failure. The model bank was run online along with the simulated system to compute the contribution of each model (validity computation) as shown in Figure 6.13 . The profile of the validity computations from aforementioned algorithms are shown in Figure 6.14. It is observed from the figures that all the validities reached their maximum value of unity except with simple validity with value of 0.5. There is however slow convergence of CKF with $W = I$ due to the high weight.

Failure of valve v_2

In this case the system is allow to run with fault in v_2 . The failure of valve v_2 is simulated at time instant of 150 seconds. It is observed from Figure 6.15 that the validity profile allows us to determine that there is a fault at valve v_2 in all the validity computations. This can be noticed at instant 155 seconds by the

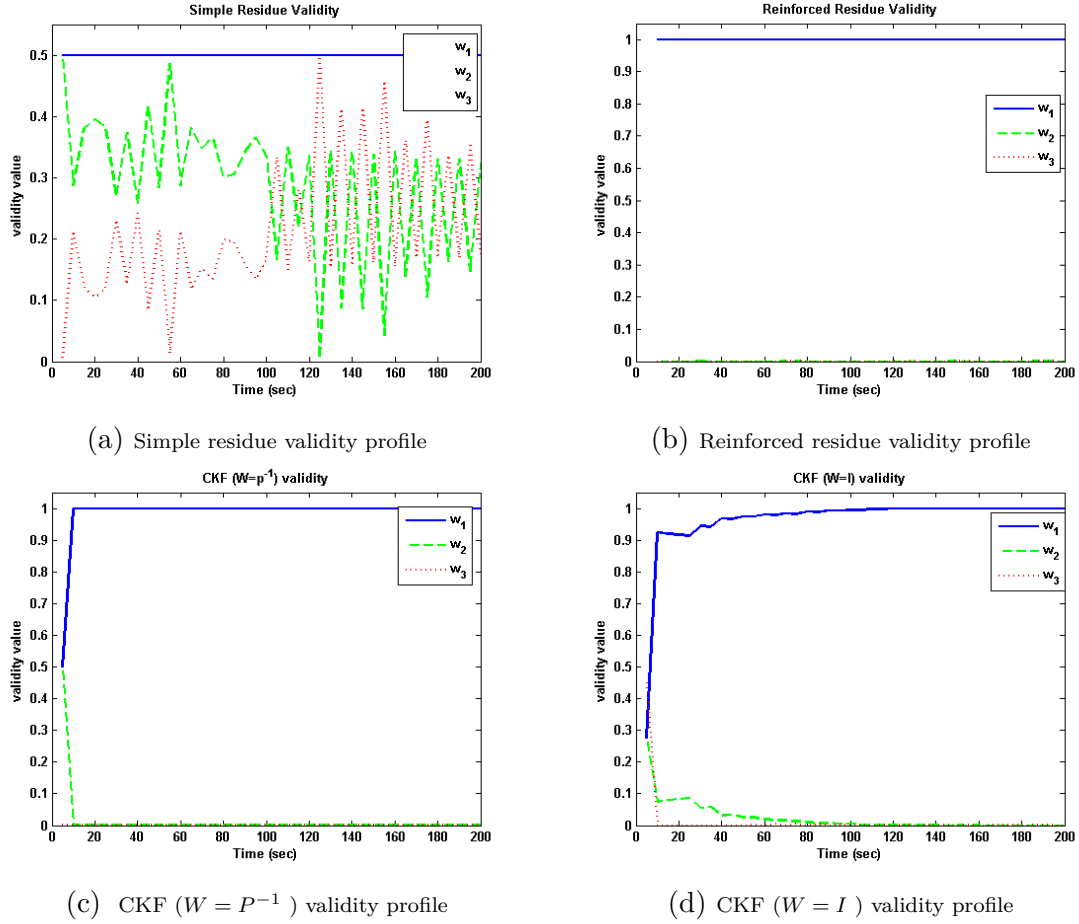


Figure 6.14: Validity profiles for normal mode

simple and reinforced residue, instant 170 seconds by CKF setting $W = P^{-1}$, and instant 175 seconds by CKF setting $W = I$. Irrespective of the time they reach the maximum value, we can notice that the validity of normal model starts decreasing at exact time instance of failure, and the validity of the v_2 fault model starts increasing. This is essentially needed for a good fault detection. Hence, the state of the system can be determined by the model that has validity value of at least 0.6 at any time instant.

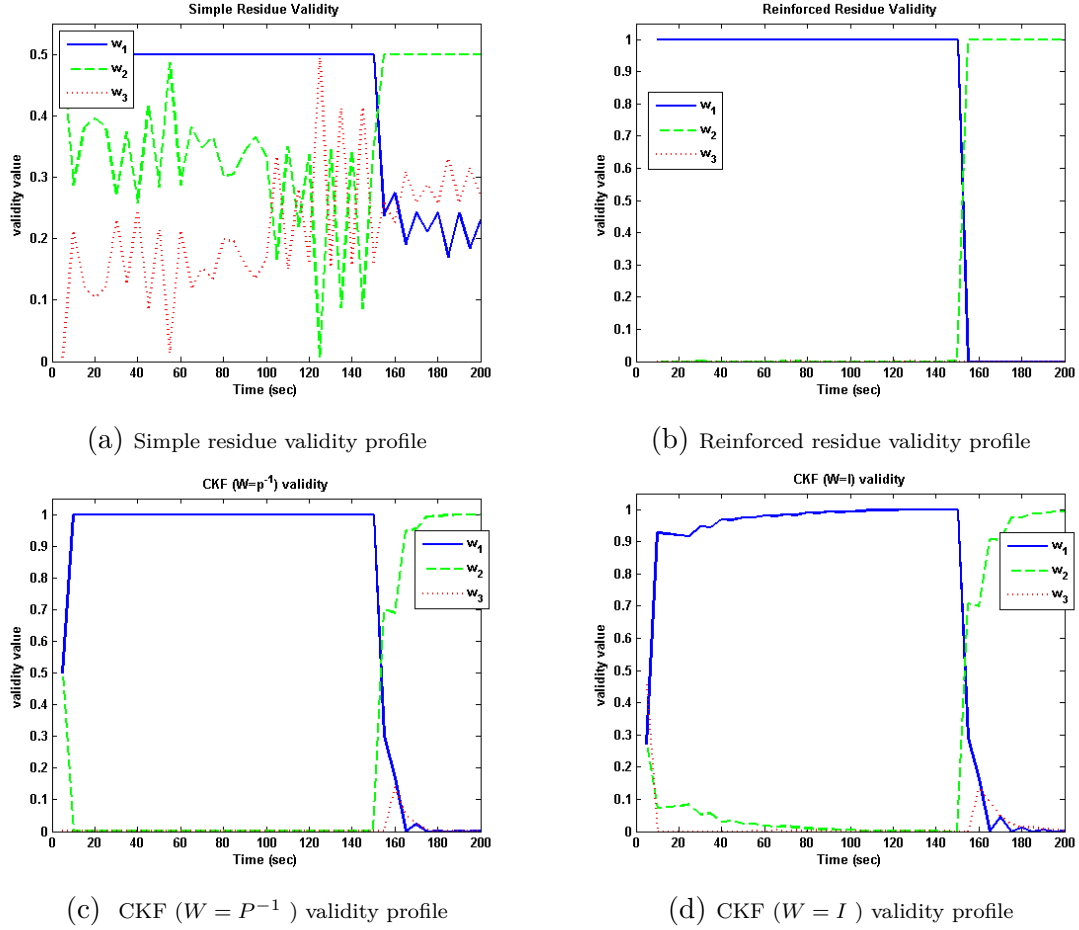


Figure 6.15: Validity profiles for fault on valve v_2

Failure of valves v_2 and v_3 at different time

Here failure of valve v_2 and valve v_3 are considered at time instant of 75 and 120 seconds respectively. The result is as shown in Fig. 6.16. All the validity computation were able to detect the fault at the exact instant by the decrement of validity value of the system's previous state. However, similar to previous fault, it is observed that CKF shows slow convergence rate to reach the maximum value of unity.

