UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

IDENTIFICATION AND ESTIMATION OF MULTI-MODAL COMPLEX DYNAMIC SYSTEM

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

YUZHEN XUE
Norman, Oklahoma
2009

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/319254

IDENTIFICATION AND ESTIMATION OF MULTI-MODAL COMPLEX DYNAMIC SYSTEM

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Dr. Thordur Runolfsson, Chair

Dr. Murad Özaydin

Dr. Joseph P. Havlicek

Dr. Choon Yik Tang

Dr. Tian-you Yu

© Copyright by YUZHEN XUE 2009
All Rights Reserved.

To My Family

Acknowledgements

I would like to acknowledge the people who have supported and guided me during the Ph.D. studies. First, I want to express my most profound gratitude to my advisor, Professor Thordur Runolfsson, for his continued mentorship. Professor Runolfsson has always been a source of inspiration and always offers instant help. His enthusiasm for the control theory area has also influenced and encouraged me. I am truly and always grateful for being offered this opportunity that has allowed me to grow both academically and professionally under his guidance. I would like to extend my thanks to the committee members: Professors Thordur Runolfsson, Murad Özaydin, Joe Havlicek, Tian-you Yu and Choon Yik Tang. My committee's feedback has served to strengthen my research. I would also like to thank Professor Victor DeBrunner and Professor David Baldwin as well as other members of the DySSC center of the University of Oklahoma for their support. Last but certainly not least, I wish to thank my family in China for being always very close to me emotionally as well as to thank my friends here in the U.S. who encouraged me all the time.

Contents

Acknowledgements iv

List of Tables vii

List of Figures viii

Abstract ix

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Related Work . . . 4
  - 1.2.1 Identification . . . 4
  - 1.2.2 Estimation . . . 5
- 1.3 Problem Description and Approach . . . 7
  - 1.3.1 Identification . . . 7
  - 1.3.2 Estimation . . . 12
- 1.4 Outline . . . 13

2 Preliminaries 15
- 2.1 Concepts for Markov Process . . . 15
- 2.2 Process with Small Noise . . . 15
- 2.3 Finite Dimensional Approximations of Markov Processes . . . 21
- 2.4 Metastability and Multi-modal Behavior . . . 23
- 2.5 Hidden Markov Models . . . 24
- 2.6 Non-negative Matrix Factorization (NMF) . . . 25
- 2.7 Kernel Principal Component Analysis . . . 27
  - 2.7.1 KPCA . . . 28
- 2.8 IMM Estimation Structure . . . 29
- 2.9 Particle Filtering . . . 30

3 Identification of Multi-Modal Complex Dynamic System 32
- 3.1 Identification of Multi-Modal Behavior . . . 32
  - 3.1.1 Estimation of State Process from Output Process . . . 34
  - 3.1.2 Transition Matrix Calculation . . . 50
  - 3.1.3 Identification of Multi-modal Behavior Based on State Process . . . 50
  - 3.1.4 Example . . . 54
  - 3.1.5 Conclusion . . . 57
- 3.2 Identification of Local Dynamics . . . 58

3.2.1 Nonlinear Identification Techniques . . . 58
3.2.2 Linear Identification Techniques . . . 67
3.2.3 Example . . . 67
3.2.4 Conclusion . . . 71
3.3 Conclusion . . . 71

4 Estimation of Hybrid Systems **72**
4.1 Algorithm Development . . . 72
4.1.1 IMMPF . . . 72
4.1.2 OTPF . . . 74
4.1.3 RMMPF . . . 75
4.2 Performance Evaluation . . . 79
4.2.1 Problem description . . . 79
4.2.2 Comparison with KMPF/OTPF . . . 80
4.2.3 Comparison with IMMPF . . . 82
4.2.4 Sensitivity to θ . . . 84
4.2.5 Number of particles . . . 85
4.3 Application: Maneuvering Target Tracking . . . 87
4.4 Conclusion . . . 91

5 Conclusion and Future Work **92**
5.1 Conclusion . . . 92
5.2 Future Work . . . 94

Bibliography **95**

List of Tables

Table 1 MAE for KMPF and RMMPF. 80

Table 2 Average Times IMMPF has been used 80

Table 3 MAE for RMMF and IMMPF 83

Table 4 Average Times IMMPF has been used. 83

Table 5 Calculation Time for RMMPF and IMMP 84

Table 6 Comparison of IMMPF and RMMPF 90

List of Figures

Figure 1 Proposed System Identification Approach 33

Figure 2 Output Trajectories 57

Figure 3 Comparison of Original and Simulated System 68

Figure 4 Evaluation of the Proposed Local Dynamics Identification Algorithm – Phase Plane 70

Figure 5 Evaluation of the Proposed Local Dynamics Identification Algorithm – Time Series 70

Figure 6 State Estimation by RMMPF 81

Figure 7 Mode Estimation by RMMPF 81

Figure 8 Performance versus Threshold 85

Figure 9 Performance Comparison 86

Figure 10 Maneuvering Target Trajectory and Tracking Results 89

Abstract

In this dissertation we study identification of complex dynamic systems as well as hybrid system estimation. For the identification part, we propose a scheme to identify an autonomous complex stochastic dynamic system based on a black-box model, that is, the system is modeled based on output data only. The system under study is a system whose underlying space is the union of strong attraction domains. The system exhibits a behavior such that it spends a long time in one strong attraction domain before transitioning to another one. Systems showing this behavior can be found in many applications ranging from biology to power systems to chemical processes. Considering the nature of this type of a system, we model it as a hybrid system. In particular, it is a strong attraction domain featured hybrid system (SAFHS). Two principal features of this type of a hybrid system are that the boundaries between the modes (strong attraction domains) are nonlinear and the dynamic behavior within each mode can be highly nonlinear, e.g. limit cycle. Identification algorithms for this kind of hybrid system are not well developed. In this dissertation we propose our first result for identification of this type of system. The resulting model is hybrid in nature. We detect the multi-modal dynamics as well as local dynamics within each mode, thus providing a complete unified approach of identification of the system dynamics. The approach developed in this dissertation is based on finite dimensional approximations of compact operators, spectral theory for non-reversible Markov chains, identification techniques for hidden Markov models (HMM), and identification techniques for linear and non-linear dynamics. Examples are carried out to verify our analysis and to illustrate the effectiveness of the proposed algorithms. In the estimation part, we present a high accuracy, low computational load method for a nonlinear/non-Gaussian hybrid system, motivated by the need to get a better trade-off between efficiency and accuracy which is a crucial issue

in real time estimation problems. The efficiency and accuracy of the proposed algorithm are illustrated by examples. Moreover, its good performance makes it practical and robust for tracking a target in a complex situation, as we demonstrate by a simulated maneuvering target tracking example.

CHAPTER 1

Introduction

1.1 Background

System identification and state estimation are important branches in modern control theory. In order to develop efficient and accurate control strategies, it is necessary to know "what the system is" and "what the system is doing", i.e. to identify the system and to estimate the state of the system.

Techniques that construct models from observed data are known as system identification in the control area. System identification bridges real applications and abstract mathematical models.

According to the degree of availability of the prior knowledge, system identification falls into three types of models [36]:

- White box models: In this case the model structure is usually completely derived from first principles, i.e., physical, chemical, biological, economical, etc. laws, while the parameters are either known or estimated from data.

- Black box models: No or very little prior knowledge is available. Both model structure and parameters are determined from experimental and mathematical modeling.

- Gray box models: These represent a compromise or combination between white and black box models. Typically, the determination of as much as possible the model structure relies on prior knowledge while the model parameters are mainly determined by measurement data.

As for real applications, most system identification algorithms are of the black box or dark gray box type. There are a lot of well developed linear black-box models, while identification of nonlinear black-box models is a challenge because nonlinear processes can have very many distinct dynamic properties. It is difficult to work out simple, accurate, and general models for nonlinear systems, especially in black-box cases.

Large scale real engineered systems that are subject to performance and operational constraints often exhibit complex behavior that was not anticipated at design time. This type of systems can be found in numerous applications [48], ranging from power [2], [60] and communication networks to biological and chemical processes [12], [21], where the system dynamic model is nonlinear and needs to be identified on the basis of experimental data since it is hard to develop models from first principles. System identification for these systems is thus a nonlinear black-box problem. Moreover, it is frequently too complicated for conventional identification techniques to work out a global model that covers many cases or branches for these systems. We denote these systems as complex dynamic systems. An alternative modeling way for complex dynamic system is to utilize the system structure or attributes for partitioning it into simpler components where conventional techniques are easier to be employed and are accurate within each component. In this context, the description of complicated dynamic systems is equivalent to a hybrid system model. Hybrid systems are heterogeneous dynamic systems whose behavior is determined by interacting continuous and discrete dynamics [37]. Moreover, it has to be stressed here that in real applications, the system usually suffers from both process noise and observation noise, which makes the system a stochastic system. Note that in many cases, the process and the observation noise are originated from system parameter variations as well as sensor noise. Frequently noise processes of this type are small and in such instances the stochastic system can be treated as a deterministic system disturbed by small noise. Note here by small noise we mean the relative value of the noise to the typical values of the system state and output is small.

In this dissertation we are going to identify complex stochastic dynamic system based on a black-box model, that is, the system is modeled based only on output data. Thus the problem considered here is a data driven identification approach.

As mentioned above, when a system is too complex to develop a global model, an alternative choice is to develop local models and treat the global model as a composition of the local models. That is, to model the complex system in hybrid system framework. In order to develop a hybrid system model that represents the original system of interest, it is necessary to specify the discrete states (modes), the number of the discrete states, the modal dynamics and the continuous dynamics (the local dynamics) of the hybrid model.

The theory and identification approaches for stochastic complex dynamic system in hybrid system modeling framework are not fully developed, but there is a lot of current research, some of which will be described in Section 1.2.1.

Given "what the system is" and the observations, it is also of interest to find out “what the system is doing”. This is an estimation problem or, equivalently, a filtering problem.

Bayesian estimation is a powerful approach for state estimation given stochastic system state-space models. The central idea of Bayesian estimation is to construct the posterior probability density function (pdf) of the state conditioned on all the available measurements. In principle, an optimal (with respect to any criterion) estimate of the state may be obtained from the pdf. [3]

In real applications, it is expected for an efficient method to be capable of on-line, real-time estimation of the stochastic dynamic systems. In this case a recursive Bayesian filtering is a convenient solution. A recursive filtering approach means that the measurements can be processed sequentially rather than as a batch so that it is not necessary to store the complete data set nor to reprocess existing data if a new measurement becomes available.[3]

There exist many well developed recursive Bayesian filters for general (non hybrid) system state estimation, e.g. Kalman filter for linear stochastic systems with Gaussian

process and measurement noises, extended Kalman filter, multigrid based filter, particle filter, etc. for nonlinear systems with Gaussian or non-Gaussian noises. Moreover, the existing recursive Bayesian filters have been successfully extended to hybrid system cases.

For the estimation problem in this dissertation, we concentrate on the hybrid system estimation for the following reasons: first, the hybrid system framework matches the model we will develop for stochastic complex dynamic system through our identification scheme; second, hybrid models have been considered as an effective way of modeling a large class of systems that exhibit complex system behavior [47], e.g. systems in fields of signal processing including seismic signal processing [34], digital communications [30] and target tracking [32], where state estimation and mode detection are of considerable interests.

Some of the current hybrid system estimation algorithms based on recursive Bayesian filters will be described in Section 1.2.2.

1.2 Related Work

1.2.1 Identification

In the last decade hybrid systems have been developed actively for the description of many complex dynamic systems. In hybrid model description for complex dynamic system, a considerable effort has been devoted to develop piecewise affine (PWA) models, see e.g. [56], [57], [4], [19]. It has been shown that PWA models are equivalent to several classes of linear hybrid system formulations.

Several different data driven techniques developed for PWA model identification were compared in [25] and a good overview was presented. One of the existing representative PWA identification approaches is based on *algebraic techniques* and *generalized principal component analysis* (GPCA) [56], [57]. In this approach the identification problem of the multiple models is formulated as an identification of a single "lifted" model in a bigger space and then the parameters of the submodels identified from the lifted model. In this ap-

proach the order of the models is not required apriori and the number of submodels is estimated as well. Another approach is based on a *Bayesian procedure*, where prior knowledge about the modes and parameters of the model are assumed and then the data is classified through a data classification procedure with the maximal probability [24]. The procedure is carried out by sequentially processing the data and iteratively updating the parameter vectors. The regions are subsequently estimated using a modification of a standard multicategory robust linear programming procedure (MRLP) [24]. In the Bayesian procedure the model order and number of submodels are fixed. The third procedure, so-called *bounded-error procedure*, is based on set membership identification ideas where model parameters are selected based on a bounded error criteria between the data and model output [4]. Another class of methods are so-called *clustering procedures* that utilize the assumption that the PWA system is locally linear (i.e. in each mode) and consequently data points belonging to the same submodel (and region) are spatially close. The method consists of constructing "feature vectors" from the data and partitioning the feature vectors into q groups by applying a so-called K-means clustering algorithm. The clustering procedure assumes that the model order and the number of submodels are fixed.

In general, the discrete state in PWA models is determined by a polyhedral partition of the state-input domain, where on each partition a linear affine system model is considered.

Finally, the identification of hybrid systems with nonlinear domain boundaries and nonlinear local dynamics has only been studied in a limited manner in the literature, e.g. [17].

1.2.2 Estimation

In the estimation of hybrid systems a multiple model (MM) approach is generally employed. It is assumed that the system obeys one of a finite number of models at each time, and switches between models are based on a finite state Markov chain. Usually filters are matched to different models and the state estimation is then a weighted sum of the estimate from each filter. The method is decision free since the weights are determined as the

probability of each model being correct given the current measurement.

A variety of algorithms has been proposed in the literature for the solution of MM problems, e.g. the Interacting multiple model (IMM) [7] and the generalized pseudo-Bayes (GPB) algorithms [11]. Both the IMM and GPB algorithms are based on deterministic finite Gaussian mixture approximations and have the assumption that the hybrid system under study is Jump Markov Linear System. Basically, these two algorithms can both be treated as the variants of a Kalman filter in multiple modal application.

Since in reality, the system models always have nonlinear characteristics, extended Kalman filters (EKF) are usually used as model matching filters. However, because of its first-order linearization of the nonlinear models, EKF can introduce large estimated errors. Another possible strategy is to compute a fixed grid approximation to the filtered state density. This involves approximating the continuous-valued process by a finite state process with fixed states. The filtered state density and filtered state estimates can then be computed easily at these grid points according to Bayes' rule. Unfortunately, such a fixed grid approximation suffers from the curse of dimensionality [3].

Recently particle filtering (PF) has been recognized as a superior alternative to traditional estimation methods as it is suitable for both linear/non-linear and Gaussian/non-Gaussian systems. Particle filter is a Monte Carlo based method that deduces the representation of the state distribution iteratively using large number of weighted samples (particles). Estimators based on particle filters have been successfully applied to stochastic hybrid systems [33] [9] [14].

In [33], a standard bootstrap PF has been extended to hybrid system estimation. A major drawback of standard bootstrap PF in hybrid system estimation is that the number of particles in a specific mode is proportional to the mode probability and, consequently, if the mode probability is very low, only few particles can reside in that mode. To alleviate this problem, several PF algorithms with fixed number of particles in each mode, e.g. the interacting multiple model particle filtering (IMMPF), the observation and transition-based

most likely modes tracking particle filtering (OTPF), etc. have been proposed [10], [5], [49]. These methods have been shown to improve the estimation accuracy to a considerable extent. In [49], the algorithm OTPF is proposed and compared with KMPF, which is a standard particle filter that has complete knowledge of the real mode at all times (KMPF can be considered the benchmark that all hybrid system particle algorithms should be compared to). It turns out that for some problems OTPF performs not much worse than KMPF but in other problems the performance is considerably worse. Another algorithm that has been proposed is the so-called IMMPF [10], [5]. This algorithm has shown very promising performance for a large class of problems but at the expense of heavy computational cost. It can be seen that compared to IMMPF algorithms the merit of OTPF is its lower computational load but at the expense of accuracy since OTPF is biased and sensitive to observation outliers.

As mentioned above, particle filter is a sample (particle) based filtering algorithm. At each time step, all the particles take part in the calculation, which introduces relatively high computational burden. Therefore, when a particle filter is applied to a hybrid system, the trade off between estimation accuracy and calculation efficiency is of major concern.

1.3 Problem Description and Approach

1.3.1 Identification

In our research we propose a hybrid model description for a complex dynamic system based on the system's inherent structure or attributes. The resulting hybrid model has discrete states (modes) represented by nonlinear partition of the underlying space of the system as well as nonlinear dynamics within each mode. The approach presented here is completely different from the ones presented in [17] and provides further insight into the intricacies of identification of complex dynamic systems.

When analyzing large scale complex dynamic system, which is complex enough so that

it contains many attractors, we are frequently interested in partitioning the underlying space into a union of strong attraction domains. In particular, we are interested in partitioning the space into regions where the system exhibits a behavior such that it spends a long time in one region before transitioning to another one. We call such regions strong attraction domains. The dynamics within each strong attraction domain can be quite complex and highly nonlinear and, in fact, can contain more than one dynamic attractor of the system. There are two principal reasons for analyzing the complex system in terms of strong attraction domains. The first is for the identification and estimation benefit. In particular, realizing that complex dynamic systems are frequently too complicated for conventional estimation and identification techniques we seek alternative methods that utilizes the system structure for partitioning it into simpler components where conventional techniques are easier to be employed. The second is for the control aim. Combining the knowledge of system behavior corresponding to the strong attraction domains, one can design a control law for the system and make it switch from the attraction domains that represent suboptimal or even failed operations to the ones that represent desired or optimal behavior. We remark that if we associate with each strong attraction domain a discrete valued modal variable and consider the corresponding dynamics within each domain, the resulting model is hybrid in nature. In order to differentiate the hybrid system described above from general hybrid systems we call this specific hybrid system a Strong Attraction domain Featured Hybrid System, denoted as SAFHS. In the above description the complex dynamic system of interest is a SAFHS with two features: first, the system switches between unknown nonlinear dynamics and second, the boundaries between the strong attraction domains are nonlinear.

Our research aims at detecting modal clusters, i.e. the strong attraction domains (modes) of the SAFHS, and the modal dynamic behavior of SAFHS. Then the data can be classified into modal clusters where the local dynamics within each mode can be identified based on the classified data series. As a result this allows for identification of nonlinear local dynamic behavior such as limit cycles. Moreover, since there is no assumption for

the structure of the underlying space partition, the partitioned regions can have nonlinear boundaries.

The methodology to be developed in this dissertation applies to discrete time system whose state behavior is governed by a Markov process and whose output is a noise corrupted function of the state. In particular, we consider an autonomous discrete time dynamic system given by a state space model of the form

$$\begin{aligned} x_{k+1} &= f(x_k, v_k) \\ y_k &= g(x_k) + w_k \end{aligned} \tag{1.1}$$

where $x_k \in \mathbb{R}^{n_x}$ is the state, $y_k \in \mathbb{R}^{n_y}$ is the output, v_k and w_k are process and output noises. We assume that the process noise is i.i.d. (independent and identically distributed). In this case x_k is a Markov process and y_k can be treated as a hidden Markov process. For $x \in \mathbb{R}^{n_x}$ and a Borel set $A \in \mathfrak{B}(\mathbb{R}^{n_x})$ we let $p(x,A) = \Pr(x_{k+1} \in A \,|x_k = x)$ be the transition function for the process. Moreover, we denote by $\mathscr{M}(\mathbb{R}^{n_x})$ the space of all probability measures on $\mathbb{R}^{n_x}$.

Example 1 *A simple example of a dynamical system that has strong attraction domains is the case when system (1.1) is deterministic system perturbed by additive noise which has q asymptotically stable equilibrium points with domains of attraction $A_1, \ldots, A_q$ that form a partition of the underlying space. If the process noise v_t is assumed to be small (with variance of order ε^2) it can be shown using large deviations theory that the transition probabilities between domains of attraction are of the order $O(e^{-\frac{1}{\varepsilon^2}})$ (see [18]).*

As we mentioned earlier, we are going to model a complex system as SAFHS and develop an identification procedure based on output data. The basic identification problem is to identify the regions where the dynamics are concentrated, the transition dynamics between the regions, and the dynamics within each region. We note that for a given division of the state space into regions there is a corresponding division of the output space. As a

result, when we discuss identification of multi-modal behavior in this dissertation we are either discussing identification in the output or state space and the space of interest should be clear from the context at each time.

To make the objectives more precise assume that the state space can be treated as the union of strong attraction domains $A_1, \ldots, A_q$ of $\mathbb{R}^{n_x}$ for which the system dynamics are concentrated on and switch between. Given a sequence of output data $y_1, y_2, \ldots$ the problem we consider is the identification of the following quantities.

- Identification of the number q of state partition components and the partition $A_1, \ldots, A_q$ of the state space.
- Identification of the dynamics that govern the transition dynamics between the partition components.
- Identification of the dynamic laws that govern the system dynamics within each partition component.

The approach developed in this dissertation is based on finite dimensional approximations of compact operators, spectral theory for non-reversible Markov chains, identification techniques for hidden Markov models (HMM), and identification techniques for linear and non-linear dynamics. It has the following main steps.

1. For the identification of the number of partition components as well as the partition itself we utilize spectral theory for non-reversible Markov chains and Hidden Markov Models (HMM). In particular, we discretize the state and output spaces and approximate the underlying Markov process and corresponding output process by finite state processes. Then utilizing HMM techniques we identify the state and transition laws for the Markov chain. Finally, spectral theory for non-reversible Markov chains is used to identify the number of partition components as well as the partition itself, and the transition law between those components. We remark that the coarseness of

the discretization of the underlying state and output spaces affects the accuracy of the description of the (boundary of) the partition components but no restriction is put on the "shape" of the partition.

2. For the identification of the local dynamics within each partition component we utilize conventional system identification ideas. In particular, once we have identified the partition $A_1, \ldots, A_q$ of the state space and the association between state and output sequences we identify a corresponding partition of the output space. We then map the output sequence onto the partition components of the output space. Based on a preliminary analysis of the output sequences within each output partition component we choose a identification procedure for the dynamics for that component. We emphasize that for the identification of the local dynamics we use the original data (i.e. *not* discretized). Furthermore, based on the local behavior, the type of model used within each partition component can different.

Once we complete the above identification procedure we obtained a finite collection of local dynamic models as well as a Markov transition law that governs the switching among the local models. The resulting model is therefore of hybrid nature, i.e. associated with each local model is a modal variable taking value in a finite set, the dynamics of the modal variable is governed by a finite state Markov process. Furthermore, each local model is associated with a particular partition component of the output space.

The identification procedure described above is developed for a class of systems described by system equation (1.1). Furthermore, we remark that the procedure assumes or requires the following.

- The stochastic complex dynamic system of interest is subjected to small process and observation noise so that the stochastic system can be treated as a deterministic system disturbed by small noise. Note, as before, here by small noise we mean the relative value of the noise to the typical values of the system state and output is small.

- The system dynamic model is autonomous. Therefore, external measurable inputs are not included or have been incorporated through a closed loop control policy.

- The state space can be partitioned into a finite number of strong attraction domains, i.e. sets that the system spends a long time in before transitioning to another such domain. This requires in particular that the local dynamics are stable in an appropriate sense.

1.3.2 Estimation

In hybrid system estimation, the system is usually described by a stochastic state-space model,

$$\begin{aligned} x_k &= f_{r_k}(x_{k-1}, v_k) \\ y_k &= g_{r_k}(x_k, w_k) \end{aligned} \tag{1.2}$$

where r_k is a discrete-time M-state Markov chain with transition probabilities $\pi_{s,t} \triangleq \Pr\{r_k = t \mid r_{k-1} = s\}$ for any $s, t \in \mathbf{M}$, where $\mathbf{M} \triangleq \{1, 2, \cdots, M\}$, $x_k \in \mathbb{R}^{n_x}$ is the system state and $y_k \in \mathbb{R}^{n_y}$ is observation at time k. The variables $v_k \in \mathbb{R}^{n_v}$ and $w_k \in \mathbb{R}^{n_w}$ are process and measurement noise vectors at time k and are assumed to be of known statistics, which can be Gaussian or non-Gaussian. The functions f_{r_k} and g_{r_k} are the state function and observation function respectively in mode r_k at time k. It is assumed that the initial distribution of state x_0 is known.