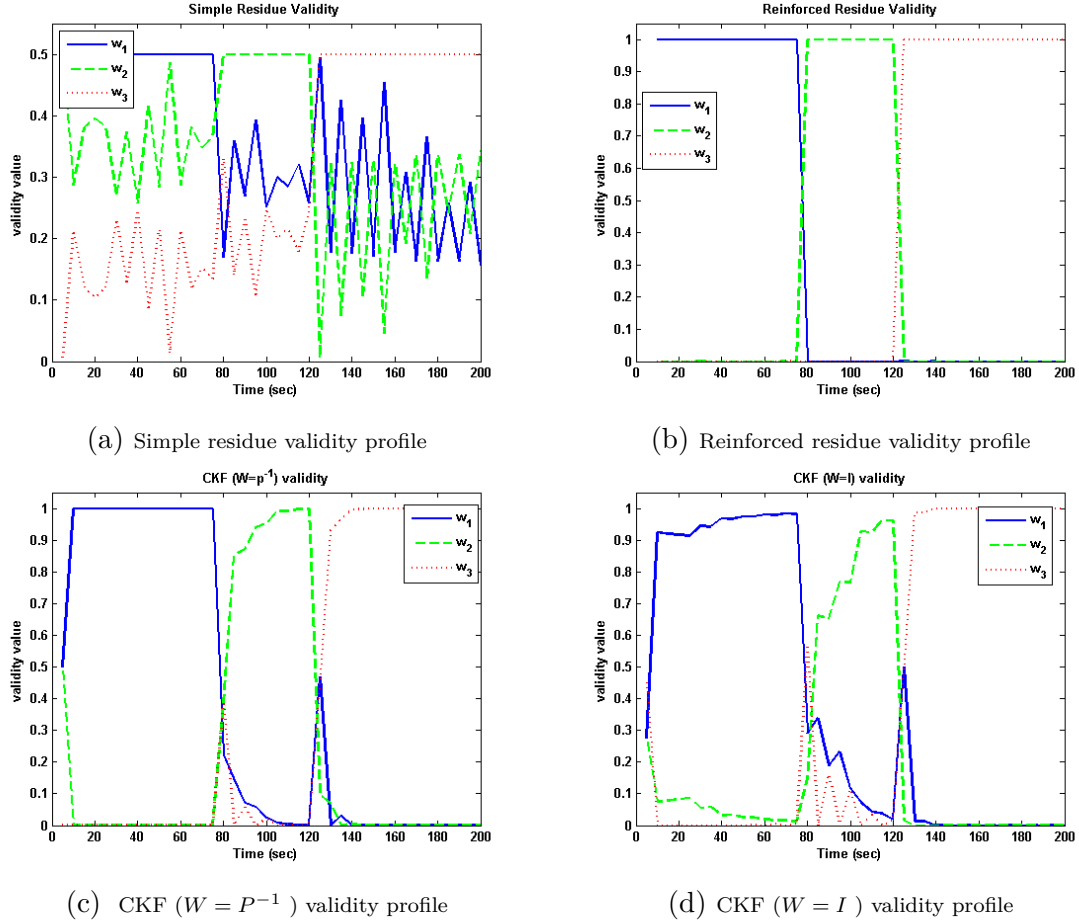


Figure 6.16: Validity profiles for fault on valve v_2 and v_3 at different time

Failure of valves v_2 and v_3 simultaneously

In this scenario valve v_2 failed at time instant of 75 seconds while both valves (v_2 and v_3) failed at time instant of 120 seconds. Note that we do not have model for simultaneous fault of v_2 and v_3 in the model library. As shown in Fig. 6.17 all the validity computation were able to detect the simultaneous fault differently despite non availability of its model in the library. It is however noted that the CKF validity profiles are more reasonable than others as they show fluctuations in normal mode of operation when both faults occur simultaneously which represents a good indication of the presence of problem in the system. In

the case of reinforced validity profile, the result shows that none of the model represents the real system behavior (validity value of all the library models are less than 0.5).

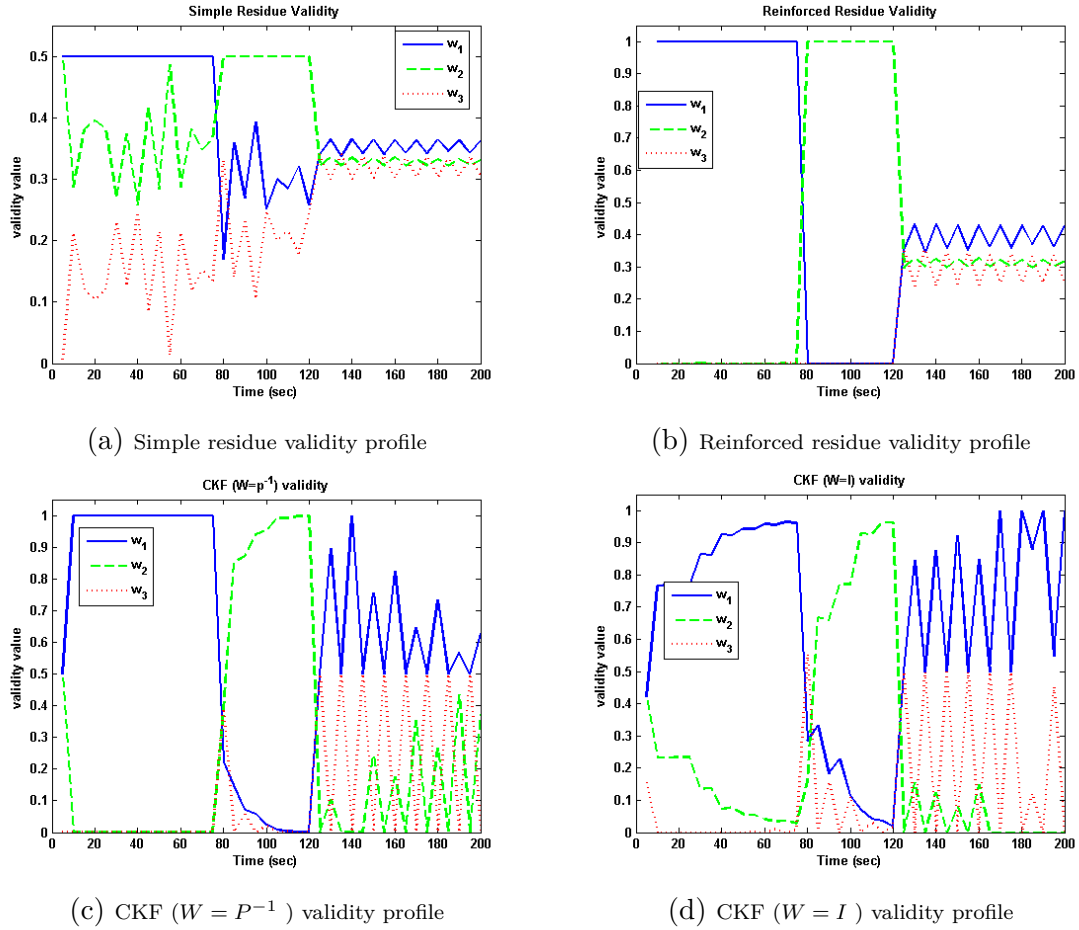


Figure 6.17: Validity profiles for fault on valves v_2 and v_3 simultaneously

Failure of valves v_2 with noise on the system output

In this case the previous simulation of failure of valve v_2 at instant of 150 seconds is carried out with noise added to the output of the system at very instant. The noise generated is normally distributed random noise scaled at some percentage of the system output. The scale of 5% to 30% is used. For the 5% scale all

the four validity computations were able to detect the fault, though a little false detection (indicated with red circle) can be noticed in the profile of simple and reinforced residue validity computation as shown in Figure 6.18a and Figure 6.18b respectively. The results of 10% and 20% scaled noise are as shown in Figure 6.19 and Figure 6.20 respectively. One can observe that as the percentage of noise increases, the false detections (indicated with red circle) in the profile of simple and reinforced residue validity computations also increases. However, as for the CKF validity computations there is no false detection can be noticed (see Figures 6.19c, 6.19d, 6.20d, and 6.20c). Although it is becoming difficult for the validity of the library models to reach their maximum value in both CKF validity settings, this does not make the fault decision obscure. As explained earlier, the state of the system can be determine by validity above 0.5. Therefore, unlike simple and reinforced residue, CKF validity computation is robust to noise environment which is common in many industrial processes.

In summary, the CKF validity computation has been found to be adequate for fault diagnosis. Although it is observed that it has slow convergence to reach its maximum validity value (one), this does not degrade its performance as the state of the system can be determine at any instance. Moreover, the CKF validity has shown to be robust to noise which is very common in the practical process environments.

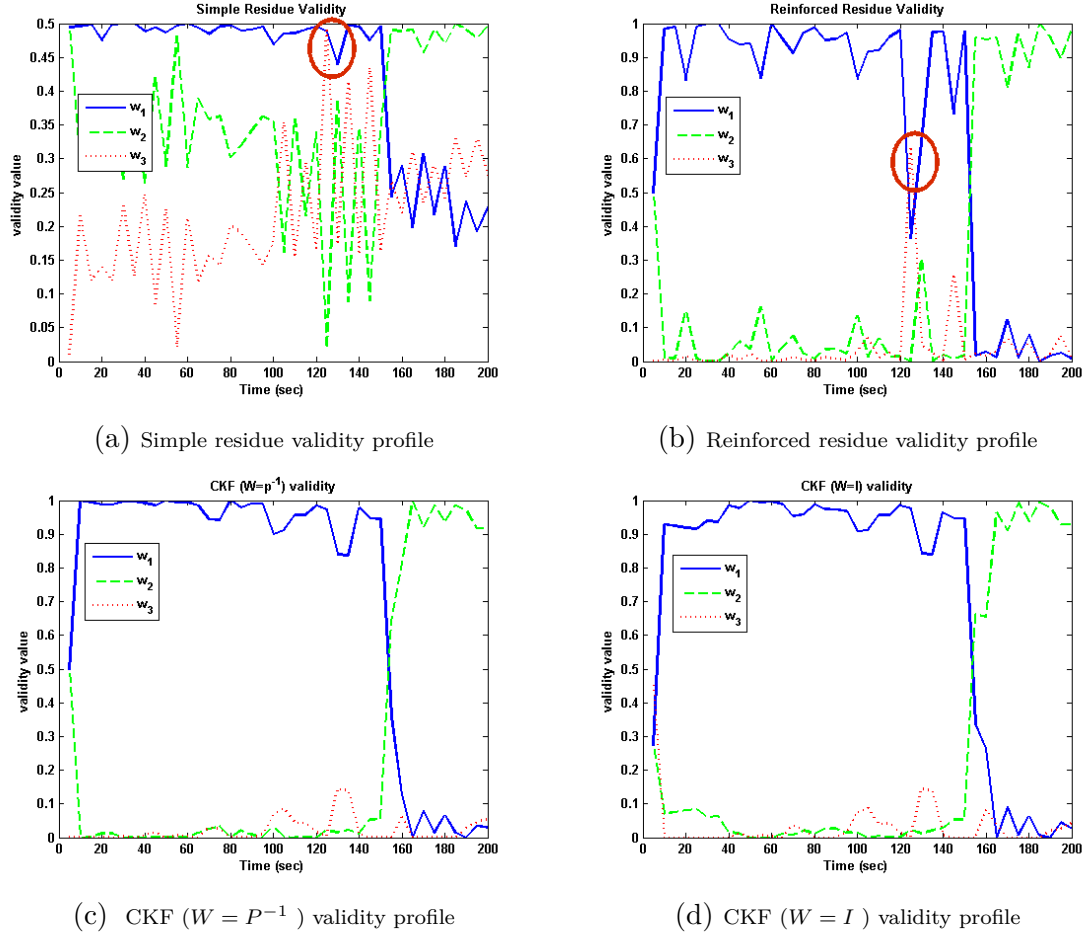
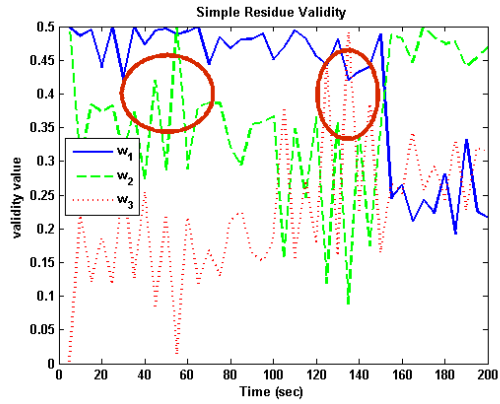


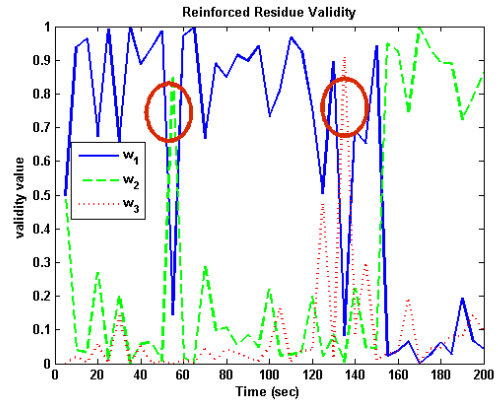
Figure 6.18: Validity profiles for fault on valve v_2 with 5% scaled noise

6.3 Conclusion

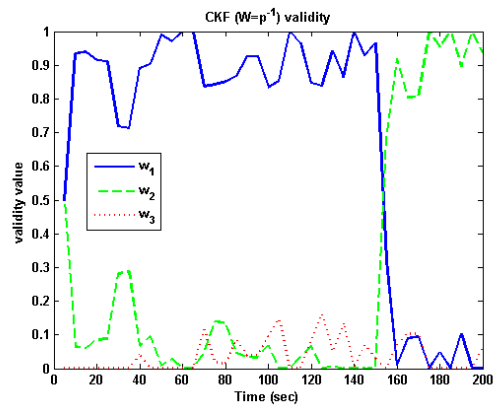
Chapter 6 presents multi-model control and fault diagnosis of a complex nonlinear system. In the control case, two multi-model controller schemes were designed for tracking problem, and were found to have good performance. In the fault diagnosis application, the CKF validity computation was investigated on a three tank system to detect and isolate the system fault. Simulation results show that CKF algorithm is of good performance and better than other commonly used validity estimation methods. Also, it has shown to be robust to noisy situation.



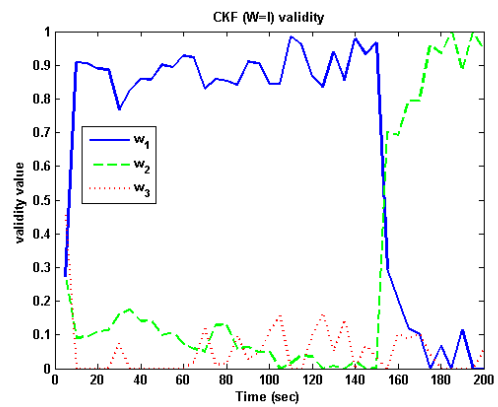
(a) Simple residue validity profile



(b) Reinforced residue validity profile

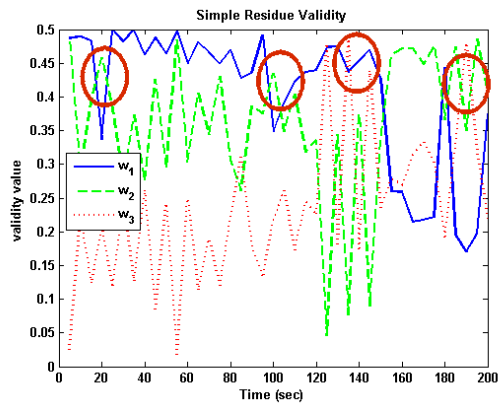


(c) CKF ($W = P^{-1}$) validity profile

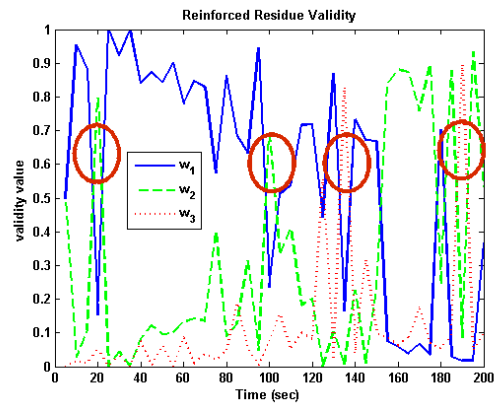


(d) CKF ($W = I$) validity profile

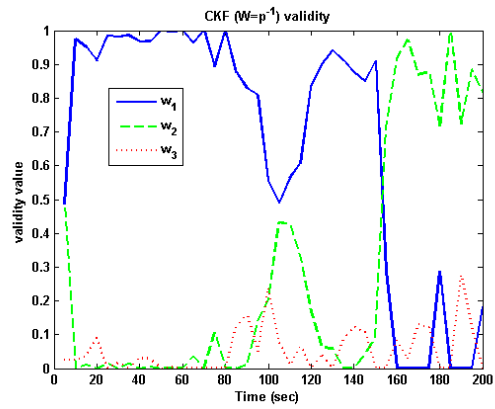
Figure 6.19: Validity profiles for fault on valve v_2 with 10% scaled noise



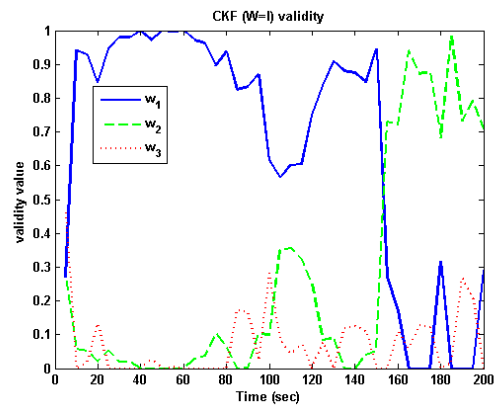
(a) Simple residue validity profile



(b) Reinforced residue validity profile



(c) CKF ($W = P^{-1}$) validity profile



(d) CKF ($W = I$) validity profile

Figure 6.20: Validity profiles for fault on valve v_2 with 20% scaled noise

CHAPTER 7

CONCLUSION AND RECOMMENDATION

" Research never ends but Researchers do".....Anonymous

Chapter 7 summarizes the contributions of the thesis to the existing research works on multi-model identification of nonlinear systems, and points out the possible future extensions of the research.

7.1 Main Contribution

Several industrial systems are characterized by high nonlinearities with wide operating ranges and large set point changes. Identification and representation of these systems represent a challenge especially for the control engineers. In recent years, much attention has been given to the multi-model-based alternative approach to describe nonlinear systems. In contrast to conventional modeling technique, a system is represented by a set of models, that are combined, in a certain way,

to form the global model. This intuitive idea comes with several paradigms such as T-S fuzzy model, piecewise continuous model (PWC), piecewise affine (PWA), linear parameter varying model (LPV), local model network (LMN), etc.. Owing to its potential benefits, this effective field of research has received several contributions, and has gained lots of interest in many application areas, particularly in control and fault diagnosis. Despite its benefits, the approach still faces several challenges.

This dissertation focuses on identification of nonlinear systems using multi-model approach, as well as its application for control and fault diagnosis. To this end, background and detailed review to interpolated multi-model representation of nonlinear system was provided in chapter 2. An exposition and review of recent developments to the design challenges encountered in multi-model framework for modeling and identification of nonlinear system is presented. Rather than enumerating all methodologies in the area, we have focused on three key challenging areas of multi-model design namely: operating space partition strategies, internal structure and parameter estimation as well as validity computation.

One major challenge of multi-model approach is the partitioning of the system's operating space to a number of sub-spaces. This translate to finding the submodels that can adequately represent the entire operating region of the nonlinear system. In chapter 3, a two-stage method to obtain the partition and parameters of the submodels in the multi-model representation is presented. In the first stage, we modified the combinatorial Particle Swarm Optimization (CPSO)

to find the number of submodels and their initial parameters. Hybrid K-means algorithm was used to obtain more efficient submodels in the second stage. This proposed approach provided us with automatic optimization of the number of submodels with respect to submodel complexity. This implies that the original nonlinear system can be partitioned into a parsimonious number of submodels and the structure of the submodels can be assumed a priori. Furthermore, the partition and selection of number of submodels is not only based on data distribution but also on the linearity of the operating region, and that of the assumed submodel structure. Indeed this made the submodels to have few effective parameters, stable with better accuracy. Benchmark simulation examples were provided to illustrate the effectiveness of the proposed method.

Chapter 4 deals with the issue of combining the submodels to completely form multi-model representation of the nonlinear system. Although during simulation studies in chapter 3 the submodels estimated were interpolated using existing validity computations in the literature. These methods have been reported to have some drawback such as lack of precision, sensitivity to parameter selection, and restriction to partition strategy. In order to overcome some of these drawbacks, constrained Kalman filter (CKF) was developed for interpolating the outputs of the submodels. The algorithm was obtained by reformulating the multi-model output equation as an estimation problem. Extensive simulation studies indicated that CKF algorithm can indeed contributes towards accurate identification of nonlinear systems. This was demonstrated by comparison studies between the

CKF and other commonly used methods.

In chapter 5, the modified combinatorial Particle Swarm Optimization algorithm in chapter 3 was extended to include estimation of order of the submodels (number of parameters), which was assumed known previously.