The estimation problem we consider is to detect the mode r_k and estimate the state x_k based on the measurements y_k and the system model formulated above.

As described in Section 1.2.2, particle filtering is a superior candidate for linear/nonlinear and Gaussian/non-Gaussian system estimation and has been successfully applied to stochastic hybrid systems. However, it suffers from the relatively high computational burden. In particular, when particle filtering is applied to a hybrid system, the trade off

between estimation accuracy and calculation efficiency is of principal concern.

In this dissertation we are going to propose a new high accuracy and reduced computational load algorithm for stochastic hybrid system state estimation and mode detection. The proposed algorithm combines two existing particle filtering based algorithms: OTPF and IMMPF. The basic idea of the new algorithm comes from observing the estimation performance of the IMMPF algorithm. It can be seen that in the IMMPF algorithm most of the time the particles in the real mode (i.e. correct mode) dominate the estimation, i.e. the particles of the real mode have dominant weights compared to the particles of other modes. Furthermore, we observe that in this situation (i.e. when the real mode dominates the estimation) the OTPF algorithm is an accurate algorithm for estimating the mode and state. On the other hand, when mode switching takes place or when there are observation outliers, the dominance of one mode is not obvious. In this case, it is not suitable to just retain particles in the mode corresponding to the largest weight and evolve based on them like what is done in OTPF. In fact in this case the IMMPF is a better choice to get high accuracy estimation. Inspired by this, we present here an algorithm that combines OTPF and IMMPF by introducing a threshold value to select which estimation algorithm should be used at each time step. This new technique aims at improving the computation efficiency without losing much estimation accuracy.

1.4 Outline

The dissertation is organized as follows. In Chapter 2, some of the basic mathematical preliminaries and concepts, including concepts relevant to Markov processes, attributes of process with small noise, finite dimensional approximations of Markov processes, metastability and multi-modal behavior, hidden Markov models, non-negative matrix factorization, kernel principal component analysis (KPCA) as well as IMM estimation structure and particle filtering, are introduced. In Chapter 3, the identification of complex dynamic system in SAFHS framework is proposed. In particular, the partitioning of underlying space

of the complex dynamic system will be specified. That partition corresponds to the discrete state (mode) of SAFHS. Moreover, the modal transition probability and the local dynamics will be identified. Chapter 3 is the main contribution in this dissertation. Chapter 4 tackles the hybrid system estimation problem. A new particle filtering based algorithm is proposed to get a better trade off between calculation efficiency and estimation accuracy. In Chapter 5 conclusions and future work are discussed.

CHAPTER 2

Preliminaries

In this Chapter, we give an overview of some of the mathematical techniques that will be employed in the solution of the complex dynamic system identification problem. A brief introduction to IMM estimation structure and particle filtering is also included, upon which the main results of system estimation rely.

2.1 Concepts for Markov Process

We begin by introducing a couple concepts of Markov Chains that are used frequently in the remainder of the dissertation.

Definition 1 *A Markov chain on* $\mathbf{N} = \{1, \cdots, N\}$ *is called reversible if it's transition matrix* $P \in \mathbb{R}^{N*N}$ *satisfies*

$$\pi_i p_{ij} = \pi_j p_{ji},\ 1 \leq i, j \leq N$$

A Markov chain that is not reversible is said to be non-reversible.

Definition 2 *A Markov chain on* $\mathbf{N}$ *is called ergodic if it is irreducible, positive recurrent and aperiodic. Equivalently, it can also be defined as a process whose statistical properties (such as its mean and variance) can be deduced from a single, sufficiently long sample (realization) of the process.*

2.2 Process with Small Noise

Here we discuss properties of a stochastic system with small noise. Consider the stochastic system (1.1), and suppose the process noise and observation noise are small. We consider

the special case of (1.1):

$$\begin{aligned} x_{k+1}^{\varepsilon} &= b\left(x_k^{\varepsilon}\right) + \varepsilon\sigma\left(x_k^{\varepsilon}\right) v_k,\ x_0^{\varepsilon} = x \\ y_k^{\varepsilon} &= g\left(x_k^{\varepsilon}\right) + \varepsilon w_k \end{aligned} \tag{2.1}$$

Assume that v_k is a sequence of i.i.d. random variables with distribution with density $q(v)$ with mean zero and finite second moments and w_k is a standard white noise process (mean zero and unit variance) independent of v_k. The processes x_k^{ε} and y_k^{ε} can be viewed as being the results of a small additive random perturbation of the deterministic system

$$\begin{aligned} x_{k+1}^{d} &= b\left(x_k^{d}\right),\ x_0^{d} = x \\ y_k^{d} &= g\left(x_k^{d}\right) \end{aligned} \tag{2.2}$$

In the stochastic system (2.1) we assume for simplicity that $Ev_k v_k^T = I$. Furthermore we assume that $b : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ and $\sigma : \mathbb{R}^{n_x \times n_r} \to \mathbb{R}^{n_x}$ satisfy a Lipschitz condition and grow no faster than linearly, i.e. there exists $C > 0$ so that

$$\begin{aligned} \|b(x) - b(y)\| &\leq C\|x-y\|,\ \sqrt{\textstyle\sum_{i,j}\left(\sigma_{ij}(x) - \sigma_{ij}(y)\right)^2} \leq C\|x-y\| \\ \|b(x)\|^2 &\leq C^2\left(1 + \|x\|^2\right),\ \textstyle\sum_{i,j}\left(\sigma_{ij}(x)\right)^2 \leq C^2\left(1 + \|x\|^2\right) \end{aligned} \tag{2.3}$$

Moreover, assume $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ is Lipschitz, i.e. for some $C > 0$,

$$\|g(x) - g(y)\| \leq C\|x-y\| \tag{2.4}$$

For a sequence r_k let $r_{i:j} = \left(r_i, r_{i+1}, \ldots, r_j\right)$ and for two sequences r_k and s_k define the distance between the strings $r_{i:j}$ and $s_{i:j}$ as

$$\operatorname{dist}\left(r_{i:j}, s_{i:j}\right) = \sqrt{\textstyle\sum_{l=i}^{j} \|r_l - s_l\|_2^2}$$

where $\|\cdot\|_2$ is the standard Euclidean norm.

The following theorem is a discrete time adoption of results from [18].

Theorem 1 *For any $L^{'} > 0$ and $\delta > 0$ we have*

$$E\left\|x_k^{\varepsilon} - x_k^d\right\|^2 \leq \varepsilon^2 a^{\varepsilon}(k)$$

for a monotonically increasing function $a^{\varepsilon}(k)$ that depends on x and C. Furthermore,

$$\lim_{\varepsilon \to 0} \Pr\left(\sup_{0 \leq k \leq L'} \left\|x_k^{\varepsilon} - x_k^d\right\| > \delta\right) = 0$$

Proof. We note that since v_k is a sequence of i.i.d. random variables it follows from (2.1) that x_k^{ε} and v_k are independent. Now,

$$\begin{aligned} E\left\|x_{k+1}^{\varepsilon} - x_{k+1}^d\right\|^2 &= E\left\|b(x_k^{\varepsilon}) - b\left(x_k^d\right) + \varepsilon\sigma(x_k^{\varepsilon})v_k\right\|^2 \\ &= E\left\|b(x_k^{\varepsilon}) - b\left(x_k^d\right)\right\|^2 + 2\varepsilon E\left\langle b(x_k^{\varepsilon}) - b\left(x_k^d\right), \sigma(x_k^{\varepsilon})v_k\right\rangle \\ &\quad + \varepsilon^2 E\left\|\sigma(x_k^{\varepsilon})v_k\right\|^2 \end{aligned} \tag{2.5}$$

For the second term on the right hand side of (2.5) we have using the independence of x_k^{ε} and v_k

$$\begin{aligned} E\left\langle b(x_k^{\varepsilon}) - b\left(x_k^d\right), \sigma(x_k^{\varepsilon})v_k\right\rangle &= E\left\langle b(x_k^{\varepsilon}), \sigma(x_k^{\varepsilon})v_k\right\rangle - E\left\langle b\left(x_k^d\right), \sigma(x_k^{\varepsilon})v_k\right\rangle \\ &= E\sum_i b_i(x_k^{\varepsilon})\sum_j \sigma_{ij}(x_k^{\varepsilon})v_k^j - E\sum_i b_i\left(x_k^d\right)\sum_j \sigma_{ij}(x_k^{\varepsilon})v_k^j \\ &= \sum_i\sum_j Eb_i(x_k^{\varepsilon})\sigma_{ij}(x_k^{\varepsilon})Ev_k^j - \sum_i\sum_j Eb_i\left(x_k^d\right)\sigma_{ij}(x_k^{\varepsilon})Ev_k^j \\ &= 0 \end{aligned}$$

Thus we have (again using the independence of x_k^ε and v_k)

$$\begin{aligned}
E\left\|x_{k+1}^\varepsilon - x_{k+1}^d\right\|^2 &= E\left\|b\left(x_k^\varepsilon\right) - b\left(x_k^d\right)\right\|^2 + \varepsilon^2 E\left\|\sigma\left(x_k^\varepsilon\right) v_k\right\|^2 \\
&= E\left\|b\left(x_k^\varepsilon\right) - b\left(x_k^d\right)\right\|^2 + \varepsilon^2 E\left\langle\sigma\left(x_k^\varepsilon\right) v_k, \sigma\left(x_k^\varepsilon\right) v_k\right\rangle \\
&= E\left\|b\left(x_k^\varepsilon\right) - b\left(x_k^d\right)\right\|^2 + \varepsilon^2 \textstyle\sum_i \sum_j \sum_l E\sigma_{ij}\left(x_k^\varepsilon\right)\sigma_{il}\left(x_k^\varepsilon\right) E v_k^j v_k^l \\
&= E\left\|b\left(x_k^\varepsilon\right) - b\left(x_k^d\right)\right\|^2 + \varepsilon^2 \textstyle\sum_{i,j} E\left(\sigma_{ij}\left(x_k^\varepsilon\right)\right)^2 \\
&\leq C^2 E\left\|x_k^\varepsilon - x_k^d\right\|^2 + \varepsilon^2 C^2\left(1 + E\left\|x_k^\varepsilon\right\|^2\right)
\end{aligned}$$

Then by Lemma 2 below we have (assuming for simplicity that $C \neq 1$) and noting that $E\left\|x_0^\varepsilon - x_0^d\right\|^2 = 0$

$$E\left\|x_k^\varepsilon - x_k^d\right\|^2 \leq \varepsilon^2 C^2\left(1 + E\left\|x_k^\varepsilon\right\|^2\right)\frac{1 - C^{2k}}{1 - C^2} \tag{2.6}$$

Thus if we show that $1 + E\left\|x_k^\varepsilon\right\|^2$ is bounded above we have the first statement of the Theorem. For this note that (using a similar argument as before)

$$\begin{aligned}
1 + E\left\|x_{k+1}^\varepsilon\right\|^2 &= 1 + E\left\|b\left(x_k^\varepsilon\right) + \varepsilon\sigma\left(x_k^\varepsilon\right) v_k\right\|^2 \\
&= 1 + E\left\|b\left(x_k^\varepsilon\right)\right\|^2 + 2\varepsilon E\left\langle b\left(x_k^\varepsilon\right), \sigma\left(x_k^\varepsilon\right) v_k\right\rangle + \varepsilon^2 E\left\|\sigma\left(x_k^\varepsilon\right) v_k\right\|^2 \\
&= 1 + E\left\|b\left(x_k^\varepsilon\right)\right\|^2 + \varepsilon^2 E\left\|\sigma\left(x_k^\varepsilon\right) v_k\right\|^2 \\
&= 1 + E\left\|b\left(x_k^\varepsilon\right)\right\|^2 + \varepsilon^2 \textstyle\sum_{i,j} E\left(\sigma_{ij}\left(x_k^\varepsilon\right)\right)^2 \\
&\leq 1 + C^2\left(1 + E\left\|x_k^\varepsilon\right\|^2\right) + \varepsilon^2 C^2\left(1 + E\left\|x_k^\varepsilon\right\|^2\right) \\
&= 1 + C^2\left(1 + \varepsilon^2\right)\left(1 + E\left\|x_k^\varepsilon\right\|^2\right)
\end{aligned}$$

Thus, by Lemma 2

$$1 + E\left\|x_k^\varepsilon\right\|^2 \leq \left(1 + \|x\|^2\right)\left(C^2\left(1 + \varepsilon^2\right)\right)^k + \frac{1 - \left(C^2\left(1 + \varepsilon^2\right)\right)^k}{1 - C^2\left(1 + \varepsilon^2\right)} = \tilde{a}^\varepsilon(k)$$

Therefore, (2.6) becomes

$$E\left\|x_k^\varepsilon - x_k^d\right\|^2 \leq \varepsilon^2 C^2 \left(1 + E\left\|x_k^\varepsilon\right\|^2\right) \frac{1-C^{2k}}{1-C^2} \leq \varepsilon^2 C^2 \frac{1-C^{2k}}{1-C^2} \tilde{a}(k) = \varepsilon^2 a^\varepsilon(k) \qquad (2.7)$$

Turning now to the second statement of the Theorem. Define

$$z_{i+1}^\varepsilon = \begin{cases} \left\|x_{i+1}^\varepsilon - x_{i+1}^d\right\|, & \text{if } \max\limits_{1\leq j\leq i} \left\|x_j^\varepsilon - x_j^d\right\| < \delta \\ z_i^\varepsilon, & \text{otherwise} \end{cases}$$

Then

$$\Pr\left(\max_{1\leq k\leq L'} \left\|x_k^\varepsilon - x_k^d\right\| > \delta\right) = \Pr\left(z_{L'}^\varepsilon > \delta\right) \leq \frac{1}{\delta^2} E\left[\left(z_L^\varepsilon\right)^2\right]$$

by Chebyshev's inequality. By definition of z_i^ε, if $\max\limits_{1\leq j\leq L'-1} \left\|x_j^\varepsilon - x_j^d\right\| < \delta$ then $z_{L'}^\varepsilon = \left\|x_{L'}^\varepsilon - x_{L'}^d\right\|$ and else $z_{L'}^\varepsilon = z_{L'-1}^\varepsilon$. Similarly, if $\max\limits_{1\leq j\leq L'-1} \left\|x_j^\varepsilon - x_j^d\right\| > \delta$ and $\max\limits_{1\leq j\leq L'-2} \left\|x_j^\varepsilon - x_j^d\right\| < \delta$ then $z_{L'}^\varepsilon = z_{L'-1}^\varepsilon = \left\|x_{L'-1}^\varepsilon - x_{L'-1}^d\right\|$. Continue in this manner until we reach a $\bar{k}$ so that $\max\limits_{1\leq j\leq \bar{k}-1} \left\|x_j^\varepsilon - x_j^d\right\| < \delta$ and thus $z_{\bar{k}}^\varepsilon = \left\|x_{\bar{k}}^\varepsilon - x_{\bar{k}}^d\right\|$. Furthermore, we note that $z_{L'}^\varepsilon = z_{L'-1}^\varepsilon = \ldots = z_{\bar{k}}^\varepsilon = \left\|x_{\bar{k}}^\varepsilon - x_{\bar{k}}^d\right\|$ and thus

$$\begin{aligned} \Pr\left(\max_{1\leq k\leq L'} \left\|x_k^\varepsilon - x_k^d\right\| > \delta\right) &= P\left(z_{L'}^\varepsilon > \delta\right) \leq \frac{1}{\delta^2} E\left[\left(z_{L'}^\varepsilon\right)^2\right] = \frac{1}{\delta^2} E\left\|x_{\bar{k}}^\varepsilon - x_{\bar{k}}^d\right\|^2 \\ &\leq \frac{1}{\delta^2} \varepsilon^2 a^\varepsilon\left(\bar{k}\right) \end{aligned}$$

■

Lemma 2 *Consider a sequence f_k that satisfies*

$$f_{k+1} \leq a f_k + b$$

for some positive constants a and b. Then

$$f_k \leq \begin{cases} f_0 + b\frac{1-a^n}{1-a} & a \neq 1 \\ f_0 + nb & a = 1 \end{cases}$$

Proof. Trivial by iteration. ■

Let x_k^ε and x_k^d be the solutions of (2.1) and (2.2) and let $x_{0:L'}^\varepsilon = (x_0^\varepsilon, \ldots, x_{L'}^\varepsilon)$ and similarly for $x_{0:L'}^d$. Define as before

$$\operatorname{dist}\left(x_{0:L'}^\varepsilon, x_{0:L'}^d\right) = \sqrt{\sum_{i=0}^{L'} \left\|x_k^\varepsilon - x_k^d\right\|^2}$$

The following follows immediately from Theorem 1.

Proposition 3 *For any $L' > 0$ and $\delta > 0$*

$$\lim_{\varepsilon \to 0} \Pr\left(\operatorname{dist}\left(x_{0:L'}^\varepsilon, x_{0:L'}^d\right) > \delta\right) = 0$$

Proof. Define $e_k = \left\|x_k^\varepsilon - x_k^d\right\|, k = 0, \ldots, L'$ and view $e = \left[e_1, \ldots, e_{L'}\right]^T$ as a random vector in $\mathbb{R}^{L'}$. Then by equivalence of norms on $\mathbb{R}^{L'}$ we have

$$\|e\|_\infty \leq \|e\|_2 \leq \sqrt{L'}\,\|e\|_\infty$$

Therefore, if $\|e\|_2 > \delta$ then $\|e\|_\infty > \frac{\delta}{\sqrt{L'}}$ and thus

$$\Pr\left(\|e\|_2 > \delta\right) \leq \Pr\left(\|e\|_\infty > \frac{\delta}{\sqrt{L'}}\right)$$

Noting that $\|e\|_2 = \operatorname{dist}\left(x_{0:L'}^\varepsilon, x_{0:L'}^d\right)$ and $\|e\|_\infty = \max_{0 \leq k \leq L'} \left\|x_k^\varepsilon - x_k^d\right\|$ completes the proof. ■

Now define $\operatorname{dist}\left(y_{1:L'}^\varepsilon, y_{1:L'}^d\right)$ in a similar way as $\operatorname{dist}\left(x_{0:L'}^\varepsilon, x_{0:L'}^d\right)$.

Proposition 4 *For any* $L^{'} > 0$ *and* $\delta > 0$,

$$\lim_{\varepsilon \to 0} \Pr\left(\text{dist}\left(y^{\varepsilon}_{1:L'}, y^{d}_{1:L'}\right) > \delta\right) = 0$$

Proof. It follows using a similar argument as in the proof of Theorem 1 that for any $k \in \left[0, \ldots, L^{'}\right]$,

$$\begin{aligned} & E\left\|y^{\varepsilon}_{k} - y^{d}_{k}\right\|^2 = E\left\|g\left(x^{\varepsilon}_{k}\right) - g\left(x^{d}_{k}\right) + \varepsilon w_k\right\|^2 = E\left\|g\left(x^{\varepsilon}_{k}\right) - g\left(x^{d}_{k}\right)\right\|^2 + \varepsilon^2 E\left\|w_k\right\|^2 \\ \leq \;& CE\left\|x^{\varepsilon}_{k} - x^{d}_{k}\right\|^2 + \varepsilon^2 E\left\|w_k\right\|^2 \leq Ca\left(\varepsilon\right) + \varepsilon^2 E\left\|w_k\right\|^2 = A\left(\varepsilon\right) \end{aligned}$$

Also,

$$\Pr\left(\left\|y^{\varepsilon}_{k} - y^{d}_{k}\right\|^2 > \delta\right) \leq \frac{1}{\delta^2} E\left\|y^{\varepsilon}_{k} - y^{d}_{k}\right\|^2 \leq \frac{1}{\delta^2} A\left(\varepsilon\right)$$

Thus, $\Pr\left(\left\|y^{\varepsilon}_{k} - y^{d}_{k}\right\|^2 > \delta\right) \to 0$ as $\varepsilon \to 0$. The rest of the proof now follows from a similar argument as before. ■

Note that the above proofs for a discrete time stochastic system can be extended easily to a discrete time discrete state stochastic system case.

2.3 Finite Dimensional Approximations of Markov Processes

In this section we present a finite dimensional approximate model for a class of general Markov processes such as the one in (1.1).

When the process noise sequence in the first equation in (1.1) is a sequence of i.i.d. random variables the state process x_k is a Markov process. Let $p\left(x, A\right), x \in \mathbb{R}^{n_x}, A \in \mathbf{B}(\mathbb{R}^{n_x})$ be the transition function for the process. Then if the initial state x_0 has distribution $\nu \in \mathscr{M}(\mathbb{R}^{n_x})$, where $\mathscr{M}(\mathbb{R}^{n_x})$ denotes the set of probability measures on $\mathbb{R}^{n_x}$, the distribution of x_1 is

$$\mathscr{P}\nu\left(A\right) = \int p\left(x, A\right) \nu\left(dx\right)$$

where $\mathscr{P}$ is the so-called Perron-Frobenius operator. By iteration the distribution of x_k is $\mathscr{P}^k \nu(A)$. Furthermore, if the process is ergodic it is well known that $\mathscr{P}^k \nu(A) \to \mu(A)$ as $k \to \infty$ where $\mu \in \mathscr{M}(\mathbb{R}^{n_x})$ is the invariant measure for the process. The invariant measure can be shown to be a fixed point of the operator $\mathscr{P}$.

Let m be the Lebesgue measure on $\mathbb{R}^{n_x}$ and assume that $p(x,\cdot)$ has a transition density $t(x,y)$ with respect to m. Obviously $t(x,\cdot) \in L_1(m)$ and $t(x,y) \geq 0$. Note that we denote by $L_K(m)$ the L^K on R^{n_x}. In this case we can consider $\mathscr{P}$ as an operator on $L_1(m)$,

$$\mathscr{P}g(y) = \int_{\mathbb{R}^{n_x}} t(x,y)\, g(x)\, m(dx) \text{ for all } g \in L_1(m)$$

Obviously, if g is the density for ν then $\mathscr{P}g$ is the density for $\mathscr{P}\nu$. If the transition density $t(x,y)$ satisfies

$$\int\int t(x,y)\, m(dx)\, m(dy) < \infty$$

then $\mathscr{P} : L_2(m) \to L_2(m)$ can be shown to be compact. In this case if V_l, $l \geq 1$ is a sequence of subspaces of L_2 and $Q_l : L_2 \to V_l$ is a sequence of projections such that Q_l converge to the identity on L_2 as $l \to \infty$, then the approximate operator $P_l = Q_l \mathscr{P}$ has the property $\|\mathscr{P} - P_l\| \to 0$ as $l \to \infty$. If we let $B_1, \ldots, B_l$ be a partition of $\mathbb{R}^{n_x}$ and let $V_l = span\{\varphi_1, \ldots, \varphi_l\}$ where φ_i are the characteristic functions of B_i and we choose Q_l to be the Galerkin projection of L_2 onto V_l then the approximate P_l has a (stochastic) matrix representation with entries $\bar{p}_{ij} = \left\langle \mathscr{P}\varphi_j, \varphi_i \right\rangle$ where $\langle \cdot, \cdot \rangle$ is the inner product on L_2.

In the following analysis it is assumed that the approximate finite dimensional Markov Chain for the dynamic system is P_l. Indeed, if a finite dimensional transition matrix P is constructed from a data sequence using a counting process as that will be described later and the process satisfies some appropriate technical conditions then for sufficiently large data records the data constructed transition matrix converges to the matrix P_l.

2.4 Metastability and Multi-modal Behavior

In this section, we introduce the notion of metastability and its relationship to multi-modal behavior.

Let A and B be measurable sets on $\mathbb{R}^{n_x}$, assume that the distribution of the initial state is the invariant measure μ and define the transition probability from B into A as

$$p(A|B) = \Pr(\xi_1 \in A | \xi_0 \in B) = \frac{\Pr(\xi_1 \in A, \xi_0 \in B)}{\Pr(\xi_0 \in B)} = \frac{\int_B p(\xi, A)\, \mu(d\xi)}{\mu(B)}$$

if $\mu(B) > 0$, and $p(A|B) = 0$ otherwise. Note that $p(A|B)$ characterizes the dynamic fluctuations of the distribution of the Markov chain within the invariant distribution μ. The following definition is from [22].

Definition 3 *A Borel set A is said to be invariant if* $p(A|A) = 1$ *and metastable if* $p(A|A) \approx 1$. *Therefore, a metastable set is almost invariant.*

If a set A is invariant then a trajectory that starts in A stays in there forever. On the other hand if the initial state belongs to a metastable set then the system state will stay there for a long time but will eventually exit the set with positive probability, which matches the multi-modal behavior of the system of interest in this dissertation where the system is represented by SAFHS. Consider all possible partitions $A_1, \ldots, A_q$ of the state space $\mathbb{R}^n$, with $\cup_{i=1}^{q} A_i = \mathbb{R}^n$ and $A_i \cap A_j = 0$ unless $i = j$. Finding a partition such that $\sum_{i=1}^{q} p(A_i|A_i) \approx q$, i.e. a metastable partition, is of interest. Note that the partition sets obtained in this manner are in fact the partition regions of interest of the underlying space of the SAFHS system under consideration. For such a partition, if it exists, the transitions between the metastable components can be approximated by an appropriately defined q dimensional Markov chain. This Markov chain describes the system's modal dynamics.

We remark that the partition region estimation of the underlying space of the system of interest is closely related to the characterization of a metastable partition in the same space.

The characterization of a metastable partition is in general a difficult problem. This problem has been considered in detail in [22], [42] for a reversible Markov chain. As most real engineered systems are represented by a non-reversible Markov process, in our research an approach for identifying a metastable partition for the general non-reversible case for finite dimensional Markov chains [38] will be utilized.

With the concept of metastability, part of the objectives of the proposed identification approach for SAFHS can be restated as identifying from noisy output data the metastable partition of the state space and the dynamics of the metastable components.

2.5 Hidden Markov Models

When only output data is available from measurements, system (1.1) can be viewed as a HMM process. Next we briefly introduce the concept of HMM.

HMM is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters and this Markov process is observed by outcomes generated according to the associated state-output probability distribution. A finite dimensional HMM is characterized by the following: N, the number of states in the model; M, the discrete alphabet size of observations; $P = \{p_{ij}\}$, the state transition probability distribution; $B = \{b_j(l)\}$, the state to observation probability distribution; $\pi^0 = \{\pi_i^0\}$, the initial state distribution [31]. By denoting the individual states and the individual observation symbols as $X = \{x^1, \ldots, x^N\}$ and $Y = \{y^1, \ldots, y^M\}$, express $p_{ij} = P(\bar{x}_{k+1} = x^j \mid \bar{x}_k = x^i)$, $b_j(l) = P\{\bar{y}_k = y^l \mid \bar{x}_k = x^j\}$ and $\pi_i^0 = P\{\bar{x}_0 = x^i\}$ where $\bar{x}_k$ and $\bar{y}_k$ are the state and output of the Markov chain at time k, respectively, and $i, j \in \{1, \ldots N\}$, $l \in \{1, \ldots M\}$. If the underlying state process is ergodic then there exists an unique stationary distribution $\pi = \{\pi_i\}$, $i \in \{1, \ldots N\}$ such that

$$\lim_{n \to \infty} p_{ij}^n = \pi_j$$

where p_{ij}^n is the (i, j) entry of P^n.

The basic identification problem for the HMM is the following: given an output observation series, estimate the corresponding state sequence and estimate model parameters $\lambda = (P, B, \pi)$. Since the complex dynamic system (1.1) can be approximated by a HMM, solutions to above problems for the HMM are suitable for characterizing the dynamic system. Estimation of the state sequence from the observation sequence is one of the principal tasks in HMM modelling. In [16] and [53] several algorithms for this purpose are discussed. The newly developed algorithm in [53] has been shown to have better performance than the classic Baum-Welch algorithm. Finally, for the complex hybrid system the identification of one additional parameter, i.e. the transition probability between modes, is also treated as a modelling problem in order to describe the system behavior between modes as well as within each mode. In fact, the identification of the modal transition matrix depends on the properties of the transition matrix P.

2.6 Non-negative Matrix Factorization (NMF)

NMF is a recently proposed method for generating a low rank approximation for a matrix V with non-negative entries (in the remainder of the dissertation we refer to such matrix as being non-negative), see e.g. [28], [29]. In particular, given a non-negative matrix $V \in \mathbb{R}^{n \times m}$, NMF finds an approximate factorization $V \approx WH$ into smaller size non-negative matrix factors $W \in \mathbb{R}^{n \times r}$ and $H \in \mathbb{R}^{r \times m}$, where $r \leq \min(n, m)$. NMF can capture the intrinsic structure underlying the object being described by the matrix V and has been successfully applied to a variety of data sets, e.g. in image processing, document classification, acoustics, and so on. In [53], NMF has been utilized for state estimation for a hidden Markov process.

Given a non-negative matrix $V \in \mathbb{R}^{n \times m}$, there are no closed form solutions for finding non-negative factors W and H which satisfy $V = WH$. Lee, et al. proposed an iterative numerical algorithm to minimize the error between V and the calculated WH. The iterative algorithm for NMF developed in [28] is as follows,

- Initialize W and H to random non-negative matrices
- *repeat until convergence*
  - $H_{i,l} \leftarrow H_{i,l} \frac{\sum_s W_{s,i} \frac{V_{s,l}}{(WH)_{s,l}}}{\sum_s W_{s,i}}$
  - $W_{k,i} \leftarrow W_{k,i} \frac{\sum_j H_{i,j} \frac{V_{k,j}}{(WH)_{k,j}}}{\sum_j H_{i,j}}$

We remark that it is shown in [28] that the W and H matrices calculated in this way minimize the Kullback-Leibler divergence, which is a matrix distance measure, between V and WH.

When NMF is utilized in seeking only a W, H pair which satisfies $WH = V$, then the uniqueness of the solution of the NMF is not necessarily important. However, when W and H have physical meanings and are calculated by applying NMF to the corresponding matrix V, then the existence of unique solution to NMF is of primary importance. Unfortunately, getting a unique solution is not guaranteed in the NMF algorithm.

Denote real true model matrices as W^{tr} and H^{tr} which satisfy $V = W^{tr}H^{tr}$. We note that for any invertible matrix T we have $V = W^{tr}H^{tr} = W^{tr}TT^{-1}H^{tr}$ and therefore, if we define $W = W^{tr}T$ and $H = T^{-1}H^{tr}$ we see that there exist a large number of matrices W and H, that lead to a positive factorization of V, i.e. any T that results in positive factors will suffice. Recall that $W, W^{tr} \in \mathbb{R}^{n\times r}$ and $H, H^{tr} \in \mathbb{R}^{r\times m}$, where $r \leq \min(n,m)$. It is easy to show that in the $r = 2$ case NMF always results in W and H matrices such that $V = WH$ and $W = W^{tr}T$, $H = T^{-1}H^{tr}$, where T is a permutation matrix. That is, when $r = 2$, NMF has a unique solution up to a permutation matrix. However, such uniqueness of NMF is not guaranteed for $r > 2$.

In many applications the data matrix V is generated as the product of real true model matrices W^{tr} and H^{tr}. In such cases when NMF is utilized to recover the (true) system matrices the issue of the uniqueness becomes of paramount importance. The uniqueness of NMF solution has been discussed in few papers for specific situations, see [50], [13], [27].

2.7 Kernel Principal Component Analysis

In this section we introduce so-called kernel principal component analysis (KPCA). For completeness, we state background on principal component analysis (PCA) and kernel methods first.

PCA is a classical method that is commonly used to find relationship between points in a data set with high dimensionality and in turn to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set [23]. In particular, PCA provides a sequence of best linear approximations to a given high-dimensional observation. The basic procedure for PCA is to execute an eigenvalue decomposition of the covariance matrix of the data, the highest eigenvalues correspond to the eigenvectors that represent the principal components. Once the principal components (major contributing components) have been found, the data can be compressed from the higher dimension space into a lower dimension space characterized by the principal components without much loss of information. PCA is a popular model reduction or data compression algorithm. However, its effectiveness is limited by its global linearity. Good tutorials on PCA can be found in [45], [44].

Kernel methods [55], [41] map an input vector $x_i \in \mathbb{R}^L$ to a high dimensional feature space of vectors $\Phi(x_i)$, where $\Phi : \mathbb{R}^L \to \mathbb{R}^H$, $L \ll H$, through a nonlinear mapping. When a kernel function $k(u,v)$ satisfies the so-called Mercer's condition [55], the inner product in the high dimensional feature space can be calculated using a positive definite kernel function without making direct reference to the vectors $\Phi(x_i)$, i.e. using the relationship $k(x_i,x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$. This is known as the "kernel trick". The most commonly used kernels include the Gaussian kernel where $k(x_i,x_j) = \exp(-\frac{\|x_i - x_j\|}{2\sigma^2})$ and the polynomial kernel where $k(x_i,x_j) = \left(1 + x_i^T x_j\right)^p$. The Gaussian kernel gives a feature space of infinite dimension, while the polynomial kernel gives a feature space of finite dimension.

The idea of kernel methods can help solving the nonlinear problem in the input space

through just performing conventional linear algorithms in the high-dimensional feature space, e.g. the kernel recursive least square (KRLS) algorithm [51], [15]. Kernel principal component analysis (KPCA) [39], [40] is also such a algorithm. As a nonlinear method, KPCA is nothing but the PCA in the feature space associated with a kernel function [61]. KPCA finds principal components that are nonlinearly related to the input space by performing PCA in the high dimensional feature space, where the discovery of low-dimensional latent structure is expected to be easier.

2.7.1 KPCA

Here we give the brief procedure of the KPCA, the reader is referred to [39], [40] for detailed information.

- Select N data points $x_1 \cdots x_N$ in a L dimensional space as a training set.
- Choose an appropriate kernel function $k(.,.)$ and calculate $k(x_i,x_j)$, $i,j \in \{1,\cdots,N\}$
- Center in order to remove the mean from data $\Phi(x)$ in feature space, i.e.

$$\tilde{k}(x_i,x_j) = k(x_i,x_j) - E_{x_i}[k(x_i,x_j)] - E_{x_j}[k(x_i,x_j)] + E_{x_i}[E_{x_j}[k(x_i,x_j)]]$$

- Let $K \triangleq \{\tilde{k}(x_i,x_j)\}$ and note that $K = K^T$. Let K have eigenvalues λ^n and eigenvectors α^n. (We note that due to symmetry the eigenvalues and eigenvectors are real.) Then, for samples in the feature space, the most representative m dimensional features are the following vectors:

$$\frac{\sum_{j=1}^{N} \alpha_j^1 k(x_j,x)}{\sqrt{\lambda^1}}, \frac{\sum_{j=1}^{N} \alpha_j^2 k(x_j,x)}{\sqrt{\lambda^2}}, \cdots, \frac{\sum_{j=1}^{N} \alpha_j^m k(x_j,x)}{\sqrt{\lambda^m}} \tag{2.8}$$

where $\alpha_j^1, \cdots \alpha_j^m$ are the m eigenvectors associated with the first m largest eigenvalues $\lambda^1, \cdots, \lambda^m$ of K. Based on the essence of the PCA methodology, the feature

extraction procedure based on (2.8) produces the minimum reconstruction error.

2.8 IMM Estimation Structure

IMM estimation was first proposed by Blom in [8] and has triggered a large variety of related approaches to hybrid system state estimation and mode detection. The IMM estimator requires the same number of filters as the number of system modes. With the filters operating in parallel at all times, all modes are processed in parallel and the mode probability is used to detect the correct mode. Moreover, a combination (weighted sum) of the state estimates of all filters yields the state estimate [32]. The IMM estimator has shown excellent estimation performance but is of relatively high computational cost. Here we give an introduction to the IMM estimation structure.

Consider a filter circle from one measurement update up to and including the next measurement update. In the IMM estimation scheme, this circle is decomposed into a sequence of transitions [7]:

$$
\begin{aligned}
p_{r_{k-1}|y_{k-1}} &\rightarrow p_{r_k|y_{k-1}} \\
p_{x_{k-1}|r_{k-1},y_{k-1}} &\rightarrow p_{x_{k-1}|r_k,y_{k-1}} \\
p_{x_{k-1}|r_k,y_{k-1}} &\rightarrow p_{x_k|r_k,y_{k-1}} \\
p_{r_k|y_{k-1}} &\rightarrow p_{r_k|y_k} \\
p_{x_k|r_k,y_{k-1}} &\rightarrow p_{x_k|r_k,y_k}
\end{aligned}
$$

This can be generalized into three steps [43]: 1) mixing/interaction of the mode-conditioned estimates at the beginning of the estimation cycle; 2) mode-conditioned state estimation (prediction and update of the state), done independently for each mode by appropriate (mode-matched) filter modules; 3) mode probability update and estimation, done using the outputs of all the mode-conditioned filters.

The IMM estimator proposed by Blom and Shalom originally made use of Gaussian mixtures as an approximate information state. Since then several algorithms that fall under this estimation structure have been presented. Of special interest are recent algorithms that combine IMM and particle filtering, readers can resort to [5] [10] [6] for more information.

2.9 Particle Filtering

A Particle filter is a fairly recently developed estimation method which can be used in both linear and nonlinear system with Gaussian or non-Gaussian noise processes. A particle filter is a sequential Monte Carlo based method that allows for a complete representation of the state distribution using weighted samples (particles). The particles evolve randomly in time according to a simulation-based rule. The weights of the particles are updated according to Bayes' rule.

Next we will introduce the basic idea of particle filtering for the system model (1.1). The key idea of particle filtering is to represent the required posterior distribution $p(x_k|y_k,\ldots,y_1) = p(x_k|y_{1:k})$ by a set of random samples $\{x_k^i, i = 1,\cdots,N_s\}$ with associated weights ω_t^i:

$$p_{x_k|y_{1:k}} \cong \sum_{i=1}^{N_s} \omega_k^i \delta(x_k - x_k^i)$$

and to compute estimates based on these samples and weights [3]. It can be shown that the weights satisfy the iterative equation,

$$\omega_k^i = \omega_{k-1}^i \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, y_{1:k})}$$

where $q(\cdot)$ is called the importance density [3]. Samples x_k^i are generated from $q(\cdot)$.

There are several ways for choosing $q(\cdot)$. In the bootstrap particle filter [20] (also called condensation algorithm) $q(\cdot)$ is chosen to be $p(x_k|x_{k-1}^i)$. This seems to be the most common choice of importance density since it is intuitive and simple to implement. With

this choice of $q(\cdot)$, the weight evolution becomes

$$\omega_k^i = \omega_{k-1}^i p(y_k|x_k^i)$$

The above bootstrap algorithm can be extended to hybrid system estimation easily as in [33] where the details can be found.

CHAPTER 3

Identification of Multi-Modal Complex Dynamic System

In Section 1.3.1 we introduced the identification problem that we study in the dissertation and how we can treat the system of interest as a SAFHS system and identify it through the modeling of SAFHS. In this Chapter, we develop the scheme of the identification of SAFHS based on output data. The basic identification procedure includes two parts, first is to identify the multi-modal behavior of the SAFHS system, which includes identifying the regions where the dynamics are concentrated and the transition dynamics between the regions. The second part of the identification procedure is to detect the dynamics within each region, i.e. to detect the local dynamics. These two identification tasks are presented in detail below.

3.1 Identification of Multi-Modal Behavior

In this section we formulate the identification of the system's multi-modal behavior. We consider the discrete time dynamic system given by (1.1), where the state process x_k is a Markov process and y_k is the output of the corresponding hidden Markov process. Suppose that the output space is discretized resulting in the discretized output process $\bar{y}_k$ on $Y = \{y^1, \ldots, y^M\}$. Moreover, with discretization, the underlying Markov process x_k on $\mathbb{R}^{n_x}$ is represented by the finite dimensional Markov Chain $\bar{x}_k$ on X, where $X = \{x^1, \ldots, x^N\}$. In order to simplify notation, in the following we use $\{1, \ldots, M\}$ to denote the output space of the chain. The corresponding finite dimensional Markov chain (to be identified) has state space $X = \{x^1, \ldots, x^N\} \triangleq \{1, \ldots, N\}$. Note that process $\bar{y}_k$ is the output of a hidden Markov process since the discretized process $\bar{x}_k$ is still Markov.

Given the output sequence y_k, after discretization we have sequence $\bar{y}_k$, the objectives for identification of the multi-modal behavior are 1) calculate the transition matrix of the hidden Markov state $\bar{x}$; 2) characterize the state-output transition probability; 3) determine the dimension of the modal dynamics, i.e. the number of clusters (modes) of the system; 4) determine the partition regions in discretized state space corresponding to the clusters (modes); 5) find the modal transition probability matrix. With these objectives, our system identification approach includes the following three steps:

- In the first step, the underlying state sequence is estimated directly from the output series. We adopt the newly developed algorithm [53] for this purpose. Moreover, we analyze the system attributes of (1.1) needed for the application of this algorithm.
- Given the state sequence estimated from Step 1, we estimate state Markovian transition matrix and the state-output transition matrix by a counting type algorithm.
- Given the state transition matrix estimated in Step 2, we determine the number of modes, determine the corresponding modal regions of the state space and calculate the mode transition matrix that describes the modal behavior. This step is based on the previous work on model reduction of non-reversible Markov processes [38].

The proposed algorithm is illustrated in Figure 1.

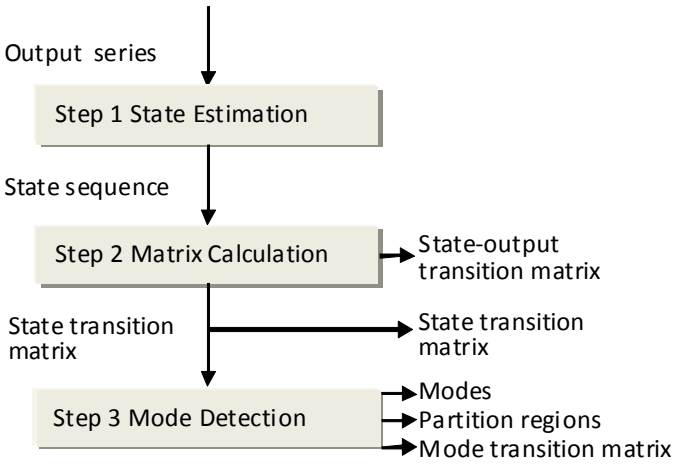


Figure 1 Proposed system identification approach

The three parts of the algorithm will be described in detail next and examples will be given to illustrate the proposed approach.

3.1.1 Estimation of State Process from Output Process

Consider the discretized system corresponding to (1.1), we can treat the discretized observation sequence $\bar{y}_k$ as the output of a finite state hidden Markov chain with $\bar{x}_k$ being the underlying finite state Markov process. Estimation of the state process from the observation sequence $\bar{y}_{1:L}$ is one of the principal tasks in HMM modelling. In [16] several algorithms for this purpose are discussed.

In this dissertation, we adopt the newly developed non-negative matrix factorization (NMF) based algorithm presented in [53] for this estimation purpose. This novel algorithm has been shown, in a 2-dimensional (denoted as $N = 2$) state space case, to lead to much better estimation accuracy compared to that of the classical Baum-Welch HMM identification algorithm. However, as we will see later, unlike the 2-dimensional state space case, when $N > 2$ the uniqueness of NMF is not guaranteed, which in turn results in non-unique state estimation. We begin in this subsection by introducing the NMF based hidden state process identification algorithm proposed in [53]. Moreover, we will extend the algorithm to high dimensional state space case as well as explore the system attributes that lead to a unique NMF based state estimation solution.

Non-negative Matrix Factorization based HMM identification in [53]

In this subsection we describe the NMF based HMM identification algorithm proposed in [53]. We begin by stating two assumptions made in [53] and discuss how these assumptions are satisfied in our system identification context. The first assumptions is that the HMM is ergodic. We note that, in our system identification context, if the underlying system can be decomposed into strong attraction domains then ergodicity of the Markov process (1.1) in an appropriate sense is a natural consequence. Furthermore, if (1.1) is ergodic then

the discretized process is ergodic as well. The second assumption is that the underlying states have initial distribution that is equal to a stationary distribution π of the HMM. This assumption can be justified as follows. For an ergodic process the distribution converges to the invariant distribution at a rate $|\lambda_1|^n$ where n denotes time and λ_1 is the eigenvalue of the transition matrix with the second largest magnitude, $|\lambda_1| < 1$. Consequently, in our system identification context, by choosing n sufficiently large the distribution can be made arbitrarily close to π. If data records are long and we dismiss the first n data points the remaining record looks like a data record from a system with initial distribution close to π.

Since the NMF based HMM estimation algorithm in [53] considers a Mealy HMM model, we will differentiate the Mealy HMM model and the alternatively defined HMM model, i.e. the Moore HMM, in order to facilitate the following discussion. The following definition of these two types of HMM models are taken from [54]:

Definition 4 *A Mealy HMM is defined as a quadruple (X, Y, Π, π), where X and Y are the finite state and output alphabet, respectively; Π is a mapping from Y to $\mathbb{R}_+^{|X|\times|X|}$ with the matrix $\Pi_X := \sum_{y\in Y}\Pi(y)$ such that $\Pi_X e = e$, $\Pi_{ij}(y) = P\{\bar{x}_{t+1} = x^j, \bar{y}_t = y \mid \bar{x}_t = x^i\}$; π is the initial state distribution. Note here $|X|$ denotes the size of state alphabet.*

Definition 5 *A Moore HMM is specified by $(X, Y, \Pi_X, \beta, \pi)$, where X , Y and π have the same meanings as in Mealy HMM; while Π_X with $\Pi_X e = e$ is the state transition matrix, defined as $(\Pi_X)_{ij} = P\{\bar{x}_{t+1} = x^j \mid \bar{x}_t = x^i\}$; β is a mapping from Y to $\mathbb{R}_+^{|X|}$ with $\beta_i(y) = P\{\bar{y}_t = y \mid \bar{x}_t = x^i\}$.*

We remark that in the Mealy formulation for HMM the state and output probabilities are combined in the map Π whereas in the alternative Moore formulation these two quantities are treated separately.

As in [53], define $\Pi(y) \triangleq \{\Pi_{ij}(y)\}$, where $\Pi_{ij}(y) = P\{\bar{x}_{t+1} = x^j, \bar{y}_t = y \mid \bar{x}_t = x^i\}$ and let $P(\mathbf{u}) \triangleq P(\bar{y}_{1:|\mathbf{u}|} = \mathbf{u}) = P(\bar{y}_1 = u_1, \bar{y}_2 = u_2, \ldots \bar{y}_{|\mathbf{u}|} = u_{|\mathbf{u}|})$ be an output string probability, where $u_i \in \{1, \ldots M\}$ and $\mathbf{u} = u_1 \ldots u_{|\mathbf{u}|}$ with $|\mathbf{u}|$ denoting the length of $\mathbf{u}$. Denote $P(\bar{x}_1 = x^i)$

by π_i (recall that π is invariant distribution). Then it is easy to show that

$$P(\mathbf{u}) = \pi\Pi(\mathbf{u})e \tag{3.1}$$

where $e = [1,...,1]^T$, $\Pi(\mathbf{u}) = \Pi(u_1)\Pi(u_2)\cdots\Pi(u_{|\mathbf{u}|})$ and $\pi = [\pi_1, \cdots \pi_N]$.