In chapter 6, applications of our proposed multi-model framework were investigated for nonlinear systems control and fault diagnosis. For the control application, two multi-model controller schemes, based on weighted one-step-ahead controller design, were derived to control nonlinear systems for reference tracking. The first scheme presented a fusion of the multiple weighted one-step-ahead controller outputs while in the second, fusion of submodels' parameters was used to derive a weighted one-step-ahead controller. The weights needed for the interpolation in both schemes were generated online using the CKF validity. Simulation studies on the previously identified systems verified the efficacy of both the control schemes. In the fault diagnosis application, the suitability of the CKF algorithm under multi-model framework was tested for fault detection and isolation on a three-tank system.

7.2 Future Recommendation

Some areas for further research may include the following:

- In chapter 2 the submodels were identified with input-output models. We recommend extending the algorithm presented to be able to cope with state space models, which is model suitable for multi-input multi-output systems.

This can be done for example by incorporating algorithm like the N4SID into the framework presented.

- We observed that using the CKF algorithm for validity estimation can sometime defy the known principle that the interpolated output will be unstable if atleast one of the submodels is unstable. This has to be investigated further as we noticed that sometimes the interpolated output is stable when only one submodel is unstable.
- Only the offline estimation of the submodels is addressed in this thesis. An important area that should be addressed in future research is online partition scheme for online identification of submodels. Although for the approach that were presented, the submodels could be updated online by using the cluster center of the partition that is the closest to a new observation, direct online partition are needed for higher dimensional problems.
- Application consideration in operating space partition may also needs to be given more attention. This may allow the submodels created, for example in control, to have inherent closed loop characteristics that will enhance control performance.
- Multi-model is versatile approach that can handle many real life process. therefore, this approach should be exploited in other application areas like, fault tolerance control, multiphase flow, systems' health management etc

Although there remains much to do, significant steps towards improving the methodology of multi-model identification, control and fault diagnosis for complex nonlinear dynamic systems have been presented in this work.

REFERENCES

- [1] L. Ljung, Ed., *System Identification (2Nd Ed.): Theory for the User*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.
- [2] W. Greblicki, “Nonparametric identification of wiener systems by orthogonal series,” *IEEE Transactions on Automatic Control*, vol. 39, no. 10, pp. 2077–2086, October 1 1994.
- [3] M. Pottmann and R. K. Pearson, “Block-oriented narmax models with output multiplicities,” *AIChE Journal*, vol. 44, pp. 131–140, 1998.
- [4] E. Eskinat, S. H. Johnson, and W. L. Luyben, “Use of hammerstein models in identification of nonlinear systems,” *AIChE Journal*, vol. 37, no. 2, pp. 255–268, February 1991.
- [5] R. K. Pearson and M. Pottmann, “Gray-box identification of block-oriented nonlinear models,” *Journal of Process Control*, vol. 10, pp. 301–315, 2000.
- [6] H. Kashiwagi and L. Rong, “Identification of volterra kernels of nonlinear van de vusse reactor,” *Trans. on Control, Automation, and Systems Engineering*, vol. 4, no. 2, pp. 109–113, June 2002.

- [7] L. Piroddi and W. Spinelli, “An identification algorithm for polynomial narx models based on simulation error minimization,” *International Journal of Control*, vol. 76, no. 17, pp. 1767–1781, September 2003.
- [8] S. Chen, S. A. Billings, and W. Luo, “Orthogonal least squares methods and their application to nonlinear system identification,” *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896, November 1989.
- [9] A. Gretton, A. Doucet, P. J. W. Herbrich, R. and Rayner, and B. Scholkopf, “Support vector regression for black- box system identification,” in *Proc. of the 11th IEEE Workshop on Statistical Signal Processing*, 2001, pp. 341–344.
- [10] S. A. Billings and H. L. Wei, “A new class of wavelet networks for nonlinear system identification,” *IEEE Trans. on Neural Networks*, vol. 16, no. 4, pp. 862–874, July 2005.
- [11] R. DeVeaux, D. Psychogios, and L. U. , “A comparison of two nonparametric estimation schemes: Mars and neural networks,” *Computers in Chemical Engineering*, vol. 17, no. 8, pp. 819–837,, 1993.
- [12] H. Zhao and J. Zhang, “Nonlinear dynamic system identification using pipelined functional link artificial recurrent neural network,” *Neurocomputing*, vol. 72, no. 13–15, pp. 3046 – 3054, 2009, hybrid Learning Machines (HAIS 2007) / Recent Developments in Natural Computation (ICNC 2007).
- [13] B. Majhi and G. Panda, “Robust identification of nonlinear complex systems

- using low complexity {ANN} and particle swarm optimization technique,” *Expert Systems with Applications*, vol. 38, no. 1, pp. 321 – 333, 2011.
- [14] I. M. Yassin, M. N. Taib, and R. Adnan, “Recent advancements & methodologies in system identification: A review,” *Scientific Research Journal (SCIRJ)*, vol. 1, no. 1, pp. 14–33, 2013.
- [15] T. Johansen and B. Foss, “A narmax model representation for adaptive control based on local models,” *Modeling Identification and Control*, vol. 13, pp. 25–39, 1992.
- [16] —, “Operating regime based process modeling and identification,” *Computers and Chemical Engineering*, vol. 21, pp. 159–176, 1997.
- [17] M. J. Dunlop, E. Franco, and R. M. Murray, “Multi-model approach to identification of biosynthetic pathways,” in *Proceedings of the 2007 American Control Conference*, Marriott Marquis Hotel at Times Square New York City, USA, July 11-13 2007.
- [18] Z.-K. Xue and S.-Y. Li, “A multi-model identification algorithm based on weighted cost function and application in thermal process,” *ACTA Automatica, SINICA*, vol. 31, no. 3, 2005.
- [19] T. Abbas and F. Tufvesson, “System identification in gsm/edge receivers using a a multi-model approach,” *ACEEE International Journal on Control System and Instrumentation*, vol. 3(1), pp. 41 – 46, 2011.

- [20] L. Gan and L. Wang, “A multi-model approach to predictive control of induction motor,” in *IECON 2011- 37th Annual Conference on IEEE Industrial Electronics Society*, Melbourne, VIC, 7-10, Nov. 2011, pp. 1704 – 1709.
- [21] M. Ltaief, K. Abderrahim, and R. Ben Abdennour, “Contributions to the multimodel approach: systematic determination of a models’ base and validities estimation,” *Journal of Automation & Systems Engineering*, vol. 2, no. 3, 2008.
- [22] M. Ronen, Y. Shabtai, and H. Guterman, “Hybrid modeling building methodology using unsupervised fuzzy clustering and supervised neural networks,” *Biotechnol Bioeng*, vol. 77, no. 4, pp. 420–429, 2002.
- [23] K. Azman and J. Kocijan, “Dynamical systems identification using gaussian process models with incorporated local models,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 2, pp. 398–408, 2011.
- [24] T. Johansen and B. Foss, “Identification of non-linear system structure and parameters using regime decomposition,” *Automatica*, vol. 31, pp. 321–326, 1995.
- [25] R. Babuska and H. Verbruggen, “Neuro-fuzzy methods for nonlinear system identification,” *Annual Reviews in Control*, vol. 27, pp. 73–85, 2003.
- [26] S. A. Billings, *Models for Linear and Nonlinear Systems*. Wiley Online Library, 2013.

- [27] T. Johansen and B. Foss, “State-space modeling using operating regime decomposition and local models,” in *Preprints 12th IFAC World Congress*, vol. 1, Sydney, Australia, 19-23, July 1993, pp. 431–434.
- [28] O. Nelles, A. Fink, and R. Isermann, “Local linear model trees (lolimot) toolbox for nonlinear system identification,” in *In: 12th IFAC Symposium on System Identification (SYSID)*, Santa Barbara, USA,, 2000, p. 845–850.
- [29] O. Nelles, S. Sinsel, and R. Isermann, “Local basis function networks for identification of a turbocharger,” in *In Proc. UKACC Int. Conf. Control*, vol. 1, Exeter, U.K., Sep. 1996, p. 7–12.
- [30] T. Tanaki and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, pp. 116–132, 1985.
- [31] J. Xu, X. Huang, and S. Wang, “Nonlinear model predictive control using adaptive hinging hyperplanes model,” in *Proc. of the 48th IEEE Conf. on Decision and Control*, Shanghai, China, December 2009, pp. 2598–2603.
- [32] B. Hartmann, O. Nelles, A. Belič, and D. Zupančič–Božič, “Local model networks for the optimization of a tablet production process,” in *Artificial Intelligence and Computational Intelligence*. Springer Berlin Heidelberg, 2009, vol. 5855, p. 241–250.
- [33] G. Cai1, G. Duan, and C. Hu, “A velocity-based LPV modeling and control

- framework for an airbreathing hypersonic vehicle,” *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 5(A), May 2011.
- [34] H. T. Toivonen, K. V. Sandstrom, and R. H. Nystrom, “Internal model control of nonlinear systems described by velocity-based linearizations,” *Journal of Process Control*, vol. 13, p. 215–224, 2003.
- [35] R. Selmic and F. Lewis, “Multimodel neural networks identification and failure detection of nonlinear systems,” vol. 4, December 2001, pp. 3128–3133.
- [36] J. Li, C. Bo, J. Zhang, and J. Du, “Fault diagnosis and accommodation based on online multi-model for nonlinear process,” *Computational Intelligence*, vol. 1, p. 661–666, 2006.
- [37] Y. Diao and K. M. Passino, “Fault diagnosis for a turbine engine,” *Control Engineering Practice*, vol. 12, p. 1151–1165, 2004.
- [38] G. G. Yen and L.-W. Ho, “Online Multiple-Model-Based Fault Diagnosis and Accommodation,” *IEEE Transaction on Industrial Electronics*, vol. 50, no. 2, APRIL 2003.
- [39] J. Tirado, D. Higuero, F. Isaila, and J. Carretero, “Multi-model prediction for enhancing content locality in elastic server infrastructures,” in *18th International Conference on High Performance Computing (HiPC)*, Bangalore, 18-21 Dec. 2011.