Consider now a matrix $V(i_1, i_2)$ with entries $(V(i_1, i_2))_{kl} = P(\mathbf{u}_k\mathbf{v}_l)$, where $\mathbf{u}_k$ denotes the kth possible i_1 length output string: $\mathbf{u}_k = \bar{y}_1\bar{y}_2\cdots\bar{y}_{i_1}$, $\mathbf{v}_l$ denotes the lth possible i_2 length output string: $\mathbf{v}_l = \bar{y}_{i_1+1}\bar{y}_{i_1+2}\cdots\bar{y}_{i_1+i2}$, with $k \in \{1,...M^{i_1}\}$, $l \in \{1,...M^{i_2}\}$, where as noted earlier M is the alphabet size for the output sequence; $P(\mathbf{u}_k\mathbf{v}_l)$ denotes probability of the $i_1 + i_2$ length output string concatenated by $\mathbf{u}_k$ and $\mathbf{v}_l$. Note that since the distribution of the states is assumed to be stationary, the probability of one specific $i_1 + i_2$ length output string is the same at all times. Thus, with the assumption of ergodicity, by observing L length output series with $L >> M^{i_1+i_2}$, $P(\mathbf{u}_k\mathbf{v}_l)$ is estimated by:

$$P(\mathbf{u}_k\mathbf{v}_l) \approx \frac{\text{Number of times } \mathbf{u}_k\mathbf{v}_l \text{ is detected}}{L - (i_1 + i_2) + 1}$$

By running through all possible string combinations we obtain the matrix V. From (3.1) we obtain,

$$P(\mathbf{u}_k\mathbf{v}_l) = \pi\Pi(\mathbf{u}_k\mathbf{v}_l)e = \pi\Pi(\mathbf{u}_k)\Pi(\mathbf{v}_l)e$$

Define a $M^{i_1} \times N$ matrix W^{tr} and a $N \times M^{i_2}$ matrix H^{tr} as

$$W^{tr} = \begin{bmatrix} \pi\Pi(\mathbf{u}_1) \\ \pi\Pi(\mathbf{u}_2) \\ \vdots \\ \pi\Pi(\mathbf{u}_{M^{i_1}}) \end{bmatrix}$$

$$H^{tr} = \begin{bmatrix} \Pi(\mathbf{v}_1)e & \Pi(\mathbf{v}_2)e & \cdots & \Pi(\mathbf{v}_{M^{i_2}})e \end{bmatrix}$$

It is easy to see that

$$V = W^{tr}H^{tr} \tag{3.2}$$

Note that N, the size of state alphabet, which satisfies $N < \min(M^{i_1}, M^{i_2})$, is assumed to be known (the N is bounded below by the rank of V and can be estimated by calculating the so-called positive rank of V, see [52]). Moreover, a simple calculation shows that the elements of W^{tr} are

$$W^{tr}_{k,i} = P(\bar{y}_{1:i_1} = \mathbf{u}_k, \bar{x}_{i_1+1} = x^i)$$

while the elements of H^{tr} are

$$H^{tr}_{i,l} = P(\bar{y}_{i_1+1:i_2} = \mathbf{v}_l \mid \bar{x}_{i_1+1} = x^i)$$

Assuming W^{tr} is known, (e.g. obtained through the factorization of V), define

$$U^{tr} = (diag(W^{tr}e))^{-1}W^{tr}$$

Then

$$U^{tr}_{k,i} = P(\bar{x}_{i_1+1} = x^i \mid \bar{y}_{1:i_1} = \mathbf{u}_k)$$

Thus given $\bar{y}_{1:i_1} = \mathbf{u}_k$, the estimated state can be chosen as the maximum likelihood estimate $\hat{x}_{i_1+1} = x^{\arg\max_\lambda U_{k,\lambda}}$. With the ability to estimate $\hat{x}_{i_1+1}$ based on each possible i_1 length output string $\bar{y}_{1:i_1}$, the state sequence $\hat{x}_{i_1+1:D+1}$ can be estimated by sliding a window of fixed length i_1 through the D length output series.

Note a critical step in applying this algorithm is utilizing NMF to calculate W^{tr} and H^{tr} based on the V matrix. We emphasize that here W^{tr} and H^{tr} are entities with specific physical meanings. That is, the uniqueness of NMF factorization is of principal importance. As we discussed in Section 2.6, when $N = 2$ (here we use N in place of the r in Section 2.6), NMF has a unique solution up to a permutation matrix. In this state estimation context,

$N = 2$ in Section 2.6 corresponds to that the state space being 2-dimensional. That is, when the state space is 2-dimensional, the NMF based estimation algorithm has unique solution up to a permutation matrix. Moreover, it has been shown in [53] that in the 2-dimensional state space case, the above algorithm can lead to much better estimation accuracy compared to that of the classical Baum-Welch HMM identification algorithm [53]. However, as we mentioned in Section 2.6, the uniqueness of NMF is not guaranteed for $N > 2$, which in turn results in non-unique state estimation.

In the next subsection we will extend the NMF based HMM identification algorithm to high-dimensional state space, i.e. $N > 2$, case. Moreover, we will explore system attributes which result in a unique solution to the NMF based estimation problem in the general $N > 2$ case. As we will see later, in the stochastic system state estimation context, utilizing further structural constraints on W^{tr} and H^{tr} combined with other system attributes leads to a unique NMF factorization of V in the sense that any T matrix obtained in NMF algorithm described in Section 2.6 is a permutation (or an approximate permutation) which, in turn, results in a unique NMF based state estimation. Since the state dimension N in the estimation context corresponds to r in NMF formulation in Section 2.6, in the following we will use N instead of r in analyzing NMF algorithm.

Non-negative Matrix Factorization based HMM identification for High-dimensional State Space

In [53], a HMM identification algorithm has been proposed based on NMF approach. However, as we mentioned before, unlike the 2-dimensional state space case, when $N > 2$, the unique estimation based on NMF is not guaranteed. We note that this issue was not discussed in [53].

In this subsection, we will extend the algorithm in [53] to high dimensional state space case and explore the system attributes that lead to unique NMF based state estimation solution. The NMF based HMM estimation algorithm here is similar to that in [53]. Our

effort here is concentrated on the discussion about system attributes that lead to unique estimation in context similar to that in [53].

Here we consider small noise disturbed stochastic system because in many cases, the process noise and the observation noise are originated from system parameter variations and sensor noise, which frequently makes the noises small. In such instances the stochastic system can be treated as a deterministic system disturbed by small noise. It has to be stressed again that by small noise we mean that the relative value of the noise to typical values of the system state and output is small.

We consider the small noise disturbed stochastic system (2.1) described in Section 2.2 and treat that as the deterministic system (2.2) disturbed by small noise. We remark that if the same discretization is used in the state space of system (2.1) and system (2.2) and the discretization is fine enough, then one can easily extend Propositions 3 and 4 to the discretized systems. We omit the details of these extensions. In the remainder of this section when we refer to the stochastic and deterministic systems we mean the discretized versions of (2.1) and (2.2).

Now we extend the state estimation based on NMF to high dimensional ($N > 2$) state space case.

Uniqueness of State Estimation The discussion in this section is divided into three parts. First, we will develop conditions on W^{tr} and H^{tr} that make NMF have unique solution up to a permutation matrix T. Second, attributes of the stochastic system will be related to the conditions on W^{tr} and H^{tr}. Finally, the uniqueness of NMF based state estimation for a system with the given attributes will be proved.

Note, as before, in the following discussion by W^{tr} and H^{tr} we mean the true matrices of interest, by W and H we mean the matrices obtained by NMF from the given V matrix, and by T matrix we mean the matrix relating W and W^{tr} (or H and H^{tr}) through $W = W^{tr}T$ (or $H = T^{-1}H^{tr}$). Note as well that matrices W^{tr}, H^{tr}, W and H all contain only non-negative

entries.

Matrix Constraints In this subsection we first explore the conditions on W^{tr} and H^{tr} which make the T matrix a permutation matrix, i.e. make NMF have unique solution up to a permutation matrix. Second, we will relax the conditions so that the T matrix is an almost permutation matrix, i.e. T has the form $T = T_0 + \varepsilon L$ where T_0 is a permutation, L is an arbitrary matrix and ε is a small parameter. The relaxed conditions will be related to attributes of the stochastic system in the next subsection. Moreover, as we will see later, with T matrix being an almost permutation matrix, the NMF based state estimation is still unique.

Proposition 5 *If all rows of H^{tr} and H sum to 1, then the T and T^{-1} matrices have rows that sum to 1 as well.*

Proof. Since $H^{tr} = TH$, we have $H^{tr}_{i,l} = \sum_k T_{i,k} H_{k,l}$ and

$$\begin{aligned} \sum_l H^{tr}_{i,l} &= \sum_l \sum_k T_{i,k} H_{k,l} \\ &= \sum_k T_{i,k} \sum_l H_{k,l} \\ &= \sum_k T_{i,k} = 1, \forall i \end{aligned}$$

That is, $\sum_k T_{i,k} = 1, \forall i$. In similar way, we also have $\sum_k T^{-1}_{i,k} = 1, \forall i$. ■

Define a matrix $U^{tr} = (diag(W^{tr}e))^{-1}W^{tr}$ and let $U = (diag(We))^{-1}W$, then

$$\begin{aligned} U &= (diag(We))^{-1}W = (diag(W^{tr}Te))^{-1}W^{tr}T \\ &= (diag(W^{tr}e))^{-1}W^{tr}T = U^{tr}T \end{aligned}$$

Here we see the T matrix relating U and U^{tr} is the same as the one relating H and H^{tr} as well as W and W^{tr}. Furthermore, through the construction of U and U^{tr} we see that these two matrices contain only nonnegative entries.

Proposition 6 *If, after appropriate reordering of columns and rows, H^{tr} and U^{tr} contain a $N \times N$ identity submatrix respectively, then T and T^{-1} contain only non-negative entries.*

Proof. Denote the $N \times N$ identity submatrix contained in H^{tr} as H_1^{tr}. Then there is a submatrix contained in H, denoted by H_1, which satisfies $T^{-1}H_1^{tr} = H_1$. It is easy to see that since H_1^{tr} and H_1 both contain positive entries and H_1^{tr} is a identity matrix, T^{-1} contains only non-negative entries.

Similarly, if U^{tr} contain $N \times N$ identity submatrix, we can show that T contains only non-negative entries. ■

Proposition 7 *If T and T^{-1} both have nonnegative entries, and $\sum_j T_{i,j} = 1, \forall i$, then T and T^{-1} are both permutation matrices.*

Proof. Let

$$T = \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1N} \\ T_{21} & T_{22} & \cdots & T_{2N} \\ \vdots & & & \vdots \\ T_{N1} & T_{N2} & \cdots & T_{NN} \end{bmatrix}$$

and

$$Q = T^{-1} = \begin{bmatrix} Q_{11} & Q_{12} & \cdots & Q_{1N} \\ Q_{21} & Q_{22} & \cdots & Q_{2N} \\ \vdots & & & \vdots \\ Q_{N1} & Q_{N2} & \cdots & Q_{NN} \end{bmatrix}$$

then

$$\begin{aligned} \sum_j T_{i,j}Q_{j,l} &= \delta(i-l), \sum_j Q_{i,j}T_{j,l} = \delta(i-l), \\ & \sum_j T_{i,j} = 1, \ \forall i \end{aligned}$$

Since $T_{i,j} \geq 0$ and $Q_{j,l} \geq 0$, we get

$$\{T_{i,j} \neq 0 \Longrightarrow Q_{j,l} = 0\}, \forall i \neq l, \forall j$$

We know that $rank(T) = N$. Therefore for any i, there exists j s.t. $T_{i,j} \neq 0 \Rightarrow Q_{j,l} = 0$, $l = 1, 2, \cdots, i-1, i+1, \cdots N$, that means $Q_{j,i} \neq 0$ (otherwise $rank(Q) \neq N$), which in turn means that $T_{i,l} = 0$, $\forall l \neq j$. So we see that in the ith row of T there is only one non-zero entry, which is $T_{i,j}$. This is true for any i. Accordingly, we see that T matrix has a general permutation form. Moreover, since $\sum_j T_{i,j} = 1$ and $T_{i,j} \geq 0, \forall i, j$, we conclude that T is a permutation matrix. With T being a permutation matrix, it is easy to show that T^{-1} is also a permutation matrix. ■

We conclude that under the above conditions where T is a permutation matrix, we know the solution H and W of NMF is equivalent to the H^{tr} and W^{tr} matrices up to a permutation matrix T. We next extend this result to the case where H^{tr} and W^{tr} have an "almost" identity submatrix. This corresponding to a system perturbed by small noise.

Proposition 8 *Assume H^{tr} and U^{tr} contain a submatrix of the form $I - \varepsilon L$, where I denotes the $N \times N$ identity submatrix, ε is a small parameter and L an arbitrary $N \times N$ matrix. Then T and T^{-1}, denoted as T_ε and T_ε^{-1} are almost permutation matrices, i.e. T_ε has the form $T_\varepsilon = T_0 + O(\varepsilon)\tilde{L}$, where T_0 is a permutation and $\tilde{L}$ is some $N \times N$ matrix (and similarily for T_ε^{-1}).*

Proof. To emphasize the dependence on ε we denote H^{tr} and U^{tr} as H_ε^{tr} and U_ε^{tr} We assume that U_ε^{tr} has a submatrix of the form $I - \varepsilon L$. Recall that for any other U_ε derived from a factorization of V_ε (again we emphasize the dependence on ε by replacing V by V_ε) we have the relationship $U_\varepsilon = U_\varepsilon^{tr} T_\varepsilon$ and thus $U_\varepsilon^1 = (I - \varepsilon L) T_\varepsilon$ for some submatrix U_ε^1 (the almost identity submatrix) of U_ε. We note that the entries of U_ε^1 take values in the interval $[0, 1]$. It follows that the limit of U_ε^1 as $\varepsilon \to 0$ exists and we denote it as U_0^1. We also note that for small enough ε we have $(I - \varepsilon L)^{-1} = \sum_{i=0}^{\infty} (\varepsilon L)^i = I + \varepsilon \sum_{i=1}^{\infty} \varepsilon^{i-1} L^i = I + \varepsilon F_\varepsilon$. It

follows that $(I-\varepsilon L)^{-1} \to I$ as $\varepsilon \to 0$. For small enough ε we have $T_\varepsilon = (I-\varepsilon L)^{-1} U_\varepsilon^1$ and thus $T_\varepsilon \to T_0 = U_0^1$ as $\varepsilon \to 0$. Note that similarly, from $H_\varepsilon = K_\varepsilon H_\varepsilon^{tr}$ we can show that K_0 exists. We want to show that T_0 and K_0 are permutations.

Recall that $V_\varepsilon = W_\varepsilon^{tr} H_\varepsilon^{tr}$ and for any other factorization of V_ε we have $V_\varepsilon = W_\varepsilon H_\varepsilon = W_\varepsilon^{tr} T_\varepsilon T_\varepsilon^{-1} H_\varepsilon^{tr}$. Under the assumption that U_ε^{tr} and H_ε^{tr} have almost identity submatrices and using the fact that their entries belong to $[0,1]$ we know that the limits $\lim_{\varepsilon\to 0} W_\varepsilon^{tr} = W_0^{tr}$ and $\lim_{\varepsilon\to 0} H_\varepsilon^{tr} = H_0^{tr}$ exist and thus $V_0 = W_0^{tr} H_0^{tr}$. Furthermore, since W_0^{tr}, H_0^{tr}, T_0 and K_0 exist we have $V_0 = W_0 H_0 = W_0^{tr} T_0 K_0 H_0^{tr}$, where K_0 can be shown to be $K_0 = T_0^{-1}$. Thus $W_0 = W_0^{tr} T_0$ and $H_0 = T_0^{-1} H_0^{tr}$ exists.

Finally, as U_0^{tr} and H_0^{tr} both have an identity submatrix, from Proposition 3.1.1 we know that any T_0^{-1} that satisfies $T_0^{-1} H_0^{tr} = H_0$ with H_0 being the limit as $\varepsilon \to 0$ of H_ε will be a permutation. Similarly we see that T_0 is a permutation.

With T_0 being a permutation matrix, we conclude that T_ε is an almost permutation matrix, denoted as $T_\varepsilon = T_0 + O(\varepsilon)\tilde{L}$, because $T_\varepsilon \to T_0$ as $\varepsilon \to 0$. In similar way, we can prove that T_ε^{-1} is also an almost permutation matrix. ■

System Attributes After working out the conditions on H^{tr} and U^{tr} which lead to uniqueness of the NMF, we will next specify system attributes that correspond to the matrix conditions. Note that the W^{tr}, H^{tr} and U^{tr} matrices defined in Section 3.1.1 in the stochastic system context match the ones used in the following. For the discretized deterministic system let $\bar{y}_{1:D}(\bar{x})$ denote the output string of length D originating at $\bar{x}_0 \in X$.

We begin with several assumptions about the system under study,

A1 The discretized deterministic system corresponding to (2.2) is observable. That is, there exists a $K > 0$ such that any initial state $\bar{x}_0 \in X$ can be uniquely (with probability 1) determined by the corresponding $\bar{y}_{1:K}(\bar{x}_0)$. Note that since, in the discretized deterministic system, each $\bar{x}_0$ leads to a unique output string $\bar{y}_{1:K}$, we can say that the initial state $\bar{x}_0$ and the output string $\bar{y}_{1:K}$ are in a one to one correspondence for the

observable system.

A2 The stochastic system of interest is given by (2.1) for some $0 < \varepsilon \ll 1$.

A3 The probability of any state at any time is bounded below.

Proposition 9 *Define* $H^{tr}_{i,l} = \Pr(\bar{y}^{\varepsilon}_{1:K} = \mathbf{v}_l \mid \bar{x}^{\varepsilon}_0 = x^i)$ *and assume A1 and A2. Then the matrix* $H^{tr} \in M^K \times N$ *contains an almost identity submatrix.*

Proof. Referring to the discretized version of Proposition 4 we see that, originating from a specific initial state $\bar{x}_0$, the stochastic system will have the same output string as the corresponding deterministic system with probability close to 1; while output strings which are not equal to that of the deterministic system will have probability close to 0. This is true for every $\bar{x}_0$. Consequently, in each row of H^{tr} there will be one element close 1, while all the other elements are close to 0. As we mentioned before, for the observable deterministic system, the initial states and possible output strings are in a one to one correspondence, thus we will have in each column of the H^{tr} matrix at most one element close to 1. That is, H^{tr} contains an almost identity submatrix of the form $I - \varepsilon L$, where ε is small. ■

Proposition 10 *Define* $U^{tr}_{k,i} = \Pr(\bar{x}^{\varepsilon}_{i_1+1} = x^i \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k)$ *and assume A1-A3. Then the matrix* $U^{tr} \in N \times M^{i_1}$ *contains an almost identity submatrix.*

Proof. Recall that

$$\begin{aligned} U^{tr}_{k,i} &= \Pr(\bar{x}^{\varepsilon}_{i_1+1} = x^i \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k) \\ &= \textstyle\sum_{x_{0l}} \Pr(\bar{x}^{\varepsilon}_{i_1+1} = x^i \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k, \bar{x}^{\varepsilon}_0 = x_{0l}) \\ &\qquad \times \Pr(\bar{x}^{\varepsilon}_0 = x_{0l} \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k) \end{aligned} \tag{3.3}$$

Consider the $\Pr(\bar{x}^{\varepsilon}_{i_1+1} = x^i \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k, \bar{x}^{\varepsilon}_0 = x_{0l})$ part in (3.3) first. For the deterministic system, given x^i, there always exists a initial state x_{0i} s.t.

$$\Pr(\bar{x}_{i_1+1} = x^i \mid \bar{x}_0 = x_{0i}) = 1$$

Then for stochastic system, referring to the Proposition 3, we have

$$\Pr(\bar{x}^{\varepsilon}_{i_1+1} = x^i \mid \bar{x}^{\varepsilon}_0 = x_{0i}) \approx 1$$

and since

$$\begin{aligned}\Pr(\bar{x}^{\varepsilon}_{i_1+1} = x^i \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k, \bar{x}^{\varepsilon}_0 = x_{0i}) \\ \geq \Pr(\bar{x}^{\varepsilon}_{i_1+1} = x^i \mid \bar{x}^{\varepsilon}_0 = x_{0i})\end{aligned}$$

we get that

$$\Pr(\bar{x}^{\varepsilon}_{i_1+1} = x^i \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k, \bar{x}^{\varepsilon}_0 = x_{0i}) \to 1$$

Consider now the $\Pr(\bar{x}^{\varepsilon}_0 = x_{0l} \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k)$ part in (3.3). From the proof of Proposition 9 we know that $\Pr(\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k \mid \bar{x}^{\varepsilon}_0 = x_{0i}) \to 1$ as $\varepsilon \to 0$ when $\mathbf{u}_k$ is the same as the output string resulting from initial state x_{0i} in the deterministic system. Moreover,

$$\begin{aligned}\Pr(\bar{y}^{\varepsilon}_{1:i_1} &= \mathbf{u}_k \mid \bar{x}^{\varepsilon}_0 = x_{0i}) \\ &= \frac{\Pr(\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k, \bar{x}^{\varepsilon}_0 = x_{0i})}{\Pr(\bar{x}^{\varepsilon}_0 = x_{0i})} \\ &= \frac{\Pr(\bar{x}^{\varepsilon}_0 = x_{0i} \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k)\Pr(\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k)}{\Pr(\bar{x}^{\varepsilon}_0 = x_{0i})} \\ &= \frac{\Pr(\bar{x}^{\varepsilon}_0 = x_{0i} \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k)}{\Pr(\bar{x}^{\varepsilon}_0 = x_{0i})} \\ &\quad \times \textstyle\sum_{x_{0l}} \Pr(\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k \mid \bar{x}^{\varepsilon}_0 = x_{0l})\Pr(\bar{x}^{\varepsilon}_0 = x_{0l})\end{aligned} \tag{3.4}$$

Since we know that $\Pr(\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k \mid \bar{x}^{\varepsilon}_0 = x_{0i}) \to 1$ as $\varepsilon \to 0$, it is easy to show $\Pr(\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k \mid \bar{x}^{\varepsilon}_0 = x_{0j}, j \neq i) \to 0$, assuming that $i_1 \geq K$. Thus we have

$$\begin{aligned}&\textstyle\sum_{x_{0l}} \Pr(\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k \mid \bar{x}^{\varepsilon}_0 = x_{0l})\Pr(\bar{x}^{\varepsilon}_0 = x_{0l}) \\ &\quad \approx \Pr(\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k \mid \bar{x}^{\varepsilon}_0 = x_{0i})\Pr(\bar{x}^{\varepsilon}_0 = x_{0i}) \\ &\quad = \Pr(\bar{x}^{\varepsilon}_0 = x_{0i})\end{aligned}$$

Finally we can express (3.4) as

$$\begin{aligned}
&\Pr(\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k \mid \bar{x}^{\varepsilon}_0 = x_{0i}) \\
&\quad \approx \frac{\Pr(\bar{x}^{\varepsilon}_0 = x_{0i} | \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k)}{\Pr(\bar{x}^{\varepsilon}_0 = x_{0i})} \Pr(\bar{x}^{\varepsilon}_0 = x_{0i}) \\
&\quad = \Pr(\bar{x}^{\varepsilon}_0 = x_{0i} \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k)
\end{aligned}$$

That means we can treat $\{\bar{x}^{\varepsilon}_0 = x_{0i}\}$ and $\left\{\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k\right\}$ as almost the same events, and $\Pr(\bar{x}^{\varepsilon}_0 = x_{0i} \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k) \to 1$ as $\varepsilon \to 0$. Moreover, we get $\Pr(\bar{x}^{\varepsilon}_0 = x_{0j}, j \neq i \mid \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k) \to 0$.

Since in the deterministic system, for each x_{0i}, there is exactly one x^i that it reaches in $i+1$ steps, then referring to the Proposition 3 we have $\Pr(\bar{x}^{\varepsilon}_{i_1+1} = x^i \mid \bar{x}^{\varepsilon}_0 = x_{0i}) \to 1$, and accordingly $\Pr(\bar{x}^{\varepsilon}_{i_1+1} = x^j, j \neq i \mid \bar{x}^{\varepsilon}_0 = x_{0i}) \to 0$ as $\varepsilon \to 0$. Moreover, since we can treat $\{\bar{x}^{\varepsilon}_0 = x_{0i}\}$ and $\left\{\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k\right\}$ as almost the same events, we have $\Pr(\bar{x}^{\varepsilon}_{i_1+1} = x^j, j \neq i \mid \bar{x}^{\varepsilon}_0 = x_{0i}, \bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k) \to 0$ as $\varepsilon \to 0$.