- [40] R. Gao and A. O. Dwyer., “Multiple model networks in non-linear system modelling for control – a review,” *DIT*, 2002.
- [41] G. Gregorčič and G. Lightbody, “Nonlinear system identification: From multiple-model networks to gaussian processes,” *Engineering Applications of Artificial Intelligence*, vol. 2, pp. 1035–1055, Oct. 2008.
- [42] D. Filev, “Fuzzy modeling of complex systems,” *International Journal of Approximate Reasoning*, vol. 5, no. 3, pp. 281–290, 1991.
- [43] R. Orjuela, B. Marx, J. Ragot, and D. Maquin, “Nonlinear system identification using heterogeneous multiple models,” *Int. J. Appl. Math. Comput. Sci.*, vol. 23, no. 1, p. 103–115, 2013.
- [44] S. C. McLoone, G. W. Irwin, and S. F. McLoone, “Constructing networks of continuous-time velocity-based models,” in *IEEE Proceedings Control Theory and Applications*, vol. 148, no. 5, 2001, p. 397–405.
- [45] D. J. Leith and W. E. Leithead, “Analytic framework for blended multiple model systems using linear models,” *International Journal of Control*, vol. 72, pp. 605–619, 1999.
- [46] S. E. Chouaba, A. Chamroo, R. Ouvrard, and T. Poinot, “A counter flow water to oil heat exchanger: {MISO} quasi linear parameter varying modeling and identification,” *Simulation Modelling Practice and Theory*, vol. 23, no. 0, pp. 87–98, 2012.

- [47] K. B. B. Vinsonneau, D.P. Goodall, “Extended global total least square approach to multiple model identification,” in *16th, IFAC World Congress*, Prague, Czech Republic, 2005.
- [48] Y. Zhu and G. L. Ji, “LPV model identification using blended linear models with given weightings,” in *In proc. 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 6-8 2009, pp. 1674–1679.
- [49] D. Driankov, R. Palm, and U. Rehfuss, “A takagi-sugeno fuzzy gain-scheduler,” in *IEEE Conf. Fuzzy Systems*, New Orleans, 1996, pp. 1053–1059.
- [50] T. Johansen, K. Hunt, P. Gawthrop, and H. Fritz, “Off equilibrium linearisation and design of gain-scheduled control with application to vehicle speed control,” *Control Engineering Practice*, vol. 6, no. 2, pp. 167–180, 1998.
- [51] O. Galan, J. A. Romagnoli, A. Palazoğlu, and Y. Arkun, “Gap metric concept and implications for multilinear model-based controller design,” *Industrial & Engineering Chemistry Research*, vol. 42, no. 10, pp. 2189–2197, 2003.
- [52] J. Du, C. Song, and P. Li, “Application of gap metric to model bank determination in multilinear model approach,” *Journal of Process Control*, vol. 19, no. 2, pp. 231–240, 2009.
- [53] —, “Multimodel control of nonlinear systems: an integrated design proce-

- ture based on gap metric and h_{inf} loop shaping,” *Industrial & Engineering Chemistry Research*, vol. 51, no. 9, pp. 3722–3731, 2012.
- [54] J. Du, C. Song, Y. Yao, and P. Li, “Multilinear model decomposition of {MIMO} nonlinear systems and its implication for multilinear model-based control,” *Journal of Process Control*, vol. 23, no. 3, pp. 271 – 281, 2013.
- [55] W. Tan, H. Marquez, and T. Chen, “Operating point selection in multimodel controller design,” in *American Control Conference, 2004. Proceedings of the 2004*, vol. 4, June 2004, pp. 3652–3657 vol.4.
- [56] W. Tan, H. J. Marquez, T. Chen, and J. Liu, “Multimodel analysis and controller design for nonlinear processes,” *Computers & Chemical Engineering*, vol. 28, no. 12, pp. 2667 – 2675, 2004.
- [57] S. C. McLoone, G. W. Irwin, and S. F. McLoone, “On gaussian weighting functions for velocity-based local model networks,” in *In Proceedings of the Irish Signals and Systems Conference ISSC2002*, Cork, Ireland, 2002, p. 135–140.
- [58] R. Gao and A. O. Dwyer, “Discrete-time velocity-based multiple model networks,” in *Portuguese Conference on Automatic Control*, v, September 2002, pp. 289–294.
- [59] K. Azman and J. Kocijan, “Fixed-structure gaussian process model,” *International Journal of Systems Science*, vol. 40, no. 12, pp. 1253–1262, 2009.

- [60] S. Kawamoto, K. Tada, A. Ishigame, and T. Taniguchi, “An approach to stability analysis of second order fuzzy systems,” in *IEEE International Conference on fuzzy Systems*, May 1992, pp. 1427–1434.
- [61] H. Wang, K. Tanaka, and M. Griffin, “An approach to fuzzy control of nonlinear systems: Stability and design issues,” *IEEE Trans. Fuzzy Syst*, vol. 4, pp. 14–23, 1994.
- [62] P. Bergsten and R. Palm, “Thau-luenberger observers for ts fuzzy systems,” in *The 9th IEEE Int. Conf. Fuzzy Systems*, San Antonio, Texas, 2000.
- [63] G. Angelis, M. van de Molengraft, J. Verstraete, and J. Kok, “Polytopic linear modeling of a class of nonlinear systems: an automatic model generating method,” in *Proceedings of the third MathMod, Report No.15*, vol. 1, Vienna, Austria, 2000, pp. 273–276.
- [64] M. Chadli, D. Maquin, and J. J. Ragot, “On the stability analysis of multiple model systems,” in *European Conference Control (ECC)*, Porto, Portugal, 2001.
- [65] A. Akhenak, M. Chadli, J. Ragot, and D. Maquin, “Estimation of state and unknown inputs of a nonlinear system represented by a multiple model,” in *11th IFAC Symposium on Automation in Mining, Mineral and Metal processing, MMM*, Nancy, France, 2004.
- [66] D. Ichalal, B. Marx, J. Ragot, and D. Maquin, “State estimation of nonlinear

- systems using multiple model approach,” in *American Control Conference, ACC*, St. Louis, Missouri, USA, 2009.
- [67] T. Guerra, A. Kruszewski, and M. Bernal, “Control law proposition for the stabilization of discrete takagi-sugeno models,” *IEEE Trans. Fuzzy Syst.*, vol. 17, pp. 724–731, 2009.
- [68] K. Tanaka, H. Ohtake, and H. Wang, “A descriptor system approach to fuzzy control system design via fuzzy lyapunov functions,” *IEEE Trans. Fuzzy Syst.*, vol. 15, pp. 333–341, 2007.
- [69] K. Tanaka and H. Wang, *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. John Wiley & Sons, Inc., 2001.
- [70] A. Nagy, G. Mourot, J. Ragot, G. Schutz, and S. Gille, “Model structure simplification of a biological reactor,” in *Proc. of the 15th IFAC Symposium on System Identification, SYSID*, vol. 1, St. Malo, France, 2009.
- [71] A. Nagy, G. Mourot, B. Marx, G. Schutz, and J. Ragot, “Systematic multi-modeling methodology applied to an activated sludge reactor model,” *Ind. Eng. Chem. Res.*, vol. 49, no. 6, pp. 2790–2799, 2010.
- [72] S. Bezzaoucha, B. Marx, D. Maquin, and J. Ragot, “State and parameter estimation for nonlinear systems: a takagi-sugeno approach,” in *American Control Conference, ACC*, Washington, 2013.
- [73] H. Ohtake, K. Tanaka, and H. Wang, “Fuzzy modeling via sector nonlin-

- erarity concept,” in *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, vol. 1, July 2001, pp. 127–132.
- [74] —, “Fuzzy modeling via sector nonlinearity concept,” *Integrated Computer-Aided Engineering*, vol. 10, no. 4, pp. 333–341, 2003.
- [75] J. Novak, P. Chalupa, and V. Bobal, “Local Model Networks For Modelling And Predictive Control Of Nonlinear Systems,” in *Proceedings 23rd European Conference on Modelling and Simulation*, 2009.
- [76] G. Gregorčič and G. Lightbody, “Local model network identification with Gaussian processes,” *IEEE Transactions on Neural Networks*, vol. 18, no. 9, pp. 1404–1423., 2007.
- [77] E. Wernholt and S. Moberg, “Nonlinear gray-box identification using local models applied to industrial robots,” *Automatica*, vol. 47, pp. 650–660, 2011.
- [78] X. Jin, B. Huang, and D. S. Shook, “Multiple model LPV approach to nonlinear process identification with EM algorithm,” *Journal of Process Control*, vol. 21, pp. 182–193, 2011.
- [79] Y. Zhu and Z. Xu, “A method of LPV model identification for control,” in *Proceedings of the 17th World Congress of the International Federation of Automatic Control*, 2008, pp. 5018–5023.
- [80] J. Huang, G. Ji, Y. Zhu, and P. van den Bosch, “Identification of multi-model LPV models with two scheduling variables,” *Journal of Process Control*, vol. 22, pp. 1198–1208, 2012.

- [81] X. Jin, S. Wang, B. Huang, and F. Forbes, “Multiple model based LPV soft sensor development with irregular/missing process output measurement,” *Control Engineering Practice*, vol. 20, pp. 165–172, 2012.
- [82] Y. Zhao, B. Huang, H. Su, and J. Chu, “Prediction error method for identification of LPV models,” *Journal of Process Control*, vol. 22, pp. 180–193, 2012.
- [83] L. Chen, A. Tulsyan, B. Huang, and F. Liu, “Multiple model approach to nonlinear system identification with an uncertain scheduling variable using {EM} algorithm,” *Journal of Process Control*, vol. 23, no. 10, pp. 1480 – 1496, 2013.
- [84] A. A. Khalate, X. Bombois, R. Toth, and R. Babuska, “Optimal experimental design for LPV identification using a local approach,” in *15th IFAC Symposium on System Identification*, Saint-Malo, France, 2009.
- [85] S. C. McLoone, S. McGinnity, and G. W. Irwin, “Comparison of two construction algorithms for local model networks,” *International Journal of Systems Science*, vol. 33, no. 13, p. 1059–1072, 2002.
- [86] O. Nelles, *Nonlinear system identification*. springer, 2001.
- [87] O. Banfer and O. Nelles, “Polynomial model tree (polymot) - a new training algorithm for local model networks with higher degree polynomials,” in *IEEE International Conference on Control and Automation Christchurch*, New Zealand,, December 9-11 2009.