In summary we have the following: if $x_{0i} \to x^i$ in deterministic system in $i+1$ steps, with $\{\bar{x}^{\varepsilon}_0 = x_{0i}\} \sim \left\{\bar{y}^{\varepsilon}_{1:i_1} = \mathbf{u}_k\right\}$, we have $U^{tr}_{k,i} \approx 1$; on the other hand, if x^i and $\mathbf{u}_k$ do not match through any x_{0i}, then $U^{tr}_{k,i} \approx 0$. That is, U^{tr} matrix contains almost identity submatrix.

■

Unique Estimation Up to this point, we have shown that, if the deterministic system is observable and the noise intensity in the stochastic system is small, then U^{tr} and H^{tr} both contain an almost identity submatrix. Combining this with the proof of Proposition 10 we know that the solution $W = W^{tr}T$, $H = T^{-1}H^{tr}$ to NMF algorithm in the stochastic system state detection has the feature that T is an almost permutation matrix. Given such T, we will prove the uniqueness of the NMF algorithm based estimation.

Proposition 11 *If T is an almost permutation matrix, and the noises in the stochastic system are small, then the optimal state estimates obtained from U and U^{tr} are the same.*

Proof. As is shown in [53], for a given U the optimal maximum likelihood state estimate for the observed output string $\bar{u}_k$ of length i_1 is $\hat{x}_{i_1+1} = \arg\max_l U_{k,l}$. Let the corresponding optimal estimate obtained from U^{tr} be $\hat{x}^{tr}_{i_1+1}$. We know that $U = U^{tr}T$ where T is a nonsingular matrix. Furthermore, in the small noise case we have established that T has the form $T = T_0 + O(\varepsilon)L$ where T_0 is a permutation matrix and the small parameter ε represents the noise intensity. From this we see that U is just the U^{tr} matrix with different column order plus small additive noise of the order $O(\varepsilon)$. That is, the entry of order one in the kth row of U is in a one to one correspondence to the order one entry in corresponding row in U_{tr} through the permutation T_0. Therefore, the optimal state estimation obtained from U and U^{tr} is the same as long as the noise intensity is sufficiently small. ■

We note that Proposition 11 is equivalent to the statement that the state estimation by NMF for the stochastic system is unique, provided the noise intensity is small enough.

Example In this subsection we will apply the NMF based state estimation algorithm to a discrete time discrete state stochastic system which is represented by an HMM. The system is constructed from an observable deterministic system that is subject to small process and output noises. We will experimentally verify the uniqueness estimation analysis presented in above as well as show the effectiveness of the proposed NMF based estimation algorithm.

Example 2 *The stochastic system under consideration has a four state alphabet* $\{1,2,3,4\}$ *corresponding to 4 real states, i.e.* $(x_k(1),x_k(2)) \in \{(1,1),(1,2),(2,1),(2,2)\}$. *The state transition matrix is denoted by P. The unperturbed output is given by* $x_k(1)+x_k(2)$ *and is easily seen to take real values in* $\{2,3,4\}$, *which corresponds to a size 3 output alphabet* $\{1,2,3\}$. *We assume that the output noise is such that the noise perturbed output process*

$$y_k = x_k(1) + x_k(2) + \bar{v}_k \tag{3.5}$$

takes values in $\{2,3,4\}$ *as well.*

State and output series were generated according to the state transition matrix P and the state-output relationship. In the following, we will consider the alphabet of the state and output rather than the real ones.

Consider a deterministic system represented by a Markov model with

$$P_0 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

and a state-output equation

$$y_k = x_k(1) + x_k(2)$$

Then we can easily see that this system is observable in $K = 2$ steps, where the following pairs have one to one relationship

x_i	1	2	3	4
$y_{i+1:i+2}$	$1,2$	$2,3$	$2,2$	$3,1$

Assume that by adding small noise to the system dynamics the Markov transition matrix P becomes

$$P = \begin{bmatrix} * & * & 0.95 & * \\ * & * & * & 0.95 \\ * & 0.95 & * & * \\ 0.95 & * & * & * \end{bmatrix}$$

where $*$ are small values. Then, with small output noise added as well, we have constructed a stochastic system from the observable deterministic system. Based on the output series generated by this stochastic system, we obtain from U generated by NMF and the true U^{tr}

the T matrix

$$T = \begin{bmatrix} 0.00 & 0.89 & 0.14 & -0.02 \\ 0.96 & 0.00 & 0.09 & -0.05 \\ -0.09 & 0.00 & 0.18 & 0.91 \\ 0.00 & -0.07 & 1.06 & 0.00 \end{bmatrix}$$

Clearly, in this case, the NMF factorization yields a factorization that is equivalent to the true matrices up to an almost permutation matrix. Furthermore, based on the calculated U matrix we see that the state series are estimated with 5.29% error rate.

In summary, through the example system simulation, we see that when NMF is applied to a system with the attributes specified earlier, the NMF based estimation is unique, and the estimation error rate is low, which not only verifies our previous analysis but also illustrates the effectiveness of NMF based estimation algorithm in this type of system.

Conclusion

In this section we introduced the novel NMF based HMM state estimation algorithm proposed in [53]. As we treat the stochastic system of interest as equivalent to a HMM, we can apply the same algorithm for the stochastic system state process estimation. Recognizing that the uniqueness of NMF based hidden state estimation is only satisfied in 2-dimensional state space case, we developed conditions under which NMF will have unique solution in the stochastic system state estimation context. Moreover, we explored the system attributes corresponding to those conditions. In this way, we extended the NMF based state estimation algorithm to high-dimensional state space case. An example was presented to illustrate the uniqueness of NMF based estimation under the stated conditions and to manifest the effectiveness of the proposed NMF based state estimation algorithm in systems that satisfy the constraints. The proposed algorithm estimates the state purely based on output series. It can be treated as a data driven state estimation algorithm and the estimated state can be used further to identify the system model, e.g. the state transition matrix, system modal

dynamics, and so on.

3.1.2 Transition Matrix Calculation

So far we have the developed an approach for estimating a state sequence from the given output series. Next we develop methods for calculating the state transition matrix and the state-output matrix. Here a counting measure m_c is introduced to facilitate the description.

Given a L length time series evolving according to an ergodic Markov model, the number of times the process visits set A_i is denoted by $(m_c(A_i))_L$, while the number of times it visits set A_j in one step from A_i is $(m_c(A_j|A_i))_L$. Let $A_i = \{x^i\}$, then the transition probability of the Markov chain is given by:

$$P(x^j \mid x^i) = \lim_{L\to\infty} P^{(L)}(x^j \mid x^i)$$

where

$$P^{(L)}(x^j \mid x^i) = \frac{(m_c(A_j|A_i))_L}{(m_c(A_i))_L}$$

The state-output matrix is calculated in similar fashion.

3.1.3 Identification of Multi-modal Behavior Based on State Process

Recall we are interested in a multi-modal dynamic system with states evolving according to a non-reversible ergodic Markov chain on a finite dimensional state space $X = \{x^1,\ldots,x^N\} \triangleq \{1,\ldots,N\}$. In the previous section we estimated the transition matrix P of the non-reversible Markov chain. We discuss next how the system modal behavior can be extracted from the state transition matrix P.

As mentioned in Section 2.4, the modal behavior analysis of SAFHS is closely related to the characterization of metastable partition of the space. Thus the previously developed technique in [38] which identifies the metastable partition for a general non-reversible finite dimensional Markov chain is used here for this purpose. We briefly present the developed

technique in two steps: 1) at given time n, construct a reduced order approximate operator π_{an}^{ν} for π_n^{ν} that is good for all initial distributions ν such that π_{an}^{ν} captures the clustering phenomena of original system. Here π_n^{ν} denotes the system distribution at time n with initial distribution ν. Note that in finite dimensional Markov system, we have $\pi_n^{\nu} = \nu P^n$. In particular, through spectral analysis of π_n^{ν} we detect the number of modes (number of clustering components); 2) based on π_{an}^{ν} we detect the modal dynamics of the original system, and estimate the partition in the state space of the original system accordingly.

Find reduced order approximate operator

Consider the non-reversible ergodic Markov chain with transition matrix P and define the *multiplicative reversibilization* $M(P)$ of P by

$$M(P) = P\tilde{P}$$

where $\tilde{P} = D^{-1}P^T D$, $D = diag(\pi_0, \ldots, \pi_{N-1})$ and $\pi = [\pi_0, \ldots, \pi_{N-1}]$ is the unique stationary distribution for P with $\pi_i > 0, i = 0, \ldots, N-1$. Then $M(P)$ is a reversible transition matrix which has eigenvalues $\{\beta_0, \ldots \beta_{N-1}\}$ with $1 = \beta_0 \geq \beta_1 \geq \ldots \geq \beta_{N-1} \geq 0$, where $\beta_i = |\lambda_i|^2, i = 0, \ldots, N-1$, with λ_i denoting the eigenvalues of P. Moreover, $M(P)$ has the same stationary distribution π as P.

The following theorem has been proved in [38].

Theorem 12 *Assume that P has a unique stationary distribution π with $\pi_i > 0, i = 0, \ldots, N-1$. Then the weighted distance between the distributions νP^n and $\nu \bar{P}_{an}$ with weights $w_i = \frac{1}{\pi_i}$, $i = 0, \ldots, N-1$ satisfies*

$$\|\nu P^n - \nu \bar{P}_{an}\|_w \leq \frac{|\lambda_q|^n}{\sqrt{\min\limits_{0 \leq i \leq N-1} \pi_i}}$$

where q is the number of dominant eigenvalues of $M(P^n)$ (and P^n) and λ_q is the eigenvalue of P such that $|\lambda_q^n|^2 = \beta_q^n$ where β_q^n is the $q+1$ largest eigenvalue of $M(P^n)$.

It can be seen from Theorem 12 that, for the original non-reversible system, the distance between π_n^ν and π is bounded from above by $O((\beta_1(M))^{n/2})$, for any initial distribution ν. If β_1 is small, the convergence of the distribution of the Markov chain at time n to the stationary distribution π is fast and the stationary distribution π is a good low dimensional approximate model. Thus we have $\pi_{an}^\nu = \pi$. On the other hand, if $M(P)$ has several eigenvalues close to one then the convergence is not as fast. In this case we want to construct an approximate operator π_{an}^ν to π_n^ν with the distance between π_n^ν and π_{an}^ν converging to zero considerably faster than $(\beta_1(M))^{n/2}$.

Let $\psi_i^{(n)}$ and $\varphi_i^{(n)}$, $i = 0,\ldots,N-1$ denote the right and left eigenvectors of $M(P^n)$ respectively. Then it is shown in [38] that

$$M(P^n) = \sum_{k=0}^{N-1} \beta_k^n \psi_k^n (\varphi_k^n)^T$$

Assume that $\beta_0,\ldots\beta_{q-1}$ are of comparable size (close to one) and $\beta_q \ll \beta_{q-1}$, i.e., q is the number of dominant eigenvalues of $M(P^n)$. Note here the number of dominant eigenvalues of $M(P^n)$ is the same as that of P^n, which is known equal to the number of the metastable sets of the original system. If the approximate reversibilized model is chosen as

$$M_a(P^n) = \sum_{k=0}^{q-1} \beta_k^n \psi_k^n (\varphi_k^n)^T$$

it is easy to see that $\|M(P^n) - M_a(P^n)\|$ is bounded above by $O(\beta_q^n)$, which means that the approximated reversibilized model at time n converges to the reversibilized model of the original non-reversible P^n system fast. Based on such an approximate model $M_a(P^n)$, we want to recover a good approximate operator $\pi_{an}^\nu = \nu\bar{P}_{an}$ to $\pi_n^\nu = \nu P^n$, with $\bar{P}_{an}$ satisfying $M_a(P^n) = \bar{P}_{an}D^{-1}\bar{P}_{an}^T D$. If we let

$$\bar{P}_{an} = D^{-1/2} V J_a V^T D^{1/2} P^n$$

where $J_a = diag(I_q, 0)$, $V = D^{\frac{1}{2}}[\psi_1^{(n)}, \ldots, \psi_{N-1}^{(n)}]$, then $\bar{P}_{an}$ can be shown as a good approximation to P^n (see [38] for details).

Modal dynamics detection and partition estimation

Based on the approximate operator $\bar{P}_{an}$, a new Markov chain on the lower q dimensional space that characterizes the modal behavior of the system can be constructed.

Recall that the distribution of the system state at time n starting form the initial distribution ν is $\pi_n^\nu = \nu P^n$ and the corresponding approximate operator is $\pi_{an}^\nu = \nu P_{an}$. For two initial distributions ν and μ define the weighted L_2 distance (so-called diffusion distance in the language of [35]) as

$$D_n^2(\mu, \nu) = \|\pi_n^\mu - \pi_n^\nu\|_w^2 = \sum_{i=0}^{N-1} \frac{\left(\left(\pi_n^\mu\right)_i - \left(\pi_n^\nu\right)_i\right)^2}{\pi_i}$$

Let ν_i be the initial distribution concentrated at state x^i. Then if X partitions into q disjoint components $A_1, \ldots, A_q$ that the system dynamics cluster on (i.e. metastable components), two initial states x^i and x^j will belong to the same cluster (metastable component) if $D_n^2(\nu_i, \nu_j)$ is small and will belong to different clusters if $D_n^2(\nu_i, \nu_j)$ is large. We remark that in terms of the approximate operator, $D_n^2(\mu, \nu) \approx D_{an}^2(\mu, \nu) = \left\|\pi_{an}^\mu - \pi_{an}^\nu\right\|_w^2$. Furthermore, using $\pi_n^\nu = \nu P^n$ we have

$$\begin{aligned} D_n^2(\nu_i, \nu_j) &= \left\|\pi_n^{\nu_i} - \pi_n^{\nu_j}\right\|_w^2 \\ &= \left(\nu_i - \nu_j\right) P^n D^{-1} \left(P^T\right)^n \left(\nu_i - \nu_j\right) \\ &= \left(\nu_i - \nu_j\right) M(P^n) D^{-1} \left(\nu_i - \nu_j\right) \\ &= \left(\nu_i - \nu_j\right) \sum_{k=0}^{N-1} \beta_k^n \psi_k^n (\psi_k^n)^T \left(\nu_i - \nu_j\right) \end{aligned}$$

Note that $\nu_i = e_i^T$ where e_i is the i^{th} unit vector in $\mathbb{R}^N$. Therefore $\nu_i \psi_k^n = e_i^T \psi_k^n = \psi_k^n(i)$ and

with $\Psi_n(v_i) = \left(\sqrt{\beta_0^n}\psi_0^n(i), \ldots, \sqrt{\beta_{N-1}^n}\psi_{N-1}^n(i)\right)^T$ we have

$$D_n^2(v_i, v_j) = \left\|\Psi_n(v_i) - \Psi_n(v_j)\right\|^2$$

where $\|\bullet\|$ is the Euclidean norm on $\mathbb{R}^N$. Furthermore,

$$D_n^2(v_i, v_j) \approx D_{an}^2(\mu, \nu) = \left\|\Psi_{an}(v_i) \ - \Psi_{an}(v_j)\right\|$$

where $\Psi_{an}(v_i) = \left(\sqrt{\beta_0^n}\psi_0^n(i), \ldots, \sqrt{\beta_{q-1}^n}\psi_{q-1}^n(i)\right)^T \in \mathbb{R}^q$. We select a threshold value $0 < \delta \ll 1$ and classify two initial states to belong to the same cluster if $D_{an}^2(v_i, v_j) < \delta$ and different clusters if $D_{an}^2(v_i, v_j) > \delta$ (we note that the value δ may have to be adjusted to identify q clustering components) This procedure results in the desired partition of X into q disjoint metastable components $A_1, \ldots, A_q$ that characterize the modal behavior of the system. Obviously, the transition matrix P_q that describes the transition dynamics of the system between these components is $P_q(i, j) = p(A_j|A_i)$.

Remark 1 *We note that Ψ_n defines a nonlinear map of the original data and in terms of the Ψ_n the diffusion distance is a simple Euclidean distance. We remark the similarity with the KPCA method in Section 2.7 with the noticeable difference that here the map Ψ_n is defined by the system itself.*

3.1.4 Example

In this section we present a complete identification procedure for a system that has two strong attraction domains.

Example 3 *Consider a nonlinear discrete time stochastic dynamic system on $\mathbb{R}^2$ described*

by the equations

$$\begin{aligned} x_{t+1} &= f(x_t) + \varepsilon_1 w_t \\ y_{t+1} &= x_t + \varepsilon_2 w_t \end{aligned}$$

where $x = \begin{bmatrix} x^1 & x^2 \end{bmatrix}^T$, $y = \begin{bmatrix} y^1 & y^2 \end{bmatrix}^T$, $w = \begin{bmatrix} w^1 & w^2 \end{bmatrix}^T$ *is a sequence of i.i.d. standard Gaussian random variables, and*

$$f(x) = \begin{bmatrix} x^1 + \delta t x^2 \\ x^2 + \delta t(-x^2 + \alpha(\beta x^1 - (x^1)^3)) \end{bmatrix}$$

Here we choose $\alpha = \beta = 1$, $\delta t = 0.2$, $\varepsilon_1 = \varepsilon_2 = 0.1$. *The system has three equilibrium points, an unstable one at the origin and stable equilibria at* $\begin{bmatrix} \pm\sqrt{\beta} & 0 \end{bmatrix}^T$. *These two stable equilibria correspond to the two modes of the system.*

A simulation of the stochastic system for a typical initial condition is shown in Figure 2. For simplicity the output space is divided into 6 pieces such that output space contains 6 alphabets. The same division is adopted for the state space, based on the assumption that the output is the state perturbed by a small additive noise. Given the observation series, we estimate the state transition matrix system using the identification approach presented in Section 3.1 to be

$$P_e = \begin{bmatrix} 0.74 & 0.26 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.17 & 0.78 & 0.05 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.18 & 0.75 & 0.07 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.07 & 0.69 & 0.24 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.08 & 0.74 & 0.18 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.25 & 0.75 \end{bmatrix}$$

which is close to the transition matrix calculated from the true state sequence,

$$P = \begin{bmatrix} 0.82 & 0.18 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.12 & 0.83 & 0.05 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.15 & 0.79 & 0.05 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.06 & 0.78 & 0.16 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.05 & 0.83 & 0.12 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.18 & 0.82 \end{bmatrix}$$

The $\bar{x}$ to $\bar{y}$ transition matrix is estimated to be:

$$B_e = \begin{bmatrix} 0.74 & 0.26 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.17 & 0.72 & 0.10 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.18 & 0.68 & 0.14 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.07 & 0.68 & 0.25 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.08 & 0.74 & 0.18 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.25 & 0.75 \end{bmatrix}$$

Since y is equal to x with small additive Gaussian noise, the B matrix is expected be close to identity. We note that for any $i \in \{1,2,...,6\}$, $\Pr(\bar{y} = i \,|\, \bar{x} = i)$ is dominant, as expected. It has been detected that the system dynamics have 2 modes and the identified modal transition matrix is

$$P_2 = \begin{bmatrix} 0.9586 & 0.0413 \\ 0.0412 & 0.9587 \end{bmatrix}$$

Furthermore, the states in state space divide into two groups. These two groups are marked in Figure 2 with different colors. The division of the state space into the two groups fits the dynamic system's two real attractors very well.

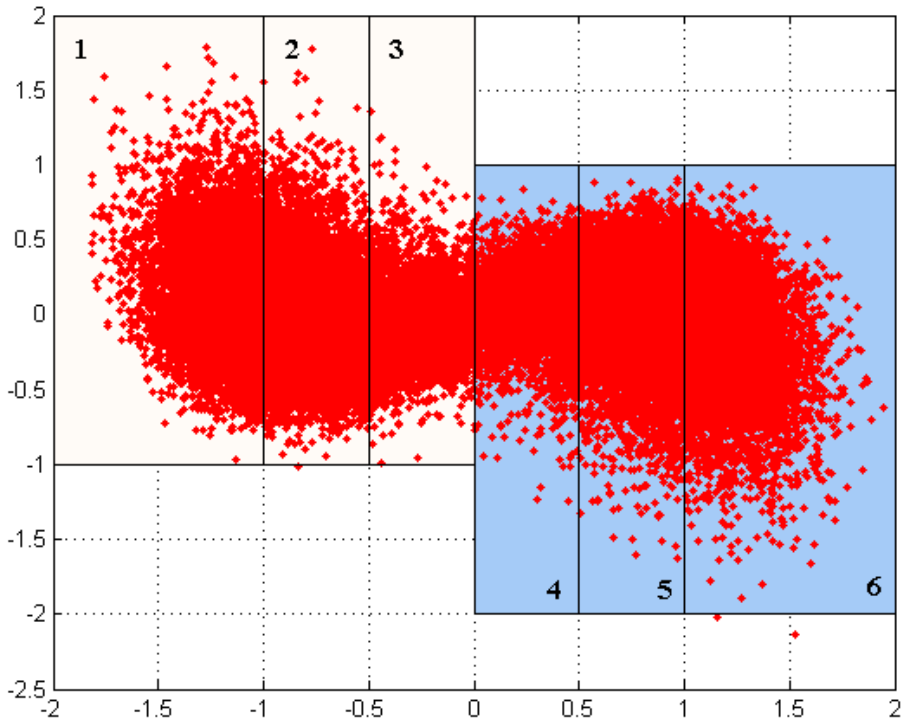


Figure 2 Output Trajectories

3.1.5 Conclusion

In this section we proposed an approach for the modal behavior identification of SAFHS equivalent complex dynamic system. It is demonstrated in an example that the state transition matrix, state-output matrix, system modes, state space partition regions as well as the modal transition matrix are accurately estimated, which demonstrates the effectiveness of the proposed method.

With the multi-modal behavior for the hybrid system being identified successfully, what remains to complete the identification of a SAFHS model is to characterize local dynamics, which corresponds to the identification of local dynamic model in each of the clustering components. We discuss the local dynamics identification next.

3.2 Identification of Local Dynamics

Assume we have completed the identification procedure in the previous section and have the partition $A_1, \ldots, A_q$ and the identified state sequence $\hat{x}_1, \hat{x}_2, \ldots$. We can map the state sequence onto the partition and through the association between the state and output strings we classify the output symbols (and substrings) into q partition components (clusters) as well, say $Y_1, \ldots, Y_q$. Note that some of the output partition components may be identical, i.e. two or more components of the state partition could be mapped into the same output components. This results in a redundancy that can be reduced after the identification procedure is complete.

Now given a sufficiently long substring, say $y_{i:i+k} \in Y_j$, of the original measured output string (not the discretized string), we can employ conventional identification techniques for the identification of a submodel on component Y_j. If a preliminary analysis of the data string $y_{i:i+k}$ indicates that the data clusters around a single point we may assume that Y_j contains a stable "equilibrium point" and use linear identification techniques. On the other hand, if the preliminary analysis of the data indicates periodic motion or some other nonlinear type behavior we have to employ nonlinear identification techniques. Since linear identification methods have been extensively studied during the last several decades we refer to the system identification toolbox from Matlab for those techniques and concentrate on the much less studied nonlinear case here.

3.2.1 Nonlinear Identification Techniques

Nonlinear local dynamic identification is difficult because of the variety of nonlinear behaviors that can be expected. In our current work, we have mainly concentrated on the identification for a nonlinear periodic behavior, i.e. limit cycle. The development of the identification techniques for the other nonlinear type behaviors will be considered in our future work.

Limit Cycle Behavior Identification

Identification of limit cycle behavior is considered to be a fundamental problem in many real systems and applications. In the following we give a description of limit cycle behavior models as well as propose an identification algorithm for that model.

System Models Many nonlinear systems, e.g. tunnel diodes, pendulums, biological predator–prey systems and frequency synthesizers [26], that generate periodic signals can be described by second-order nonlinear ordinary differential equations (ODEs) with polynomial right hand side [59], [58], [46], [1].

In [59], it has been shown that the second order ODE

$$\ddot{y}(t) = \tilde{f}(\dot{y}(t), y(t), \tilde{\theta})$$

accurately represents a large class of periodic signals. We assume that $y(t)$ is our measured signal and choose state variables as

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} y(t) \\ \dot{y}(t) \end{pmatrix}$$

Then we have in state space form

$$\begin{aligned} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} &= \begin{pmatrix} x_2(t) \\ \tilde{f}(x_1(t), x_2(t), \tilde{\theta}) \end{pmatrix} \\ y(t) &= \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \end{aligned} \tag{3.6}$$

Note that the states are deferent from those defined in Section 3.1. In particular, the states defined here represent the local dynamics corresponding to the output data string y within

Y_j. As we mentioned earlier, it has been shown that the right-hand-side of the state space function can be represented accurately by a multivariable polynomial which has an unknown parameter $\tilde{\theta}$ to be identified.