- [88] S. Ahmadi and M. Karrari, "An iterative approach to determine the complexity of local models for robust identification of nonlinear systems," *International Journal of Control, Automation and Systems*, vol. 10, no. 1, pp. 1–10, 2012.
- [89] J. Rezaie, B. Moshiri, A. Rafati, and B. Araabi, "Modified lolimot algorithm for nonlinear centralized kalman filtering fusion," in *Information Fusion, 2007 10th International Conference on*, July 2007, pp. 1–8.
- [90] M. Nekoui and S. Sajadifar, "Nonlinear system identification using locally linear model tree and particle swarm optimization," in *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on*, Saint-Malo, France, 15-17 Dec. 2006, pp. 1563–1568.
- [91] R. Mehran, A. Fatehi, C. Lucas, and B. Araabi, "Particle swarm extension to lolimot," in *Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on*, vol. 2, Oct 2006, pp. 969–974.
- [92] A. Sarabi Jamab and B. Araabi, "A learning algorithm for local linear neuro-fuzzy models with self-construction through merge & split," in *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, 7-9 June 2006, pp. 1–6.
- [93] A. Sarabi-Jamab and B. N. Araabi, "Pilimot: A modified combination of lolimot and pln learning algorithms for local linear neurofuzzy modeling," *Journal of Control Science and Engineering*, vol. 2011, 2011.
- [94] M. Mola, M. Khanesar, and M. Teshnehlab, "Subspace identification of dy-

- namical neurofuzzy system using lolimot,” in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, Oct 2010, pp. 366–372.
- [95] S. Jakubek and C. Hametner, “Identification of neurofuzzy models using gtl’s parameter estimation,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, pp. 1121–1133, 2009.
- [96] H. Hassani, M. Abdollahzadeh, H. Iranmanesh, and A. Miranian, “A self-similar local neuro-fuzzy model for short-term demand forecasting,” *Journal of Systems Science and Complexity*, vol. 27, no. 1, pp. 3–20, 2014.
- [97] S. Jakubek and C. Hametner, “A local neuro-fuzzy network for high-dimensional models and optimization,” *Engineering Applications of Artificial Intelligence*, vol. 19, pp. 705–717, 2006.
- [98] L. Breiman, “Hinging hyperplanes for regression, classification, and function approximation,” *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 999–1013, May 1993.
- [99] J. Xu, X. Huang, and S. Wang, “Adaptive hinging hyperplanes and its applications in dynamic system identification,” *Automatica*, vol. 45, pp. 2325–2332., 2009.
- [100] X. Huang, J. Xu, and S. Wang, “Operation optimization for centrifugal chiller plants using continuous piecewise linear programming,” in *Proc. IEEE International Conf. on Systems, Man, and Cybernetics*, Istanbul, Turkey, October 2010, pp. 1121–1126.

- [101] P. Pucar and M. Millnert, “Smooth hinging hyperplanes: A alternative to neural nets,,” in *Eur. Control Conf. (ECC)*,, Rome, Italy, 1995, pp. 1173–1178.
- [102] S. Ernst, “Hinging Hyperplane Trees for Approximation and Identification,” in *Proceedings of the 37th IEEE Conference on Decision & Control*, Tampa, Florida USA, December 1998.
- [103] O. Nelles, “Axes-oblique partitioning strategies for local model networks,” in *in Proc. Int. Symp. Intell. Control*, Munich, Germany, Oct. 2006, pp. 2378–2383.
- [104] B. Hartmann and O. Nelles, “On Smoothness in Local Model Networks,” in *2009, American Control Conference*, Hyatt Regency Riverfront, St. Louis, MO, USA, June 10-12 2009.
- [105] T. Fischer, B. Hartmann, and O. Nelles, “Increasing the Performance of a Training algorithm for Local Model Networks,” in *Proceedings of the World Congress on Engineering and Computer Science*, 2012.
- [106] B. Hartmann, O. Nelles, I. Skrjanc, and A. Sodja, “Supervised hierarchical clustering (suhiclust) for nonlinear system identification,” in *IEEE Symposium on Computational Intelligence in Control and Automation, CICA*, 2009.
- [107] T. Kenesei, B. Feil, and J. Abonyi, “Identification of dynamic systems by hinging hyperplane models,” in *Proceedings of the 7th International Confer-*

- ence on Applied Informatics*, vol. 2, Eger, Hungary, January 28–31 2007, p. 359–366.
- [108] J. K. Gugaliya, R. D. Gudi, and S. Lakshminarayanan, “Multimodel decomposition of nonlinear dynamics using fuzzy-cart approach,” *Journal of Process Control*, vol. 15, pp. 417–434, 2005.
- [109] N. Elfelly, J. Dieulot, M. Benrejeb, and P. Borne, “A new approach for multimodel identification of complex systems based on both neural and fuzzy clustering algorithms,” *Engineering Applications of Artificial Intelligence*, vol. 23, pp. 1064–1071, 2010.
- [110] N. Elfelly, J.-Y. Dieulot, and P. Borne, “A neural approach of multimodel representation of complex processes,” *Int. J. of Computers, Communications and Control*, vol. III, no. 2, pp. 149–160, 2008.
- [111] L. Teslic, B. Hartmann, O. Nelles, and I. Škrjanc, “Nonlinear system identification by gustafson-kessel fuzzy clustering and supervised local model network learning for the drug absorption spectra process,” *IEEE Transactions on Neural Networks*, vol. 22, no. 12, December 2011.
- [112] J. Abonyi, R. Babuska, and F. Szeifert, “Modified Gath-Geva fuzzy clustering for identification of takagi-sugeno fuzzy models,” *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 32, 2002.
- [113] R. Babuška and H. Verbruggen, “Constructing fuzzy models by prod-

- uct space clustering,” in *Fuzzy Model Identification: Selected Approaches*. Springer Berlin Heidelberg, 1997, pp. 53–90.
- [114] R. Babuška, L. Alic, M. Lourens, A. Verbraak, and J. Bogaard, “Estimation of respiratory parameters via fuzzy clustering,” *Artificial Intelligence in Medicine*, vol. 21, pp. 91–105, 2001.
- [115] J. Abonyi and B. Feil, *Cluster Analysis for Data Mining and System identification*. Birkhäuser Verlag AG, Basel · Boston · Berlin, 2007.
- [116] A. N. Venkat, P. Vijaysai, and R. D. Gudi, “Identification of complex nonlinear process based on fuzzy decomposition of the steady state space,” *Journal of Process Control*, vol. 13, pp. 473–488, 2003.
- [117] T. Samia, R. Ben Abdennour, A. Kamel, and P. Borne, “A Systematic Determination Approach of a Models’ Base For Uncertain Systems: Experimental Validation,” *IEEE International Conference on Systems Man and Cybernetics*, vol. 6, pp. 73–81, 2002.
- [118] T. Samia, A. Kamel, B. A. Ridha, and K. Mekki, “Multimodel Approach Using Neural Networks for Complex Systems Modeling and Identification,” *Nonlinear Dynamics and Systems Theory*, vol. 8, no. 3, pp. 299–316, 2008.
- [119] R. B. Mohamed, H. Ben Nasr, and F. MSahli, “A multi-model approach for a nonlinear system based on neural network validity,” *International Journal of Intelligent Computing and Cybernetics*, vol. 4, no. 3, pp. 331–352, 2011.

- [120] S. Bedoui, M. Ltaief, and K. Abderrahim, “New method for systematic determination of the model’s base of time varying delay system,” *International Journal of Computer Applications*, vol. 46, no. 1, 2012.
- [121] K. Demirli and P. Muthukumaran, “Higher order fuzzy system identification using subtractive clustering,” *J. of Intelligent and Fuzzy Systems*, vol. 9, pp. 129–158, 2000.
- [122] K. Demirli, S. X. Chengs, and P. Muthukumaran, “Subtractive clustering based on modeling of job sequencing with parametric search,” *Fuzzy Sets and Systems*, vol. 137, pp. 235–270, 2003.
- [123] J. K. Gugaliya and R. D. Gudi, “Multimodel decomposition of nonlinear dynamics using fuzzy classification and gap metric analysis,” in *proceedings of the 9th International Symposium on Dynamics and Control of Process systems*, 2010.
- [124] J. Novak, P. Chalupa, and V. Bobal, “Modelling and predictive control of nonlinear systems using local model networks,” in *18th IFAC Congress*, Milano (Italy), August 28- September 2, . 2011.
- [125] R. Murray-Smith and K. J. Hunt, “Local model architectures for nonlinear modelling and control,” in *Neural network engineering in dynamic control systems*. Berlin: Springer, 1997, pp. 61–82.
- [126] V. Verdult, L. Ljung, and M. Verhaegen, “Identification of composite local

- linear state-space models using a projected gradient search.” *International Journal of Control*, vol. 75, no. 16, p. 1385–1398, 2002.
- [127] R. Murray-Smith and T. A. Johansen, “Local learning in local model networks,” in *IEEE Conference on Artificial Neural Networks*, no. 409, June, 26-28 1995.
- [128] J. Yen, L. Wang, and C. W. Gillespie, “Improving the interpretability of ts-k fuzzy models by combining global learning and local learning,” *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 530–537, 1998.
- [129] T. A. Johansen and R. Babuska, “Multi-objective identification of takagi–sugeno fuzzy models,” *IEEE Trans. Fuzzy Syst.*, vol. 11, pp. 847–860, 2003.
- [130] X. Yang and H. Gao, “Multiple model approach to linear parameter varying time-delay system identification with {EM} algorithm,” *Journal of the Franklin Institute*, vol. 351, no. 12, pp. 5565 – 5581, 2014.
- [131] Y. Lu and B. Huang, “Robust multiple-model {LPV} approach to nonlinear process identification using mixture t distributions,” *Journal of Process Control*, vol. 24, no. 9, pp. 1472 – 1488, 2014.
- [132] R. Orjuela, D. Maquin, and J. Ragot, “Nonlinear system identification using uncoupled state multiple model approach,” in *Workshop on Advanced Control and Diagnosis, ACD’2006*, 2006.

- [133] S. C. McLoone and G. W. Irwin, "On nonlinear modelling using velocity based multiple model networks," in *Proceedings of the American Control Conference*, Arlington, VA, June 2001, pp. 25–27.
- [134] T. Johansen and B. Foss, "Constructing narmax models using armax models," *Int. J. Contr.*, vol. 58, pp. 1125–1153, 1993.
- [135] S. A. Billings and W. S. F. Voon, "Piecewise linear identification of nonlinear systems," *Int. J. Control*, vol. 46, pp. 215–235, 1987.
- [136] J. Z. Xu, J. Q. Zhao, and Y. Zhu, "Nonlinear MPC using identified LPV model," *Industrial & Engineering Chemistry Research*, vol. 6, pp. 3043–3051, 2009.
- [137] R. Toth, P. S. C. Heuberger, and P. M. J. V. den Hof, "A prediction-error identification framework for linear parameter-varying systems," in *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems – MTNS 2010*, Budapest, Hungary, 5-9 July 2010, pp. 1053–1059.
- [138] M. Kozek and S. Sinanović, "Identification of wiener models using optimal local linear models," *Simulation Modelling Practice and Theory*, vol. 16, pp. 1055–1066, 2008.
- [139] E. Arslan, M. C. Camurdan, A. Palazoglu, and Y. Arkun, "Multimodel scheduling control of nonlinear systems using gap metric," *Industrial & Engineering Chemistry Research*, vol. 43, no. 26, pp. 8275–8283, 2004.

- [140] E. Arslan, M. Camurdan, A. Palazoglu, and Y. Arkun, "Multi-model control of nonlinear systems using closed-loop gap metric," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 3, June 2004, pp. 2374–2378 vol.3.
- [141] D. Yubo, Y. Sai, and L. Bin, "Structural robustness and multi-model control in gap metric," *International Journal of Control and Automation*, vol. 6, no. 6, pp. 381–392, 2013.
- [142] B. Hartmann and O. Nelles, "Structure trade-off strategy for local model networks," in *IEEE International Conference on Control Application (CCA) Part of 2012 IEEE Multi-Conference on Systems and Control*, Dubrovnik, Croatia, October, 3-5 2012.
- [143] S. Mezghani, A. Elkamel, and P. Borne, "Multimodel control of discrete systems with uncertainties," *The Electronic International Journal Advanced Modeling and Optimization*, vol. 3, no. 2, pp. 7–17, 2001.
- [144] T. Samia, A. Kamel, B. A. Ridha, and K. Mekki, "A New Technique of validities' Computation for Multimodel Approach," *Wseas Transaction on circuit and systems*, vol. 2, no. 4, pp. 680–685, 2003.
- [145] B. Aufderheide and B. Bequette, "Extension of dynamic matrix control to multiple models," *Computers & Chemical Engineering*, vol. 27, no. 8-9, pp. 1079 – 1096, 2003, 2nd Pan American Workshop in Process Systems Engineering.

- [146] N. N. Nandola and S. Bhartiya, “A Multiple Model Approach for predictive Control of Nonlinear Hybrid Systems,” *Journal of Process Control*, vol. 18, no. 2, pp. 131–148, February 2008.
- [147] A. E. Kamel, M. Ksouri-Lahmari, and P. Borne, “Contribution to multi-model analysis and control,” *Studies in Informatics and Control, Vol. 9, pp.*, vol. 9, pp. 29–38, 2000.
- [148] R. Toscano and P. Lyonnet, “Robustness analysis and synthesis of a multi-pid controller based on an uncertain multimodel representation,” *Computers & Chemical Engineering*, vol. 31, no. 2, pp. 66 – 77, 2006.
- [149] R. Toscano, “Robust synthesis of a {PID} controller by uncertain multimodel approach,” *Information Sciences*, vol. 177, no. 6, pp. 1441–1451, 2007.
- [150] K. Srinivasan and T. V. Reddy, “Design of multi-model-based controller design and implementation using microcontroller for blood glucose regulation of type 1 diabetic system,” *International Journal of Biomedical Engineering and Technology*, vol. 5, pp. 343–359, 2011.
- [151] N. ElFelly, J.-Y. Dieulot, P. Borne, and M. Benrejeb, “A multimodel approach of complex systems identification and control using neural and fuzzy clustering algorithms,” in *Proceedings of the 2010 Ninth International Conference on Machine Learning and Applications*, ser. ICMLA '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 93–98.

- [152] A. Jamab and I. Mohammadzaman, “Predictive control of an electromagnetic suspension system via modified locally linear model tree with merging ability,” in *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, June 2006, pp. 1–6.
- [153] J. Novák, P. Chalupa, and V. Bobál, “Mimo model predictive control with local linear models,” in *Proceedings of the 13th WSEAS International Conference on Automatic Control, Modelling & Simulation*, ser. ACMOS’11. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2011, pp. 189–194.
- [154] Y. Li, J. Shen, K. Y. Lee, and X. Liu, “Offset-free fuzzy model predictive control of a boiler–turbine system based on genetic algorithm,” *Simulation Modelling Practice and Theory*, vol. 26, pp. 77–95, 2012.
- [155] T. V. Costa, A. M. Fileti, L. C. Oliveira-Lopes, and F. V. Silva, “Experimental assessment and design of multiple model predictive control based on local model networks for industrial processes,” *Evolving Systems*, pp. 1–11, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s12530-014-9113-1>
- [156] L. Serir, E. Ramasso, P. Nectoux, and N. Zerhouni, “Development of a novel soft sensor using a local model network with an adaptive subtractive clustering approach,” *Mechanical Systems and Signal Processing*, vol. 37, pp. 213–228, 2013.

- [157] T. Pan, D. Wong, and S. Jang, “Development of a novel soft sensor using a local model network with an adaptive subtractive clustering approach,” *Ind. Eng. Chem. Res.*, vol. 49, pp. 4738–4747, 2010.
- [158] M. Ksour-Lahmari, P. Borne, and M. Benrejeb, “Multimodel: the construction of Model Bases,,” *Studies in Informatics and Control*, vol. 13, no. 3, pp. 199–210, September 2004.
- [159] R. Shorten, R. Murry-Smith, R. Bjorgan, and H, “On the interpretation of local models in blended multiple structures,” *International Journal of Control*, vol. 72, no. 2, pp. 620–628, 1999.
- [160] B. Jarboui, M. Cheikh, P. Siarry, and A. Rebai, “Combinatorial particle swarm optimization (cpso) for partitional clustering problem,” *Applied Mathematics and Computation*, vol. 192, no. 2, pp. 337 – 345, 2007.
- [161] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.
- [162] T. Hachino, K. Shimoda, and H. Takata, “Identification of discrete-time hammerstein systems by using automatic choosing function model and particle swarm optimization,” in *Proceedings of the International Conference on Electrical Engineering*, 2008, pp. O–095–1–O–095–6.
- [163] H. Modares, A. Alfi, and M.-M. Fateh, “Parameter identification of chaotic

- dynamic systems through an improved particle swarm optimization,” *Expert Systems with Applications*, vol. 37, no. 5, pp. 3714 – 3720, 2010.
- [164] H. Modares, A. Alfi, and M.-B. N. Sistani, “Parameter estimation of bilinear systems based on an adaptive particle swarm optimization,” *Engineering Applications of Artificial Intelligence*, vol. 23, no. 7, pp. 1105 – 1111, 2010b.
- [165] B. Majhi and G. Panda, “Development of efficient identification scheme for nonlinear dynamic systems using swarm intelligence techniques,” *Expert Systems with Applications*, vol. 37, no. 1, pp. 556 – 566, 2010.
- [166] Y. Tang, L. Qiao, and X. Guan, “Identification of wiener model using step signals and particle swarm optimization,” *Expert Systems with Applications*, vol. 37, no. 4, pp. 3398 – 3404, 2010.
- [167] M. Cornoiu, C. Bara, and D. Popescu, “Metaheuristic approach in nonlinear systems identification,” *U.P.B. Sci. Bull., Series A*, vol. 75, no. 3, 2013.
- [168] J. Rissanen, “Stochastic complexity and modeling,” *Ann. Statist.*, vol. 14, no. 3, p. 1080–1100, 1986.
- [169] R. Stanforth, E. Kolossov, and B. Mirkin, “Hybrid k-means: Combining regression-wise and centroid-based criteria for qsar,” in *Selected Contributions in Data Analysis and Classification*, ser. Studies in Classification, Data Analysis, and Knowledge Organization, P. Brito, G. Cucumel, P. Bertrand, and F. Carvalho, Eds. Springer Berlin Heidelberg, 2007, pp. 225–233.

- [170] K. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *Neural Networks, IEEE Transactions on*, vol. 1, no. 1, pp. 4–27, Mar 1990.
- [171] F.-C. Chen and H. Khalil, "Adaptive control of a class of nonlinear discrete-time systems using neural networks," *Automatic Control, IEEE Transactions on*, vol. 40, no. 5, pp. 791–801, May 1995.
- [172] C. Wen, S. Wang, X. Jin, and X. Ma, "Identification of dynamic systems using piecewise-affine basis function models," *Automatica*, vol. 43, pp. 1824–1831, 2007.
- [173] J. Du, C. Song, and P. Li, "Application of gap metric to model bank determination in multilinear model approach," *Journal of Process Control*, vol. 19, no. 2, pp. 231 – 240, 2009.
- [174] S. Haykin, *Kalman Filters*. John Wiley & Sons, Inc., 2002, ch. Chapter 1, pp. 1–21.
- [175] W. Wen and H. Durrant-Whyte, "Model-based multi-sensor data fusion," in *IEEE Conference on Robotics and Automation*, Nice, France, 1992, p. 1720–1726.
- [176] J. Porrill, "Optimal combination and constraints for geometrical sensor data," *International Journal of Robotics Research 1988*, vol. 7, no. 6, pp. 66–77, 1988.