In real applications, since the measurements are usually in discrete time, we study the discretized form of (3.6). The discretization can be developed using any of a number of well known procedures which we omit here. The resulting discretized system is

$$\begin{aligned} \begin{pmatrix} x_1(k+1) \\ x_1(k+2) \end{pmatrix} &= \begin{pmatrix} x_2(k)T + x_1(k) \\ f(x_1(k), x_1(k+1), \hat{\theta}) \end{pmatrix} \\ y(k) &= \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} \end{aligned} \tag{3.7}$$

where T denotes the sampling interval. As the states here correspond to the output data string y within Y_j, if we have the polynomial function f identified, we have equivalently figured out the regression for y .

Remark 2 *Obviously the above discussion assumes that the output $y(t)$ is scalar valued. The multivariable case will be considered in our future work.*

Algorithm Development In the past decade, kernel methods have attracted much attention in regression. Moreover, using kernel functions, many linear methods can be extended to the nonlinear case almost straightforwardly, e.g., the KPCA extended from PCA. When KPCA is applied to extract features or applied to regression, all kernel functions are calculated based on the current sample and the feature vectors which are calculated based on all the training samples. Accordingly, as the size of the training sample set increases, the computational complexity increases rapidly.

Here we develop a reduced KPCA algorithm where the feature vectors are calculated from the considerably reduced number of training data, which will in turn reduce the computational complexity.

Suppose we want to identify the function $f(x)$. A reduced KPCA is used as the first step to detect the feature vectors $\{x_{q1}, x_{q2}, \cdots, x_{qq}\}$ from the training set $\{x_1, x_2, \cdots, x_M\}$, where $x_i \in \mathbb{R}^L$, $q \ll M$. The local dynamics of interest can then be expressed as

$$f(x) = \sum_{l=1}^{q} \theta_l k(x, x_{ql}) + b$$

Since for any x, $k(x, x_{ql})$, $l \in \{1, \cdots, q\}$ can be calculated based on the pre-chosen kernel function $k(\cdot, \cdot)$, the only unknown parameters are θ_l and b. Note as we have noted before, $f(x)$ has polynomial form so we consider a polynomial kernel here. The second step is to estimate the unknown parameters through minimum least square method. In the following we focus on introducing the first identification step since the least square estimation in the second step is a well developed algorithm.

Reduced Kernel PCA We have introduced the idea of KPCA in Section 2.7.1 and mentioned that in KPCA, all the training data are used as the feature vectors. In the following we introduce a reduced KPCA algorithm which reduces the number of feature vectors which are chosen from the training data sets.

Given a training set $\{x_1, x_2, \cdots, x_M\}$ with size M, we regularize the training set by multiplying vector x_i by a scalar so that $\forall i$, $|x_i| \leq 1$. Suppose M is large enough so that $\forall x \in \mathbb{R}^L$, there exists $\{\beta_1, \beta_2, \cdots, \beta_M\}$ s.t. $\Phi(x) = \sum_{i=1}^{M} \beta_i \Phi(x_i)$, where $\Phi : \mathbb{R}^L \to F$ is the function that maps the data to the feature space with $\left\langle \Phi(x_i), \Phi(x_j) \right\rangle = k(x_i, x_j)$ and $F \subset \mathbb{R}^H$ is the range of Φ. We remark that Φ is injective and for a polynomial kernel H is finite.

Construct the matrix

$$
\begin{aligned}
C_{M\times M} &= \begin{bmatrix} \Phi^T(x_1)\Phi(x_1) & \Phi^T(x_1)\Phi(x_2) & \cdots & \Phi^T(x_1)\Phi(x_M) \\ \vdots & & \ddots & \\ & & & \\ \Phi^T(x_M)\Phi(x_1) & & \cdots & \Phi^T(x_M)\Phi(x_M) \end{bmatrix} \\
&\triangleq \begin{bmatrix} c_1 & c_2 & \cdots & c_M \end{bmatrix}
\end{aligned}
$$

Since the C matrix is symmetric, it has M orthonormal eigenvectors $\{v_1, v_2, \cdots, v_M\}$, $v_i \in \mathbb{R}^M$ corresponding to M real eigenvalues assigned in decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$. Then by executing PCA on the C matrix, we can get a set of orthonormal eigenvectors $\{v_1, v_2, \cdots, v_q\}$, $v_i \in \mathbb{R}^M$ corresponding to the dominant eigenvalues, $\lambda_1, \lambda_2, \cdots \lambda_q$ of C. Construct a matrix $\tilde{C}$ as

$$
\begin{aligned}
\tilde{C} &= \begin{bmatrix} v_1 & v_2 & \cdots & v_q \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_q \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_q \end{bmatrix}^T \\
&\triangleq \begin{bmatrix} \tilde{c}_1 & \tilde{c}_2 & \cdots & \tilde{c}_M \end{bmatrix}
\end{aligned}
$$

Note that $\left\|C - \tilde{C}\right\| = \lambda_{q+1}$, where λ_{q+1} is the $q+1^{th}$ largest eigenvalue of C matrix. By choice of q, λ_{q+1} is generally very small. Moreover, we have

$$\tilde{C}v_i = \lambda_i v_i, \forall i = 1, ...q$$

thus

$$v_i = \frac{1}{\lambda_i}\tilde{C}v_i = \sum_{j=1}^{M} \eta_j \tilde{c}_j, \forall i = 1, ...q$$

that is, v_i can be represented as a linear combination of $\{\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_M\}$. It can also be

shown that $R(\tilde{C})$, the range of $\tilde{C}$, is spanned by v_i, $i = 1,...q$. Accordingly, given q linear independent vectors $\{\tilde{c}_{q1}, \tilde{c}_{q2}, \cdots, \tilde{c}_{qq}\}$, $\tilde{c}_{qi} \in R(\tilde{C})$, we can express v_i as

$$v_i = \sum_{j=1}^{q} \tilde{\eta}_{i,j} \tilde{c}_{qj}, \forall i = 1,...q$$

Here we give the algorithm to select q linear independent vectors $\{\tilde{c}_{q1}, \tilde{c}_{q2}, \cdots, \tilde{c}_{qq}\}$ from $\{\tilde{c}_1, \tilde{c}_2, \cdots, \tilde{c}_M\}$

- *let* $l_i = \|\tilde{c}_i\|_2$, $i \in \{1,...M\}$, $L = \{l_1, \cdots, l_M\}$, $C_{temp} = \{\tilde{c}_1, \cdots, \tilde{c}_M\}$;
- $n = 1$, $\tilde{c}_{q1} = \{\tilde{c}_i : \varepsilon_i = \max\{L\}\}$;
- *remove* $\tilde{c}_i$ *from* C_{temp};
- *when* $n \leq q$
  - *do until* $\tilde{c}_{qn}$ *is independent to* $\tilde{c}_{q1}, \cdots \tilde{c}_{q(n-1)}$ *the angle between* $\tilde{c}_{qn}$ *and the space spanned by* $\tilde{c}_{q1}, \cdots \tilde{c}_{q(n-1)}$ *is bigger than a preset value* ϖ.

    $\tilde{c}_{qn} = \{\tilde{c}_i : l_i = \max\{L\}\}$;

    remove $\tilde{c}_i$ *from* C_{temp};
  - *end do*
  - $n = n + 1$;
- *end when*
- *end*

Using this algorithm we get q linear independent vectors $\{\tilde{c}_{q1}, \cdots, \tilde{c}_{qq}\}$ which are close to the vectors $\{c_{q1}, \cdots, c_{qq}\}$ because $\left\|C - \tilde{C}\right\| = \lambda_{q+1}$. Note

$$\begin{aligned} c_{qi} &= \begin{bmatrix} \Phi^T(x_{qi})\Phi(x_1) & \Phi^T(x_{qi})\Phi(x_2) & \cdots & \Phi^T(x_{qi})\Phi(x_M) \end{bmatrix}^T \\ &= \begin{bmatrix} k(x_{qi}, x_1) & k(x_{qi}, x_2) & \cdots & k(x_{qi}, x_M) \end{bmatrix}^T \end{aligned}$$

where $k(\cdot,\cdot)$ is the kernel function. Since c_{qi} is close to $\tilde{c}_{qi}$ and $k(\cdot,\cdot)$ is continuous, we approximate

$$\begin{aligned} \tilde{c}_{qi} &\approx \begin{bmatrix} k(x_{qi},x_1) & k(x_{qi},x_2) & \cdots & k(x_{qi},x_M) \end{bmatrix}^T \\ &= \begin{bmatrix} \Phi^T(x_{qi})\Phi(x_1) & \Phi^T(x_{qi})\Phi(x_2) & \cdots & \Phi^T(x_{qi})\Phi(x_M) \end{bmatrix}^T \end{aligned} \tag{3.8}$$

Proposition 13 *$f(x)$ can be approximated by the linear combination of $k(x,x_{qi})$, $i \in \{1,\cdots,q\}$ in the form $f(x) \approx \sum_{l=1}^{q} \theta_l k(x,x_{qi}) + b$, where the error bound can be shown to be less or equal to $O(\lambda_{q+1})$, that is, x_{qi}, $i \in \{1,\cdots,q\}$ can be treated as the feature vectors.*

Proof. We start with the expression of $f(x)$ based on the feature vectors that contain all the training data, we have

$$\begin{aligned} f(x) &= \textstyle\sum_{i=1}^{M} \alpha_i k(x,x_i) + b \\ &= \textstyle\sum_{i=1}^{M} \alpha_i \Phi^T(x_i)\Phi(x) + b \\ &= \textstyle\sum_{i=1}^{M} \alpha_i \sum_{j=1}^{M} \beta_j \Phi^T(x_i)\Phi(x_j) + b \\ &= \textstyle\sum_{i=1}^{M} \alpha_i \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} \\ &\quad \times \begin{bmatrix} \Phi^T(x_i)\Phi(x_1) & \Phi^T(x_i)\Phi(x_2) & \cdots & \Phi^T(x_i)\Phi(x_M) \end{bmatrix}^T + b \\ &\approx \textstyle\sum_{i=1}^{M} \alpha_i \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} \sum_{l=1}^{q} \gamma_{i,l} v_l + b \\ &= \textstyle\sum_{i=1}^{M} \alpha_i \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} \sum_{l=1}^{q} \gamma_{i,l} \sum_{j=1}^{q} \tilde{\eta}_{l,j} \tilde{c}_{qj} + b \end{aligned} \tag{3.9}$$

Now, substituting (3.8) into (3.9) gives

$$\begin{aligned}
f(x) &\approx \sum_{i=1}^{M} \alpha_i \left[\begin{array}{cccc} \beta_1 & \beta_2 & \cdots & \beta_M \end{array} \right] \sum_{l=1}^{q} \gamma_{i,l} \\
&\quad \times \sum_{j=1}^{q} \tilde{\eta}_{l,j} \left[\begin{array}{cccc} \Phi^T(x_{qj})\Phi(x_1) & \Phi^T(x_{qj})\Phi(x_2) & \cdots & \Phi^T(x_{qj})\Phi(x_M) \end{array} \right]^T + b \\
&= \sum_{i=1}^{M} \alpha_i (\left[\begin{array}{cccc} \beta_1 & \beta_2 & \cdots & \beta_M \end{array} \right] \sum_{l=1}^{q} \gamma_{i,l} \\
&\quad \times \sum_{j=1}^{q} \tilde{\eta}_{l,j} \left[\begin{array}{cccc} \Phi(x_1) & \Phi(x_2) & \cdots & \Phi(x_M) \end{array} \right]^T \Phi(x_{qj})) + b \\
&= \sum_{i=1}^{M} \alpha_i (\sum_{l=1}^{q} \gamma_{i,l} \sum_{j=1}^{q} \tilde{\eta}_{l,j} (\sum_{t=1}^{M} \beta_t \Phi^T(x_t)) \Phi(x_{qj})) + b \\
&\approx \sum_{i=1}^{M} \alpha_i (\sum_{l=1}^{q} \gamma_{i,l} \sum_{j=1}^{q} \tilde{\eta}_{l,j} \Phi^T(x) \Phi(x_{qj})) + b \\
&= \sum_{l=1}^{q} \theta_l \Phi^T(x) \Phi(x_{qj}) + b
\end{aligned}$$

Note that the following approximation has been made in (3.9)

$$\begin{aligned}
&\sum_{i=1}^{M} \alpha_i \left[\begin{array}{cccc} \beta_1 & \beta_2 & \cdots & \beta_M \end{array} \right] \\
&\qquad \times \left[\begin{array}{cccc} \Phi^T(x_i)\Phi(x_1) & \Phi^T(x_i)\Phi(x_2) & \cdots & \Phi^T(x_i)\Phi(x_M) \end{array} \right]^T + b \\
\approx\ &\sum_{i=1}^{M} \alpha_i \left[\begin{array}{cccc} \beta_1 & \beta_2 & \cdots & \beta_M \end{array} \right] \sum_{l=1}^{q} \gamma_{i,l} v_l + b
\end{aligned}$$

We now calculate the error bound for this approximation. Note that

$$
\begin{aligned}
& \textstyle\sum_{i=1}^{M} \alpha_i \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} \\
& \qquad \times \begin{bmatrix} \Phi^T(x_i)\Phi(x_1) & \Phi^T(x_i)\Phi(x_2) & \cdots & \Phi^T(x_i)\Phi(x_M) \end{bmatrix}^T + b \\
= & \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} \textstyle\sum_{i=1}^{M} \alpha_i c_i + b \\
= & \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} \textstyle\sum_{i=1}^{M} \alpha_i \sum_{l=1}^{M} c_i^T v_l v_l + b \\
= & \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} \textstyle\sum_{i=1}^{M} \alpha_i \sum_{l=1}^{M} \lambda_l v_l^i v_l + b \\
= & \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} \textstyle\sum_{i=1}^{M} \alpha_i \begin{bmatrix} v_1 & v_2 & \cdots & v_M \end{bmatrix} \begin{bmatrix} \lambda_1 v_1^i & \lambda_2 v_M^i & \cdots & \lambda_M v_M^i \end{bmatrix}^T \\
= & \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_M \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_M \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_M^T \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_M \end{bmatrix} \\
= & \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} C \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_M \end{bmatrix}
\end{aligned}
$$

We remark that in the above $\lambda_j v_j^i$ is the $\gamma_{i,j}$ in (3.9). Now using the $\{v_1, v_2, \cdots, v_q\}$ basis is equivalent to approximate the above equation by

$$
\begin{aligned}
& \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_M \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \lambda_q & \\ & & & \mathbf{0} \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_M^T \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_M \end{bmatrix} \\
= & \textstyle\sum_{i=1}^{M} \alpha_i \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_M \end{bmatrix} \textstyle\sum_{l=1}^{q} \gamma_{i,l} v_l + b
\end{aligned}
$$

It is easy to show that the error bound of the approximation is less or equal to $\lambda_{q+1} \|\alpha\| \|\beta\| = O(\lambda_{q+1})$.

In summary we have shown that $f(x) \approx \sum_{i=1}^{M} \alpha_i k(x, x_{qi}) + b$, $\forall x \in \mathbb{R}^L$, which in turn shows that x_{qi}, $i \in \{1, \cdots, q\}$ can be treated as feature vectors. ■

As we mentioned before, once we identify the feature vectors $\{x_{q1}, x_{q2}, \cdots, x_{qq}\}$, the unknown parameters θ_l and b can be easily found by the well known minimum least square method, which completes the identification of limit cycle system.

3.2.2 Linear Identification Techniques

If based on some preliminary analysis of the data the attractor is detected to have an "equilibrium point", then conventional linear identification techniques can be utilized to approximate the local dynamics. As we mentioned previously, there exists a large body of well developed linear identification techniques that have been developed into commercial codes. In this dissertation, we use the system identification toolbox from Matlab for the identification of linear local dynamics. We will not describe the application of linear identification methods in detail and refer the reader to the system identification toolbox in Matlab for further details.

3.2.3 Example

Now we illustrate the performance of techniques discussed above through examples for the identification of both linear and limit cycle local dynamics cases.

Strong Attraction Point Case (linear case)

Consider the example in Section 3.1.4, where the state space of the system has been decoupled into two clusters. We map the output data sequences into the two identified (output) groups and analyze the output data within each group. It can be easily found that the data clustered around a point in each group and therefore a linear type system model is

appropriate.

Since the output is two dimensional we consider a state space model of the affine form

$$\begin{aligned} x(k+1) &= A(r)x(k)+b(r)+K(r)w(k) \\ y(k+1) &= C(r)x(k+1)+w(k) \end{aligned}$$

where $b(r)$ is a bias vector, $w(k)$ is a standard Gaussian noise and r denotes the state of the modal transition dynamics taking value in $\{1,2\}$. The system identification toolbox in Matlab is used in the identification of each of the two local affine models. It is found that a system with state dimension $n = 8$ fits the data nicely. Figure 3 gives the comparison between a realization of the identified system and the original output series. As is clearly seen from Figure 3 the estimated system matches the original system very nicely.

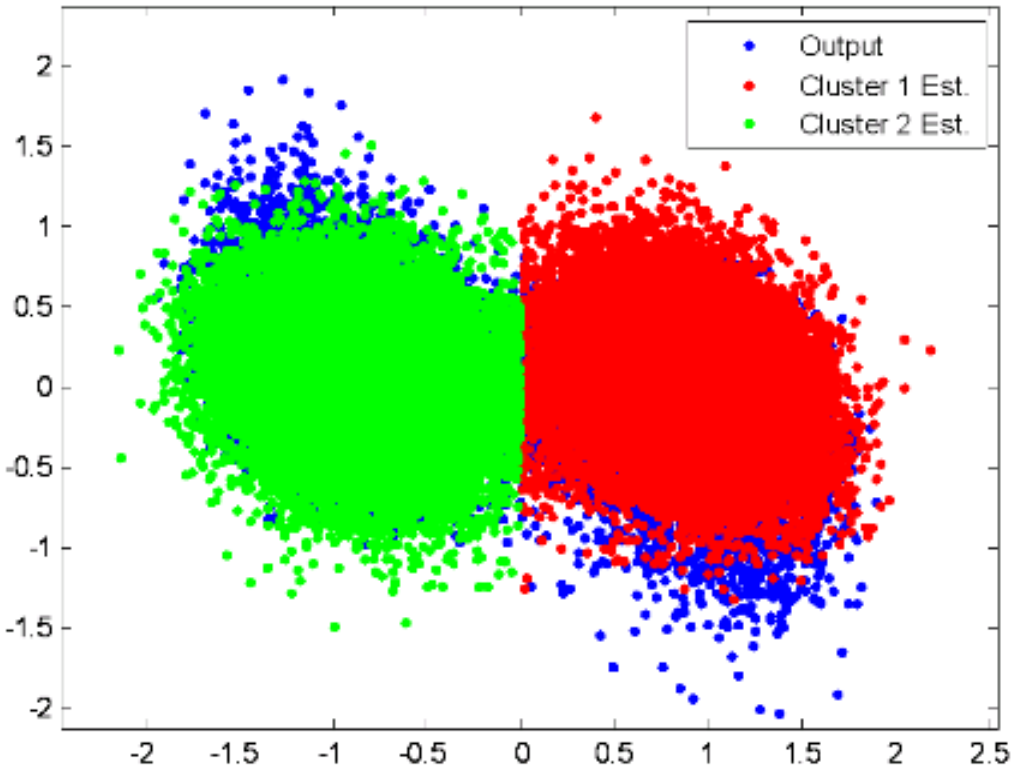


Figure 3. Comparison of original and simulated system dynamics.

Limit Cycle Case

In this example the identification of Van Der Pol system is considered. The Van Der Pol system is a limit cycle system given by (in continuous form)

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ 0.5 \times (1 - x_1^2) \times x_2 - x_1 \end{pmatrix}$$

The Matlab routine ode45 was used to solve the equation. The sampling interval was chosen as $T_S = 0.1s$ and the initial state was selected as $(x_1(0)\ x_2(0))^T = (1.5\ 1)^T$. An observation noise was added to the measured signal,

$$y(k) = x_1(k) + w(k)$$

where $w(k)$ is Gaussian noise with $\sigma = 0.002$ (here $y(k) = y(kT_S)$). We note that in practice we could only observe the process at discrete times.

Next we model the discrete system corresponding to the original Van Der Pol system based on the noisy observation $y(k)$. The proposed reduced KPCA is used to identify the system model. The kernel function is selected as a 3^{rd} order polynomial kernel. Two hundred data points have been used as training set in the kernel method and 18 feature vectors chosen from the training set have been used for system identification. Figure 4 and Figure 5 show the true and estimated phase plane plots and time series for the system considered. We can see, either from the phase plane plot or from the time series graph, that

the periodic dynamics of the Van Der Pol system can be identified very accurately.

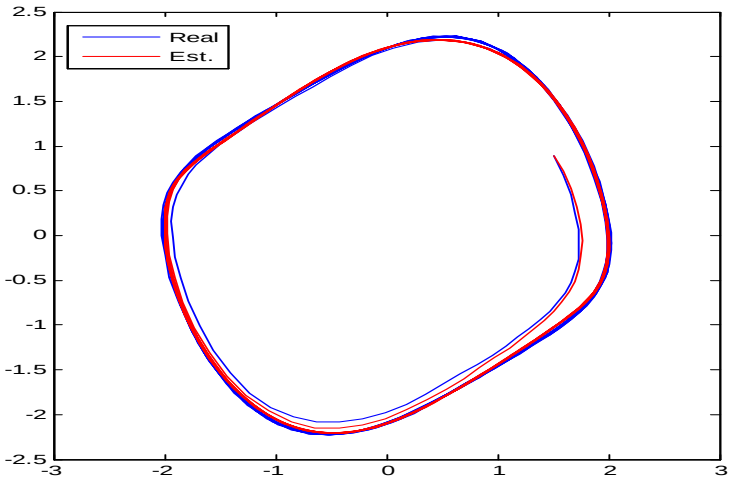


Figure 4. Evaluation of the proposed local dynamics identification algorithm–Phase plane

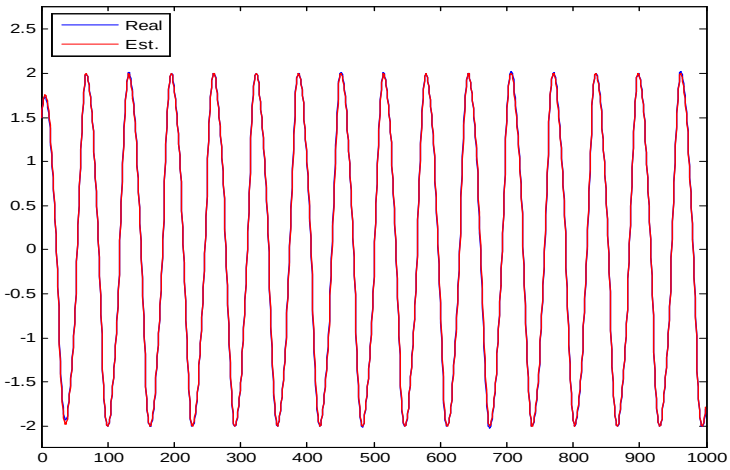


Figure 5. Evaluation of the proposed local dynamics identification algorithm–Time series

3.2.4 Conclusion

In this subsection we assume that a sufficiently long substring, say $y_{i:i+k} \in Y_j$, of the original measured output string is given. We described how the local dynamics, both the nonlinear case and the linear case can be identified. For the linear case, we adopt conventional linear identification techniques and utilized the Matlab system identification toolbox for this purpose and have shown through an example that the estimated system matches the original system very nicely. For the nonlinear limit cycle case, we proposed a reduced KPCA based identification algorithm, which has also been shown through a Van Der Pol system example that it can identify the dynamic model accurately.

3.3 Conclusion

In this Chapter the identification of a complex dynamic system that exhibits multi-modal behavior is proposed based on SAFHS modeling. The developed approach divides the underlying space of the system of interest in terms of strong attraction domains which usually have nonlinear boundary, and then detects the local dynamics on the corresponding divided output space. It is demonstrated in an example that the state transition matrix, state-output association, system modes, state space partition regions, output partition region as well as the modal transition matrix are accurately estimated, which illustrates the effectiveness of the proposed multi-modal behavior identification method. Furthermore, it is demonstrated that, for some equilibrium attractor based systems, conventional identification techniques can be successfully used for the identification of the local dynamic behavior within each clustering component, while for the nonlinear local dynamics, e.g. the limit cycle case, the reduced KPCA based identification algorithm shows good performance.

CHAPTER 4

Estimation of Hybrid Systems

In this Chapter we present a new high accuracy and reduced computational load particle filtering based algorithm for stochastic hybrid system state estimation and mode detection.

4.1 Algorithm Development

The estimation algorithm we present here combines IMMPF and OTPF filters (to be described in detail below) to estimate linear/nonlinear hybrid system mode and state. The proposed algorithm is called reduced multiple model particle filter (RMMPF). To facilitate the development we organize this section into three parts: 1) IMMPF [5]; 2) OTPF [49]; 3) RMMPF.

4.1.1 IMMPF

As discussed in Section 2.8 the IMM estimation scheme can be generalized into three steps. We now describe the interacting multiple model particle filtering (IMMPF) algorithm developed in [5] in terms of these three steps.

Consider again the hybrid system in (1.2). Suppose that at time $k-1$ there are N samples in total for the M modes and each mode m has sample set S_m of size N/M (for simplicity we assume that N is divisible by M). We denote the state and its associated weight by $\{x_{k-1}^{i,m}, \omega_{k-1}^{i,m}\}$, $i \in \{1,2,\ldots N/M\}$, $m \in \{1,2,\ldots M\}$. In addition, mode m has posterior probability $p(r_{k-1} = m|y_{0:k-1})$ at time $k-1$. The Markov transition matrix for the modal process r_k is denoted by Π, with entries $\pi_{s,t}$, $s,t \in \{1,2,\ldots M\}$. One cycle of the IMMPF algorithm at time k is described as follows:

- Mixing/interaction step:

Mode switching:

$$p(r_k = m|y_{0:k-1}) \approx \Omega^*(r_k = m) = \sum_{s=1}^{M} \sum_{i=1}^{N/M} \Pi_{s,m} \omega_{k-1}^{i,s}$$

Interaction resampling:

$${}^*\omega_k^{i,m} = \Omega^*(r_k = m)M/N$$

If $\Omega^*(r_k = m) = 0$, then ${}^*x_{k-1}^{i,m} = x_{k-1}^{i,m}$, else:

$${}^*x_{k-1}^{i,m} \sim \sum_{s=1}^{M} \sum_{j=1}^{N/M} \Pi_{s,m} \omega_{k-1}^{j,s} \delta(x - x_{k-1}^{j,m})/\Omega^*(r_k = m)$$

where $i \in \{1,2,\ldots N/M\}, m \in \{1,2,\ldots M\}$

- Mode-conditioned state estimation step:

Prediction:

$$x_k^{i,m} = f_{r_k=m}({}^*x_{k-1}^{i,m}, v_k^{i,m})$$

Correction:

$$\omega_k^{i,m} =^* \omega_k^{i,m} p(y_k|x_k^{i,m}, r_k = m)$$

Then normalize $\omega_k^{i,m}$, $i \in \{1,2,\ldots N/M\}, m \in \{1,2,\ldots M\}$.

- Output step:

$$p(r_k = m|y_{0:k}) \approx \Omega(r_k = m) = \sum_{i=1}^{N/M} \omega_k^{i,m}$$

if $\Omega(r_k = m) > 0$:

$$p(x_k|y_{0:k}, r_k = m) = \sum_{i=1}^{N/M} \omega_k^{i,m} \delta(x - x_k^{i,m})/\Omega(r_k = m)$$

4.1.2 OTPF

The observation and transition-based most likely modes tracking particle filtering (OTPF) algorithm is proposed in [49] as a method for both fault detection (discrete state) and state estimation. In contrast to IMMPF, which takes all modes into account and assigns a fixed number of particles to each mode at each time step with a fixed number of particles in each mode evolving in time, OTPF chooses the most likely mode at each time step and only particles in this mode evolve into the next time step.

Next we present more details on how the OTPF algorithm works. Consider again the hybrid system in (1.2). Suppose at time $k-1$ the most likely mode is chosen to be $r_{k-1} = m$, there are N/M samples for mode $r_{k-1} = m$, denoted as $\{x_{k-1}^{i,m}, \omega_{k-1}^{i,m} = 1/(N/M)\}$, $i \in \{1,2,\ldots N/M\}$, where M is the number of modes. As before the modal Markov transition matrix is denoted as Π. One cycle of the OTPF algorithm at time k is described as follows:

- Interaction step:

For any mode $r_k = s, s \in \{1,2,\ldots M\}$ such that $\pi_{m,s}$ is not zero,

$$^*\omega_{k-1}^{i,s} = \pi_{m,s}\omega_{k-1}^{i,m}$$

$$^*x_{k-1}^{i,s} = x_{k-1}^{i,m}$$

- Mode-conditioned state estimation step:

Prediction:

$$^*x_k^{i,s} = f_{r_k=s}(^*x_{k-1}^{i,s}, v_k^{i,s})$$

Correction:

$$\omega_k^{i,s} =^* \omega_{k-1}^{i,s} p(y_k|x_k^{i,m}, r_k = s)$$

where $i \in \{1,2,\ldots N/M\}$.

- Mode selection step:

Mode weight of $r_k = s, s \in \{1, 2, \ldots M\}$ is:

$$\Omega(r_k = s) = \sum_{i=1}^{N/M} \omega_k^{i,s}$$

Find the most likely mode:

$$r_k = \underset{s \in \{1,2,\ldots M\}}{\arg\max} \{\Omega(r_k = s)\}$$

Normalize the weights of particles within the most likely mode r_k.

Distribution in the most likely mode r_k :

$$p(x_k | y_{0:k}, r_k) = \sum_{i=1}^{N/M} \omega_k^{i,r_k} \delta(x - x_k^{i,r_k})$$

here ω_k^{i,r_k} is the normalized weight of particles in r_k.

- Resampling step:

Resample within the most likely mode r_k at time k to get N/M new particles $\{x_k^{i,r_k=s}$, $\omega_k^{i,r_k=s} = 1/(N/M)\}$.

4.1.3 RMMPF

In the previous Sections we described in detail the IMMPF and OTPF algorithms, respectively. It has been shown in [5] [43] that the IMMPF algorithm developed in [5] has very good estimation performance. On the other hand, the OTPF algorithm is biased and is easily affected by observation outliers. Thus in the terms of estimation accuracy, IMMPF outperforms OTPF, but OTPF is simpler and requires less computational effort.

It is well known that when particle filters are used in real time estimation their high computational cost is of major concern. Therefore, the fact that OTPF outperforms IMMPF in

terms of computational effort is of notable importance. To further illustrate this we analyze the computational complexity for both IMMPF and OTPF following an approach similar to the one in [62]. Suppose there are M modes and recall that in IMMPF each mode has N/M samples. For one cycle in IMMPF the Mixing/interaction step for each mode requires about $O(N)$ operations to calculate the sample weights, additional $O((N/M)\log N)$ operations are necessary if the basic systematic resampling method is used; in the Mode-conditioned state estimation step $O(k_1 N/M)$ operations are needed to perform the prediction and $O(k_2 N/M)$ operations for the correction, where k_1 and k_2 are model dependent constants, independent of N and M. Finally, in the output step, estimation is achieved with about $O(N)$ operations for the whole system. Consequently, one cycle of the IMMPF algorithm requires approximately $(M+1)O(N) + MO((N/M)\log N) + M(O(k_1 N/M) + O(k_2 N/M))$ operations. On the other hand, for one cycle in OTPF no computation is necessary in the interaction step since $\omega_{k-1}^{i,m} \equiv 1/(N/M)$ and $\pi_{m,s}\omega_{k-1}^{i,m}$ are previously known; in the Mode-conditioned state estimation step, around $O(k_1 N/M) + O(k_2 N/M)$ operations are necessary to perform the prediction and correction in each mode; in the mode selection and estimation step, about $O(N)$ operations are needed for whole system; finally, the last step, resampling, can be performed with about $O((N/M)\log(N/M))$ operations. Thus approximately $O(N) + O((N/M)\log(N/M)) + M(O(k_1 N/M) + O(k_2 N/M))$ operations are required in the OTPF algorithm. For large N , the calculation load in resampling step dominates in both OTPF and IMMPF. It is easy to see that the computational load of the OTPF algorithm is much less than that of the IMMPF algorithm or on the order of $(M-1)O((N/M)\log(N/M))$, where $M << N$.

When applying IMMPF in hybrid system estimation, the weight of the true mode $\Omega(r_k = m)$ is dominant most of the time, i.e. if the system is in mode $r_k = m$ at time k then $\Omega(r_k = m) \approx 1$ while the weight of other modes is close to 0. In this case the difference between the performance of OTPF and IMMPF is small. In particular, in the IMMPF method, if the particles of one mode have dominant weights (which makes the weight for

this mode close to 1) then even if particles from the other modes are mixed with particles from the almost dominant mode, during the resampling in the mixing/interaction step of the IMMPF, most the particles from the non-dominant modes will be left out due to their very small weights. Consequently we can use the OTPF algorithm in this case to save computation without losing much accuracy. In other cases, which may happen after the system switches modes, or when there are observation outliers, the dominance of one mode will not be obvious. In order to avoid the evolution based on a wrong mode, which may occupy similar mode weight to the correct one in this case, it is better to take all possible modes into account rather than choose one mode based on a crude criteria as in OTPF. In this case selecting the IMMPF algorithm is the better choice to improve estimation accuracy.

Based on the above observations a new estimator integrating IMMPF and OTPF is proposed for the hybrid system (1.2). Initially a threshold $0 < \theta < 1$ is selected (typically θ is close to 1). This value is compared with $\max_{s\in\{1,2,\ldots M\}}\{\Omega(r_k = s)\}$ after the calculation at time step k. Note that $\Omega(r_k = s)$ denotes the probability for mode s at time k, thus $\sum_s \Omega(r_k = s) = 1$. If $\max_{s\in\{1,2,\ldots M\}}\{\Omega(r_k = s)\} > \theta$ it implies that at time k one mode is dominant, i.e. the mode with maximum mode weight is very likely the true mode. Thus the index factor, denoted by γ, is set to 1, which means that in the next time step only particles switched from the most likely mode at time k are considered, i.e. OTPF is selected in the next time step. If, on another hand, $\max_{s\in\{1,2,\ldots M\}}\{\Omega(r_k = s)\} < \theta$, it implies that at time k, no mode is obviously dominant, i.e. more than one mode is competitive to be the true mode, which happens when the system mode is switching or there are observation outliers. At this time, an index factor γ is set to 0, which means at the next time step particles switched from the all modes should be considered, i.e. IMMPF algorithm is selected at the next time step.

One cycle of the proposed RMMPF algorithm is described as follows:

if γ is 1:

- *the most likely mode* $r_{k-1} = m$ *at* $k-1$, *feed particles* $\{x_{k-1}^{i,m}, \omega_{k-1}^{i,m}\}$, $i \in \{1,2,\ldots N/M\}$,

$\sum_{i=1}^{N/M} \omega_{k-1}^{i,m} = 1$ *into OTPF;*

- *use the OTPF algorithm to calculate the mode weight* $\Omega(r_k = s), s \in \{1,2,\ldots M\}$ *and to estimate state at time k;*
- *if* $\max_{s\in\{1,2,\ldots M\}}\{\Omega(r_k = s)\} > \theta$ *and* $r_k = \arg\max_{s\in\{1,2,\ldots M\}}\{\Omega(r_k = s)\}$ *equal to* $r_{k-1} = m$

γ *= 1, keep only the particles in the most likely mode and normalize.*

else

γ *= 0, keep the particles in all modes*

end

else if γ is 0:

- *feed particles* $\{x_{k-1}^{i,m}, \omega_{k-1}^{i,m}\}$, $i \in \{1,2,\ldots N/M\}, m \in \{1,2,\ldots M\}$ *into IMMPF;*
- *use the IMMPF algorithm to calculate the mode weight* $\Omega(r_k = s), s \in \{1,2,\ldots M\}$ *and to estimate state at time k*;
- *if* $\max_{s\in\{1,2,\ldots M\}}\{\Omega(r_k = s)\} > \theta$ *and* $r_k = \arg\max_{s\in\{1,2,\ldots M\}}\{\Omega(r_k = s)\}$ *equal to* $r_{k-1} = m$

γ *= 1, keep only the particles in the most likely mode and normalize.*

else

γ *= 0, keep the particles in all modes*

end

end

4.2 Performance Evaluation

In this section we will apply the RMMPF algorithm to the nonlinear hybrid system estimation example from [49] and compare it with KMPF/OTPF and IMMPF algorithm. Moreover, based on the same example, we will show the sensitivity of RMMPF to the value of parameters to address the performance of RMMPF better.

4.2.1 Problem description

Consider the following nonlinear hybrid system from [49]. The system has 3 modes. The state equation of the system has form

$$x_k = f_i(k, x_{k-1}, v_k) \tag{4.1}$$

where i denotes the mode and

$$\begin{aligned}
f_1(k, x_{k-1}, v_k) &= 0.5x_{k-1} + 25\frac{x_{k-1}}{1+x_{k-1}^2} \\
&\quad + 8\cos(1.2k) + v_k \\
f_2(k, x_{k-1}, v_k) &= 0.5x_{k-1} + 25\frac{x_{k-1}}{1+x_{k-1}^2} \\
&\quad + 8\cos(1.2k) + 2 + v_k \\
f_3(k, x_{k-1}, v_k) &= 25\frac{x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) + v_k
\end{aligned}$$

For all the three modes the output equation is

$$y_k = \frac{x_k^2}{20} + w_k$$

In the above equations v_k and w_k are zero mean Gaussian noises with variances $\sigma_{v_k}^2$ and $\sigma_{w_k}^2$, respectively. In the remainder of this section we set $\sigma_{v_k}^2 = \sigma_{w_k}^2 = \sigma^2$.

In simulation the system stays in mode 1 during $1 \leq t < 30$, switches to mode 2 at time

$t = 30$ and stays in that mode for the interval $30 \leq t < 60$, then during $60 \leq t < 100$ the system is in mode 3.

The initial distribution of system is assumed to be known. For all filters the system starts at same point. 100 runs have been performed for each of the filters. Furthermore, in order to make the comparison more meaningful, the same random number streams were used for all filters.

4.2.2 Comparison with KMPF/OTPF

In this part we compare the proposed algorithm with KMPF/OTPF. In order to make a fair comparison with OTPF, we use the same performance measure, mean absolute error (MAE), the same particle size ($N = 100$), for KMPF (recall KMPF is the benchmark filter that has full knowledge of the real mode), and the same variance parameters ($\sigma^2 = 0.01, 0.05, 0.1, 0.5, 1$), as have been used in [49]. We also use the same mode transition matrix Π as in [49], such that for each mode the mode transition probability to itself is 0.95, while 0.025 to the other two modes. The threshold value θ for RMMPF is set to be 0.99.

Table 1 MAE for KMPF and RMMPF

σ^2	0.01	0.05	0.1	0.5	1
KMPF	0.29	0.44	0.55	1.03	1.41
RMMPF(N=100)	0.32	0.49	0.71	1.37	1.78

Table 2 Average times IMMPF has been used

σ^2	0.01	0.05	0.1	0.5	1
N = 100	29.7	44.0	50.1	69.3	78.73

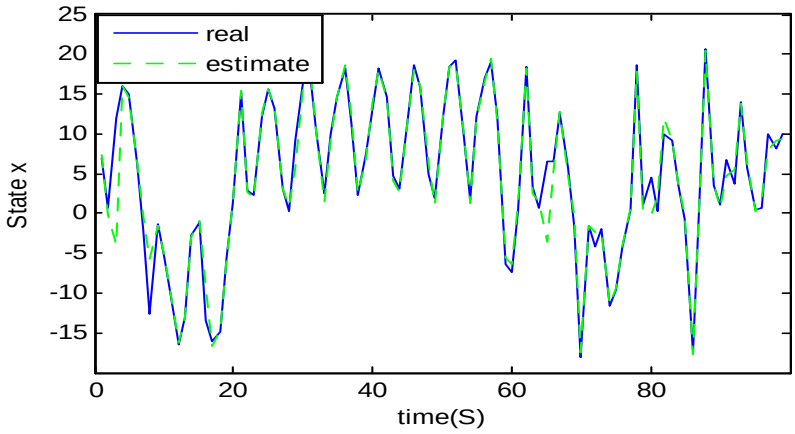


Figure 6 State estimation by RMMPF

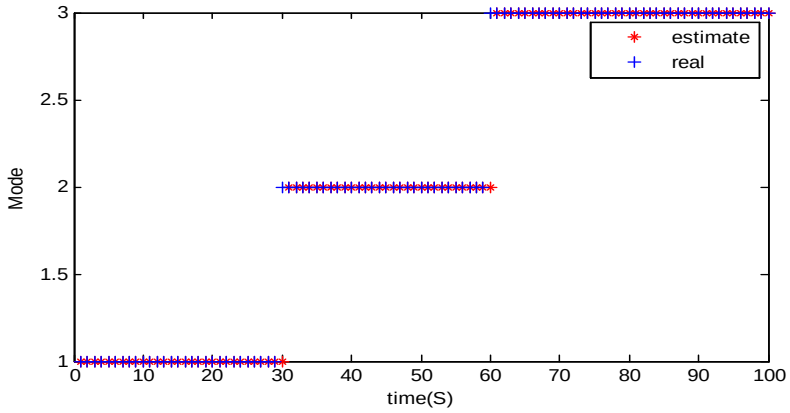


Figure 7 Mode estimation by RMMPF

Table 1 compares the performance for the estimation of the system state by KMPF and RMMPF. The proposed RMMPF estimator was run with $N/M = 100/3 \approx 33$ particles in each mode whereas KMPF was run with $N = 100$ particles. As is seen in Table 1 the proposed method performs only about 10% worse than KMPF for noise $\sigma^2 \leq 0.05$, around 30% worse for noise $\sigma^2 = 0.1$ and 0.5, and around 25% worse for $\sigma^2 = 1$. On the other hand, it has been reported in [49] that in a similar comparison with KMPF with $N = 100$

particles, the OTPF algorithm with $N/M = 300/3 = 100$ particles in one mode performs more than 50% worse for $\sigma^2 \leq 0.05$, more than 75% worse for $\sigma^2 = 0.1$, and more than 45% and 40% worse with $\sigma^2 = 0.5$ and 1, respectively. Moreover, in this example, it is apparent that the RMMPF algorithm has less computational load than the OTPF algorithm since the RMMPF uses only $N/M = 100/3 \approx 33$ particles in each mode, which is one-third size of that used in OTPF. Note however, that the number of times the IMMPF algorithm is used in the execution of the RMMPF will affect the computational load, i.e. the more times IMMPF is used the higher the computation cost will be. In order to evaluate how often IMMPF has been executed in the proposed RMMPF estimator, we display in Table 2 the average time in which IMMPF algorithm is executed for one 100 time step long run. We can tell from Table 2 that, with the same threshold value, as the variance σ^2 increases, the rate of IMMPF being used in RMMPF increases as well. This is due to the fact that as the noise level increases the mode weight gap between all possible modes decreases. Thus the dominance of one mode does not happen frequently, which leads to high frequency of the selection of IMMPF in the algorithm. Finally, in Figure 6 and Figure 7 we show the state estimate and mode estimate performance of one typical run of the proposed algorithm for $N = 100$ particles and variance $\sigma^2 = 0.5$. It can be seen that the estimated state matches the real state very well. Furthermore, as is seen in Figure 7 the mode estimation is very accurate as well, only 1 time step delay happens when the system switches.

4.2.3 Comparison with IMMPF

In this section some simulation results are presented for the IMMPF and the proposed RMMPF estimator. Both algorithms run with $N/M = 300/3 = 100$ particles in each mode.

The mode transition matrix Π is given by

$$\Pi = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}$$

and threshold value θ is set to be 0.9. The performance index is MAE, the same as before.

We present in Table 3 the performance of the IMMPF and RMMPF algorithms for various noise variance values. These results show that the proposed RMMPF algorithm can get similar estimation accuracy as IMMPF. Table 4 shows the average number of time steps in which IMMPF algorithm is used in the RMMPF for one 100 time step run. Note that when $\sigma^2 \leq 0.1$, IMMPF is used in less than 20 time steps in a 100 time step run. When $\sigma^2 = 0.5$ or 1, the IMMPF is executed more often, but still less than 50% in one run. The low frequency in which IMMPF is used in the RMMPF estimation contributes to its computation efficiency, which is summarized in Table 5. Table 5 gives comparison between the IMMPF and RMMPF in computational effort for various noise variance values. The computation times are counted for 100 runs for each different variance case. We note that the RMMPF saves 22.4% to 40.3% computational time compared to IMMPF algorithm.

Table 3 MAE for RMMF and IMMPF

σ^2	0.01	0.05	0.1	0.5	1
RMMPF	0.25	0.47	0.63	1.29	1.72
IMMPF	0.26	0.46	0.62	1.29	1.73

Table 4 Average times IMMPF has been used

σ^2	0.01	0.05	0.1	0.5	1
N = 300	11.5	16.6	19.2	33.4	44.9

Table 5 Calculation time for RMMPF and IMMPF

σ^2	0.01	0.05	0.1	0.5	1
RMMPF	20.1s	21.0s	21.9s	24.6s	26.1s
IMMPF	around 33.6s				

4.2.4 Sensitivity to θ

In order to gain insight into the sensitivity of RMMPF to the threshold θ we carried out several simulations by varying the value of θ, while keeping all other parameters fixed, i.e. $\sigma^2 = 0.1$, Π the same as in Section 4.2.2 and $N = 300$. The performance of RMMPF as function of the threshold value θ is compared with IMMPF both in terms of efficiency and accuracy in Figure 8. We observe that as the threshold varies from a low to a high value, the accuracy of RMMPF is improved, while the relative computational time of RMMPF increases as the threshold increases. Note that when θ has value around 0.93, RMMPF has performance as good as that of IMMPF, while the relative computational time is only about 62% compared to IMMPF. When θ is increased further the performance of RMMPF remains the same while the relative calculation time (compared to IMMPF) increases quickly and ends close to 1, corresponding to $\theta = 1$. In fact, when $\theta = 1$ RMMPF is the same algorithm as IMMPF. We conclude that in this case, the value $\theta = 0.93$ is the optimal

threshold value.

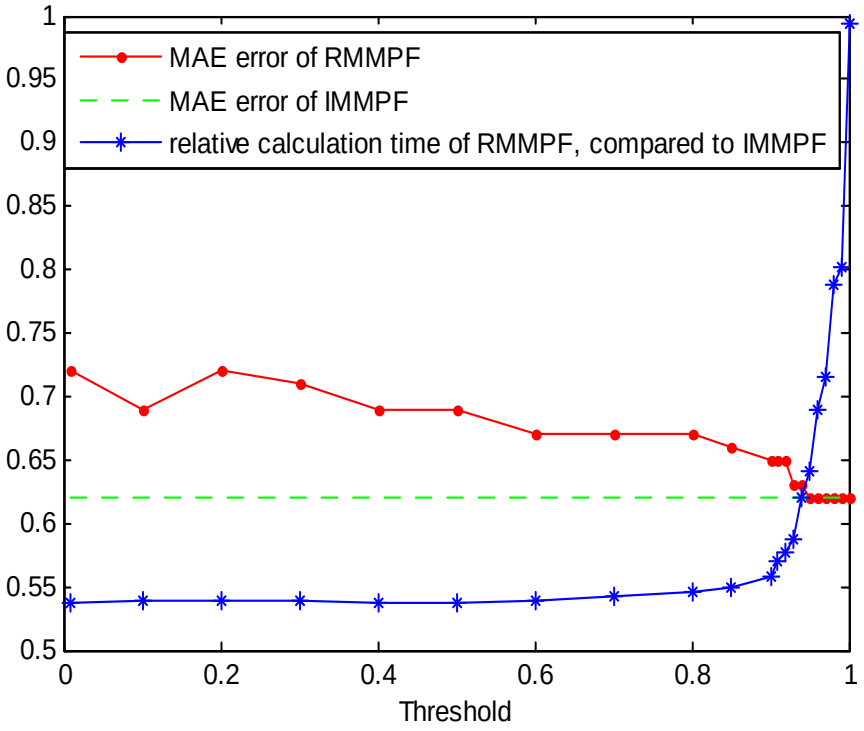


Figure 8. Performance versus Threshold

4.2.5 Number of particles

Next we illustrate the performance of RMMPF with respect to different number of particles. Here we have selected simulation parameters as: $\sigma^2 = 0.05$, Π the same as in Section 4.2.2 and $\theta = 0.95$. Similar to Section 4.2.4, the performance of RMMPF has been compared with IMMPF in two ways, i.e. efficiency and accuracy. Moreover, KMPF has also been

realized to offer a performance reference for different number of particles.