- [177] A. Alouani and W. Blair, “Use of a kinematic constraint in tracking constant speed, maneuvering targets,” *IEEE Trans. Autom. Control*, vol. 38, no. 7, pp. 1107–1111, 1993.
- [178] S. D. and T. L. Chia, “Kalman filtering with state equality constraints,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 128–136, 2002.
- [179] N. Gupta and R. Hauser, “Kalman filtering with equality and inequality state constraints,” Technical Report of Computing Laboratory, Numerical Analysis Group, University of Oxford, Tech. Rep., 2007.
- [180] B. Texeira, J. Chandrasekar, J. Palanthandalam-Madapusi, H., L. Torres, A. L., and D. Bernstein, “Gain-constrained kalman filtering for linear and nonlinear systems,” *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4113–4123, 2008.
- [181] S. Ko and R. Bitmead, R., “State estimation for linear systems with state equality constraints,” *Automatica*, vol. 43, no. 8, pp. 1363–1368, 2007.
- [182] R. Nikoukhah, S. Willsky A., and C. Levy, B., “Kalman filtering and riccati equations for descriptor systems.” *IEEE Transactions on Automatic Control*, vol. 37, no. 9, pp. 1325–1342, 1992.
- [183] N. Gupta, “Mathematically equivalent approaches for equality constrained kalman filtering,” Oxford University Numerical Analysis Group Technical Report, Tech. Rep., 2/10 2009. [Online]. Available: arXiv:0902.1565v1

- [184] S. D. and S. D. L., “Aircraft turbofan engine health estimation using constrained kalman filtering,” *ASME Journal of Engineering for Gas Turbines and Power*, vol. 127, pp. 323–328, 2005.
- [185] S. V., I. J., H. G., C. H., and R. J., “State estimation under nonlinear state inequality constraints. a tracking application,” in *16th Mediterranean Conf. on Control Automation*, Ajaccio, France, 2008, pp. 1669–1674.
- [186] S. N., S. Y., K. Y., and M. J., “Hand gesture estimation and model refinement using monocular camera. ambiguity limitation by inequality constraints,” in *IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, 1998, p. 268–273.
- [187] N. L. Nihan and G. A. Davis, “Recursive estimation of origin-destination matrices from input/output counts,” *Transportation Research Part B: Methodological*, vol. 21, no. 2, pp. 149 – 163, 1987.
- [188] D. Simon, “Kalman filtering with state constraints: a survey of linear and nonlinear algorithms,” *Control Theory Applications, IET*, vol. 4, no. 8, pp. 1303–1318, 2010.
- [189] D. M. Walker, “Parameter estimated using kalman filters with constraints,” *International Journal of Bifurcation and Chaos*, vol. 16, no. 04, pp. 1067–1078, 2006.
- [190] D. Simon, *Kalman filter generalizations*. John Wiley & Sons, Inc., 2006, pp. 183–227.

- [191] D. Alspach, “A parallel filtering algorithm for linear systems with unknown time varying noise statistics,” *Automatic Control, IEEE Transactions on*, vol. 19, no. 5, pp. 552–556, Oct 1974.
- [192] R. Rahbari, B. W. Leach, J. Dillon, and C. W. de Silva, “Adaptive tuning of a kalman filter using the fuzzy integral for an intelligent navigation system,” in *in Proceedings of IEEE International Intelligent Control Symposium*, 2002, p. 252–257.
- [193] R. Clements, P. Tavares, and P. Lima, “Small satellite attitude control based on a kalman filter,” in *Intelligent Control, 2000. Proceedings of the 2000 IEEE International Symposium on*, 2000, pp. 79–84.
- [194] O. V. Korniyenko, M. Sharawi, and D. N. Aloï, “Neural network based approach for tuning kalman filter,” in *IEEE International Conference on Electro Information Technology*, vol. 25, no. 5, May 2005, p. 1–5.
- [195] J. Macias and A. Gomez Exposito, “Self-tuning of kalman filters for harmonic computation,” *Power Delivery, IEEE Transactions on*, vol. 21, no. 1, pp. 501–503, Jan 2006.
- [196] B. J. Odelson, M. R. Rajamani, and J. B. Rawlings, “A new autocovariance least-squares method for estimating noise covariances,” *Automatica*, pp. 303–308, 2006.
- [197] B. M. Akesson, J. B. Jorgensen, N. K. Poulsen, and S. B. Jorgensen, “A

- generalized autocovariance least-squares method for kalman filter tuning,” *Journal of Process Control*, vol. 18, no. 7-8, pp. 769 – 779, 2008.
- [198] M. R. Rajamani and J. B. Rawlings, “Estimation of the disturbance structure from data using semidefinite programming and optimal weighting,” *Automatica*, pp. 142–148, 2009.
- [199] M. Saha, R. Ghosh, and B. Goswami, “Robustness and sensitivity metrics for tuning the extended kalman filter,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 63, no. 4, pp. 964–971, April 2014.
- [200] K. E. Rifai, O. E. Rifai, and K. Youcef-Toumi, “On robust adaptive switched control,” in *2005 American Control Conference*, June 8-10 2005, pp. 160–163.
- [201] Y. Xudong, “Nonlinear adaptive control by switching linear controllers,” *Systems & Control Letters*, vol. 61, no. 4, pp. 617 – 621, 2012.
- [202] Z. Zhengtao, Y. Lei, H. Jun, F. Shumin, and Y. Gang, “Multi-model switching control for siso discrete systems,” in *Control Conference (CCC), 2013 32nd Chinese*, July 2013, pp. 160–163.
- [203] L. Yu, S. Fei, and W. Qian, “Robust adaptive control for single input/single output discrete systems via multi-model switching,” in *proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 2013.

- [204] S. Kolyubin, D. Efimov, O. V. Nikiforov, and A. Bobtsov, “Control of non-linear systems using multiple model black-box identification,” in *9th IFAC Symposium on Nonlinear Control Systems*, vol. 9 part 1, 2013, pp. 582–587.
- [205] O. Galan, J. A. Romagnoli, and A. Palazoglu, “Real-time implementation of multi-linear model-based control strategies:an application to a bench-scale ph neutralization reactor,” *Journal of Process Control*, vol. 14, no. 5, pp. 571–579, 2004.
- [206] R. Gao, A. O. Dwyer, and E. Coyle, “Model predictive control of CSTR based on local model networks,” in *Proceedings of the Irish Signals and Systems Conference, University College Cork*, June 2002, pp. 397–402.
- [207] Z. K. Xue and S. Y. Li, “Multi-model modelling and predictive control based on local model networks,” *Control Intell. Syst.*, vol. 34, no. 2, pp. 105–112, May 2006.
- [208] D. Theilliol, H. Noura, and J.-C. Ponsart, “Fault diagnosis and accommodation of a three-tank system based on analytical redundancy,” *ISA Transactions*, vol. 41, no. 3, pp. 365 – 382, 2002.

Vitae

Name: Ahmed Adebawale Adeniran

Nationality: Nigerian

Date of Birth: 23/09/1974

Email: *adeniran_aa@yahoo.com*

Permenant Address: 2. Shofoluwe Lane Mushin, Lagos Nigeria

Education

BSc in Electrical/Electronics Eng. University of Ibadan, Ibadan, Nigeria
(2001)

MSc. in Computer Science (2006) University of Ibadan, Ibadan, Nigeria

MS in System Engineering (2009) King Fahd Univ. of Pet. & Minerals, Dhahran,
Saudi Arabia

PhD in System Engineering (2015) King Fahd Univ. of Pet. & Minerals, Dhahran,
Saudi Arabia

PUBLICATIONS

Journal

- Surajdeen A. Iliyas, M. E. Shafei and M. A. Habib, Ahmed A. Adeniran, " RBF neural network inferential sensor for process emission monitoring " , Control Engineering Practice 21 (2013) 962–970

- T.B. Asafa, Ahmed A. Adeniran and S.O.Olatunji, “ Functional networks models for rapid characterization of thin films deposited by chemical vapour deposition: an application to ultrathin poly-SiGe films” Applied Soft Computing, 28, 11-18, 2015.

Conference

- Ahmed A. Adeniran and Sami Elferik, “Validity Estimation For Multi-Model Identification Using Constrained Kalman Filter”, IASTED on Modeling Identification and Control (MIC) 2014.
- Ahmed A. Adeniran, Moustafa Elshafei and Gharib Hamada. “Functional networks softsensor for formation porosity and water saturation in oil Wells” Presented at IEEE International Instrumentation and Measurement Conference, I2MTC, Singapore, May, 2009.

Book:

- Ahmed Adeniran, Moustafa Elshafei and Gharib Hamada. “Artificial Intelligence Techniques In Reservoir Characterization: Functional networks softsensor for formation porosity and water saturation in oil Wells” VDM Verlag Dr. Paperback, June 2010, ISBN 978-3-639-24426-7.

Invited Presentations

- “Fault Diagnostics using Constrained Kalman Filter”, 7Th symposium on Maintenance, 2014, A symposium organized by center of excellence on maintenance, KFUPM

- “Constrained Kalman Filter as validity for multi-model identification and fault diagnosis of complex systems”, 8Th symposium on Industrial systems and control, 2014, A symposium organized by Department of Systems Engineering, KFUPM.

Workshop Attended

- Student Motivation: A workshop organized by Teaching and Learning Center, DAD, KFUPM March 12, 2012.
- Active Learning Strategies: A workshop organized by Teaching and Learning Center, DAD, KFUPM, August 20, 2014.
- Instructional Objectives: A workshop organized by Teaching and Learning Center, DAD, KFUPM, November 12, 2014.