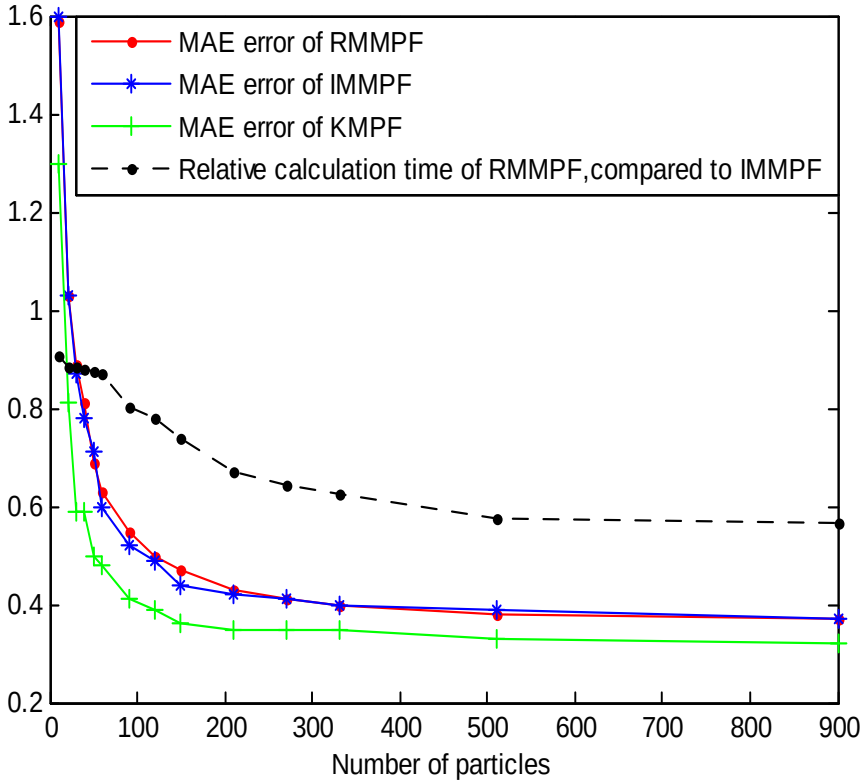


Figure 9. Performance comparison

In Figure 9 we see that RMMPF and IMMPF have similar estimation accuracy as a function of the number of particles. Furthermore, as the number of particles increases the performance of both two filters improves but remain inferior to the performance of KMPF. We note at the same time that the relative calculation time of RMMPF to that of IMMPF is decreasing. In particular, as the number of particles increases, RMMPF not only offers more accurate estimation but also offers better efficiency relative to IMMPF.

4.3 Application: Maneuvering Target Tracking

In this Section we intend to further validate the viability of the proposed RMMPF algorithm through a simulated maneuvering target tracking example. We adopt the same maneuvering target tracking example as the one addressed in [14]. Performance of RMMPF algorithm is compared to that of IMMPF to illustrate the algorithm's efficiency.

The maneuver system of interest has the form (1.2), where now

$$f_{r_k}(x_{k-1}, v_k) = Ax_{k-1} + Bv_k + u_{r_k}$$
$$g_{r_k}(x_k, w_k) = Cx_k + Dw_k$$

and

$$A = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = 0.1I_4$$
$$C = I_4, \; D = \frac{\sqrt{3}}{2} \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The state of the target at the kth time step is defined as $x_k \triangleq \left[d_{x,k}, s_{x,k}, d_{y,k}, s_{y,k}\right]$, where $d_{x,k}$(or $d_{y,k}$) and $s_{x,k}$(or $s_{y,k}$) denote the position and velocity of the target in the x (or y) direction respectively. Noises v_k and w_k are both modeled as zero mean Gaussian sequences with unit standard deviation. The sample time T_s is $2s$. The switching relevant term is u_{r_k},

where r_k is a three-state Markov chain with transition matrix Π given by

$$\Pi = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}$$

The following three modes correspond to three possible maneuver commands:

1. Straight, with $u_1 = [0,0,0,0]^T$;
2. Left turn, with $u_2 = [-1.225,-0.35,1.225,0.35]^T$;
3. Right turn, with $u_3 = [1.225,0.35,-1.225,-0.35]^T$.

A 100-time-step long target trajectory is generated as follows: 1) the target starts from state $[-500,0,-500,5]$ at time $k = 0$ and goes straight for 25 time steps; 2) a right turn is executed for 10 time steps; 3) the target goes straight for another 25 time steps; 4) the target turns left for 20 time steps; 5) the target goes straight again for 20 time steps.

In order to make a fair comparison, as before, we have the system start at a fixed point for all filters. One hundred simulation runs have been performed and the same random number streams were used for all filters. For RMMPF algorithm, the threshold θ is set to be 0.85.

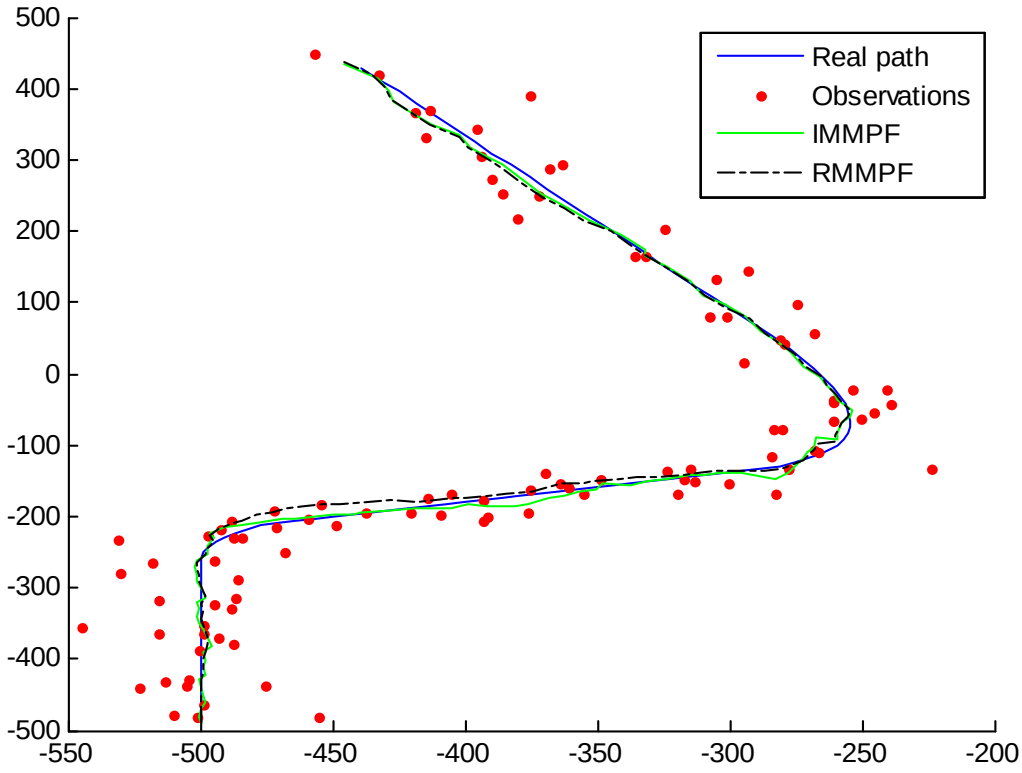


Figure 10 Maneuvering target trajectory and tracking results

Figure 10 shows the simulated maneuvering target trajectory, the observations and the estimated target trajectories resulting from the IMMPF and RMMPF algorithms for one typical run. As displayed in Figure 10, both IMMPF and RMMPF can estimate the traject-

ory from observations very well.

Table 6 Comparison of IMMPF and RMMPF

Algorithm	Time(s)	MAE	Times IMMPF used
IMMPF	838.9	7.25	/
RMMPF	651.3	7.43	56.9

In order to illustrate the viability and efficiency of RMMPF, we have compared in Table 6 the performance of RMMPF to IMMPF, based on 100 runs. The comparison of estimation error in Table 6 shows the similar estimation accuracy of IMMPF and RMMPF. As for the calculation efficiency, we see that RMMPF's calculation time is 22.4% less than that of IMMPF. It is also shown in Table 6 that for one 100 time step long run, the average number of times the IMMPF algorithm was executed was 56.9. It is worth mentioning that in this example, the system is quite noisy. As we discussed in Section 4.2.2, the high noise level would reduce the mode weight gap among all possible modes. Thus here one mode does not dominate often and the frequency of using IMMPF in RMMPF algorithm is high, which in turn causes more relative calculation time (compared to IMMPF). Another factor causing the increase in the computation time is the high mode switching rate. In general, mode switching makes several modes have comparable weights before the algorithm gets stabilized in one mode. Consequently, the more often the mode switches, the higher the computational cost in RMMPF comes. Note that although the high mode switching rate increases the computation time of RMMPF, the increase is not as obvious as that caused by the high noise level. As we can see, for this example, although the system is suffering from those two factors mentioned above, RMMPF can still offer good computation efficiency, together with good estimation accuracy.

Remark 3 *In order to be consistent with [49] we have made all performance comparisons here using the MAE index. However, we have also made the comparisons using the root*

mean square (RMS) of the estimation error which has shown similar results.

4.4 Conclusion

In this Chapter we presented a high accuracy, low computation load particle filter algorithm and compared its performance with OTPF and IMMPF algorithms. We demonstrated in simulation that the proposed method outperforms OTPF not only in estimation accuracy but also in terms of computational effort. Compared with IMMPF, the proposed method has been shown to work almost as well as IMMPF in terms of estimation accuracy but with considerably lower computational effort. The efficiency and good performance of the proposed algorithm makes it practical and robust for tracking a target in a complex situation, as we have demonstrated in Section 4.3 by the maneuvering target tracking example.

It is easy to see that the threshold value will affect both the estimation accuracy and the computational time. Indeed, the closer the threshold value is to 1 the algorithm will select the IMMPF at a higher rate at the expense of higher computational cost but with increased accuracy in the estimation. As we demonstrated in Section 4.2.4, there exists an optimal value for threshold value which offers the best combination of low computational cost and high estimation accuracy for RMMPF algorithm. However, finding the optimal value in practical applications may require some extra research efforts. Furthermore, when the number of particles is large, say $N = 10^4$, it may not be necessary to use all the particles at each time step to select the correct mode. Instead, a smaller number of particles in each mode may be sufficient for this purpose. After the correct mode is estimated, full set of particles in this mode can be used to estimate the state and to evolve into the next time step. In this way, the computational efficiency is expected to get improved even further.

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

In this dissertation we studied the identification of complex dynamic systems as well as hybrid system estimation.

For the identification part, we proposed a scheme to identify a complex stochastic dynamic system based on a black-box model, that is, the system is modeled based only on output data. The system under study is a system whose underlying space is the union of strong attraction domains. The system exhibits a behavior such that it spends a long time in one strong attraction domain before transitioning to another one. System showing this behavior can be found in many applications. Considering the nature of this system, we modeled it as a hybrid system, in particular, it is a strong attraction domain featured hybrid system (SAFHS). Two principal features for this type of a hybrid system are that the boundaries between the strong attraction domains (modes) are nonlinear and the dynamic behavior within each strong attraction domain (modes) can be highly nonlinear, e.g. limit cycle.

The identification scheme developed in this dissertation was based on finite dimensional approximations of compact operators, spectral theory for non-reversible Markov chains, identification techniques for hidden Markov models (HMM), and identification techniques for linear and non-linear dynamics.

First, after we discretized the state and output spaces and approximated the underlying Markov process and corresponding output process by a finite state processes, we detected the state sequence based on the output sequence through the identification of a HMM,

where we have adopted a newly developed HMM identification algorithm and extended it to high dimensional state space case. Moreover, we have discussed the system attributes which lead to the unique estimation for the hidden state. Then, through examples, we have experimentally verified the presented uniqueness estimation analysis as well as shown the effectiveness of the proposed HMM identification algorithm.

After identifying the state sequence utilizing HMM modeling techniques we worked out the transition laws for the Markov chain. Then spectral theory for non-reversible Markov chains was used to identify the number of partition components (strong attraction domains, modes) as well as the partition itself, and the transition law between those components. An example for identifying a discrete time dynamics system that contains two strong attraction domains (modes) was carried out. Through this example, it has been illustrated that the proposed identification approach can accurately estimate the state transition matrix, state-output matrix, system modes, state space partition regions as well as the modal transition matrix.

With the state space partition regions being identified successfully, we next mapped the state sequence onto the partition and through the association between the state and output strings we classified the output symbols (and substrings) into partition components (clusters) in the output space corresponding to that in the state space. It has to be stressed that the dynamics that govern the modal behavior in state space are the same as the ones that govern the modal behavior in the output space. Given the output strings in each components in output space, we identified the local dynamics. For a partition component that only contains an equilibrium point we adopted linear identification techniques, while for a partition component that contains highly nonlinear behavior, we explored a kernel based identification approach. So far we have identified a typical nonlinear behavior, i.e. limit cycle behavior through a kernel principal component analysis algorithm. For the local dynamics identification, examples were presented, both in linear identification and nonlinear identification cases, to demonstrate the performances of the techniques adopted or

developed.

Once we complete the above identification procedure we obtained a finite collection of local dynamic models as well as a Markov transition law that governs the dynamics between the local models. The resulting model is therefore of hybrid nature, i.e. associated with each local model is a modal variable taking value in a finite set, the dynamics of the modal variable is governed by the finite state Markov process. Furthermore, each local model is associated with a particular partition component of the output space. This fulfilled our goals for the identification of a complex dynamic system.

In the estimation part, we presented a high accuracy, low computational load method for nonlinear/non-Gaussian hybrid system. The efficiency and accuracy of the proposed algorithm have been illustrated by examples. Moreover, its good performance makes it practical and robust for tracking a target in a complex situation, as we have demonstrated by a simulated maneuvering target tracking example.

5.2 Future Work

In this dissertation, the complex dynamic system under study is an autonomous system, i.e., it is a system without input. We plan to extend our research to systems with inputs and develop input dependent hybrid models for such systems.

Second, in the local dynamics identification part, we have only studied the behavior of a limit cycle. The identification of local dynamics for other types of highly nonlinear strong attraction domains that contain more than one attractors or contain one attractor with other nonlinear behaviors, e.g. strange attractors, have not yet been attempted. These will be part of our future research interests as well.

Bibliography

[1] E. Abd-Elrady and T. Soderstrom. Bias analysis in least-squares estimation of periodic signals using nonlinear ODEs. *Preprint*, 2004.

[2] P.M. Anderson and A. A. Fouad. *Power System Control and Stability*. John Wiley and Sons, New York, 2002.

[3] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

[4] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Trans Automat Contr*, AC-50:1567–1580, 2005.

[5] H. Blom and E. Bloem. Exact bayesian and particle filtering of stochastic hybrid systems. *IEEE Transactions on Aerospace and Electronic Systems*, 43:55–70, 2007.

[6] H. Blom and E.Bloem. Interacting multiple model joint probabilistic data association avoiding track coalescence. In *Proceedings of the 41st IEEE CDC*, pages 3408–3415, 2002.

[7] H. Blom and Y. Shalom. The interacting multiple model algorithm for systems with markovian switch coefficients. *IEEE Transactions on Automatic Control*, 33:780–783, 1988.

[8] H.A.P. Blom. An efficient filter for abruptly changing systems. In *Proceedings of the 23rd IEEE CDC*, pages 656–658, 1984.

[9] Y. Boers and H. Driessen. Hybrid state estimation: a target tracking application. *Automatica*, 38:2153–2158, 2002.

[10] Y. Boers and J.N. Driessen. Interacting multiple model particle filter. *IEE proc. Radar Sonar Navig.*, 150:344–349, 2003.

[11] J.M. Bruckner, H.R.W. Scott, and G.R. Rea. Analysis of multimodal systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-9:883–888, 1973.

[12] M. Dellnitz and R. Pries. Congestion and almost invariant sets in dynamical systems. In *Proc. of the Symbolic and Numerical Scientific Computation(SNSC'01)*, pages 183–209. Springer-Verlag Berlin Heigelberg, 2003.

[13] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts. In *Advances in Neural Information Processing Systems 16(Proc. NIPS*2003)*, pages 44–48. MIT Press, 2004.

[14] A. Doucet, N. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, 49:613–624, 2001.

[15] Y. Engel, S. Mannor, and R. Meir. The kernel recursive least squares algorithm. *IEEE Transactions on Signal Processing*, 52(8):2275–2285, 2004.

[16] Y. Ephraim and N. Merhav. Hidden markov processes. *IEEE Transactions on Information Theory*, 48:1518–1569, 2002.

[17] F.Lauer and G. Bloch. Switched and piecewise nonlinear hybrid system identification. In *Proc. of the 11th Int. Conf. on Hybrid Systems: Computation and Control(HSCC)*, pages 340–343. Springer, 2008.

[18] M.I. Freidlin and A.D. Wentzell. *Random Perturbations of Dynamical Systems*. Springer, New York, 1998.

[19] Ferrari-Trecate G., Muselli M., Liberati D., and Morari M. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.

[20] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Nover approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F*, 140:107–113, 1993.

[21] W. Huisinga, S. Meyn, and C. Schutte. Phase transitions and metastability in markovian and molecular systems. *The Annals of Applied Probability*, 14(1):419–458, 2004.

[22] W. Huisinga and B. Schmidt. Metastability and dominant eigenvalues of transfer operators. *'Advances in Algorithms for Macromolecular Simulation', Lecture Notes in Computational Science and Engineering*, 2005.

[23] I.T. Jolliffe. *Principal component analysis, 2nd Edition*. Springer, New York, 2002.

[24] A Juloski and W.P.M.H. Hemmels S. Wieland. A bayesian approach to the identification of hybrid systems. *IEEE Trans Automat Contr*, AC-50:1520–1533, 2005.

[25] A.L Juloski, W.P.M.H. Heemels, G. Ferrari-Trecate, R. Vidal, and J.H.G. Niessen. Comparison of four procedures for the identification of hybrid systems. *Hybrid Systems: Computation and Control*, Volume 3414:354–369, 2005.

[26] H.K. Khalil. *Nonlinear systems (2nd ed)*. Prentic-Hall, Upper Saddle River, NJ, 1996.

[27] H. Laurberg. Uniqueness of non-negative matrix factorization. In *IEEE/SP 14th Workshop on Statistical Signal Processing*, pages 44–48, 2007.

[28] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[29] D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13:556–562, 2001.

[30] A. Logothetis and V. Krishnamurthy. Expectation-maximization algorithms for map estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, 47:2139–2156, 1999.

[31] L.R.Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

[32] E. Mazor, A. Averbuch, Y. shalom, and J. Dayan. Interacting multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, pages 103–123, 1998.

[33] S. McGinnity and G. Irwin. Multiple model bootstrap filter for maneuvering target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 36:1006–1011, 2000.

[34] J.M. Mendel. *Maximum-Likelihood Deconvolution: A Journey into Model-Based Signal Processing*. Springer, New York, 1990.

[35] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *in Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press, 2005.

[36] O. Nelles. *Nonlinear system identification*. Springer, New York, 2001.

[37] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems: A tutorial. *European Journal of Control*, 13:242–260, 2007.

[38] T. Runolfsson and Y. Ma. Model reduction of nonreversible markov chains. In *Proceedings of the 46th IEEE CDC*, pages 3739–3744, 2007.

[39] B. Scholkopf, A. Smola, and K.R. Muller. Kernel principal component analysis. In *Artificial Neural Networks-ICANN'97*, pages 583–588. Berlin, 1997.

[40] B. Scholkopf, A. Smola, and K.R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[41] B. Scholkopf and A.J. Smola. *Learning with kernels*. MIT Press, 2002.

[42] Ch. Schütte, W. Huisinga, and P. Deuflhard. Transfer operator approach to conformational dynamics in biomolecular systems. *Ergodic Theory, Analysis, and Efficient Simulation of Dynamical Systems*, pages 191–223, 1999.

[43] Y. Shalom, S. Challa, and H.Blom. IMM estimator versus optimal estimator for hybrid systems. *IEEE Transactions on Aerospace and Electronic Systems*, 41:986–991, 2005.

[44] J. Shlens. A tutorial on principal components analysis. 2005.

[45] L.I. Smith. A tutorial on principal components analysis. February 26, 2002.

[46] T. Soderstrom, T. Wigren, and E. Abd-Elrady. Periodic signal analysis by maximum likelihood modeling of orbits of nonlinear ODEs. *Automatica*, 41:793–805, 2005.

[47] E.D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions Automatic Control*, AC-26:326–356, 1981.

[48] S.H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001.

[49] S. Tafazoli and X. Sun. Hybrid system state tracking and fault detection using particle filters. *IEEE Transactions on Control Systems Technology*, 14:1078–1087, 2006.

[50] F.J. Theis, K. Stadlthanner, and T. Tanaka. First results on uniqueness of sparse non-negative matrix factorization. In *European Signal Processing Conference*, 2005.

[51] S.V. Vaerenbergh, J. Via, and I. Santamaria. Nonlinear system identification using a new sliding-window kernel RLS algorithm. *Journal of Communications*, 2:1–8, 2007.

[52] J.M. van den Hopf and J.H. van Schuppen. Positive matrix factorization via extremal polyhedral cones. *Linear Algebra and its Applications*, 293:171–186, 1999.

[53] B. Vanluyten, J.C. Willems, and B. Moor. A new approach for the identification of hidden markov models. *Proceedings of the 2007 IEEE CDC*, pages 4901–4905, 2007.

[54] B. Vanluyten, J.C. Willems, and B. Moor. Equivalence of state representations for hidden markov models. *Systems and Control Letters*, 57:410–419, 2008.

[55] V. Vapnik. *Statistical learning theory*. Wiley-Interscience, New York, 1998.

[56] R. Vidal. Identification of PWARX hybrid models with unknown and possibly different orders. In *Proceedings of the 2004 American Control Conference, Boston, MA*, 2004.

[57] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions Pattern Anal and Machine Learn*, 27:1945–1959, 2005.

[58] T. Wigren, E. Abd-Elrady, and T. Soderstrom. Least squares harmonic signal analysis using periodic orbits of ODEs. In *Proc. of 13th IFAC Symposium on System Identification*, pages 1584–1589. Rotterdam, The Netherlands, August 2003.

[59] T. Wigren and T. Soderstrom. Second order ODEs are sufficient for modeling of many periodic signals. *Tech. Rep 2003-025 Department of Information Technology, Uppsala University, Uppsala, Sweden*, 2003.

[60] A.S. Willsky and B.C. Levy. Stochastic stability research for complex power systems. *Lab. Inf. Decision Systems, MIT*, Report ET-76-C-01-2295, 1979.

[61] Y. Xu, D. Zhang, F. Song, J. Yang, Z. Jing, and M. Li. A method for speeding up feature extraction based on KPCA. *Neurocomputing*, 70(4-6):1056–1061, 2007.

[62] Y. Zhai and M. Yeary. An intelligent video surveillance system based on next-generation stochastic tracking. 2007. Preprint.