# INFERENCE OF PIECEWISE LINEAR SYSTEMS WITH AN IMPROVED METHOD EMPLOYING JUMP DETECTION

AHMET MELİH SELÇUK

SEPTEMBER 2007

# INFERENCE OF PIECEWISE LINEAR SYSTEMS WITH AN IMPROVED METHOD EMPLOYING JUMP DETECTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

AHMET MELİH SELÇUK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF SCIENTIFIC COMPUTING

SEPTEMBER 2007

Approval of the Graduate School of Applied Mathematics

Prof. Dr. Ersan AKYILDIZ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Bülent KARASÖZEN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Hakan ÖKTEM
Supervisor

Examining Committee Members

Assist. Prof. Dr. Hakan ÖKTEM

Assoc. Prof. Dr. Azize HAYFAVİ

Assist. Prof. Dr. Kasırga YILDIRAK

Dr. Serhan DURAN

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Ahmet Melih Selçuk

Signature:

# Abstract


INFERENCE OF PIECEWISE LINEAR SYSTEMS WITH AN IMPROVED METHOD EMPLOYING JUMP DETECTION



Selçuk, Ahmet Melih
M.S., Department of Scientific Computing
Supervisor: Assist. Prof. Dr. Hakan ÖKTEM



September 2007, 60 pages



Inference of regulatory relations in dynamical systems is a promising active research area. Recently, most of the investigations in this field have been stimulated by the researches in functional genomics. In this thesis, the inferential modeling problem for switching hybrid systems is studied. The hybrid systems refers to dynamical systems in which discrete and continuous variables regulate each other, in other words the *jumps* and *flows* are interrelated. In this study, piecewise linear approximations are used for modeling purposes and it is shown that piecewise linear models are capable of displaying the evolutionary characteristics of switching hybrid systems approximately. For the mentioned systems, detection of switching instances and inference of locally linear parameters from empirical data provides a solid understanding about the system dynamics. Thus, the inference methodology is based on these issues. The primary difference of the inference algorithm is the idea of transforming the switching detection problem into a jump detection problem by derivative estimation from discrete data. The jump detection problem has been studied extensively in signal processing literature. So, related techniques in the literature has been analyzed carefully and suitable ones adopted in this thesis. The primary advantage of proposed method would be its robustness in switching detection and derivative estimation. The theoretical background of this robustness claim and the importance of robustness for real world applications are explained in detail.

Keywords: Piecewise linear models, hybrid systems, inferential modeling, switching networks, gene networks, local polynomial fitting, jump detection

# Öz

## PARÇALI DOĞRUSAL SİSTEMLERİN SIÇRAMA TESBİTİ KULLANAN GELİŞTİRİLMİŞ BİR METOT İLE ÇIKARIMI

Selçuk, Ahmet Melih
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi: Assist. Prof. Dr. Hakan ÖKTEM

Eylül 2007, 60 sayfa

Dinamik sistemlerdeki yönetici ilişkilerin çıkarımı, gelişme vaat eden aktif bir inceleme alanıdır. Son zamanlarda, bu alanda yapılan çoğu araştırma fonksiyonel genomik çalışmaları tarafından teşvik edilmiştir. Bu tezde, değişmeli hibrit sistemlerin çıkarımsal modellenmesi üzerine çalışılmıştır. Hibrit sistemler, kesikli ve sürekli değişkenlerin birbiriyle etkileştiği, yani *akış* ve *sıçrama* hareketlerinin birbiriyle ilişkili olduğu dinamik sistemlerdir. Bu çalışmada, modelleme için parçalı doğrusal yaklaştırma kullanılmış ve parçalı doğrusal modellerin değişmeli hibrit sistemlerin evrilme karakteristiğini yaklaşık olarak ifade edebilieceği gösterilmiştir. Sözü edilen sistemlerde, değişme anlarının tesbiti ve lokal doğrusal parametrelerin ampirik verilerden çıkarımı sistem dinamiği hakkında sağlam bilgiler sağlar. Dolayısıyla, çıkarım metodolojisi bu hususlar üzerine oturtulmuştur. Çıkarım algoritmasının başlıca farklılığı, değişme anı tesbiti probleminin, kesikli veriler üzerinden türev tahmini metoduyla sıçrama tesbiti problemine dönüştürülmesi fikridir. Sıçrama tesbiti problemi, sinyal işleme literatüründe geniş ölçüde çalışılmıştır. Bu tez kapsamında, bu literatürdeki ilgili teknikler dikkatlice incelenmiş ve uygun olanlar kullanılmıştır. Sunulan metodun temel avantajı, değişme anı tesbiti ve türev tahmininin gürbüz (robust) olarak yapılabilmesidir. Metodun gürbüzlüğünün teorik temeli ve gürbüzlüğün

gerçek uygulamalardaki önemi detaylı olarak açıklanmıştır.

Anahtar Kelimeler: Parçalı doğrusal modeller, hibrit sistemler, çıkarımsal modelleme, değişmeli ağlar, gen ağları, lokal polinom yakınlaştırması, sıçrama tesbiti

# Acknowledgments

I would like to express my sincere gratitude to my supervisor for his help and guidance in my studies

I thank my project partner Nilufer, for her support and friendship

I deeply thank the members of the Graduate School of Applied Mathmeatics and my friends for their suggestions during the period of writing the thesis.

This work has been supported by Tubitak (The Scientific Technological Research Council of Turkey) under project "Modeling Mulistationary Processes by Using Hybrid System Formulation" (104T133). The simulations used for empirical data generation are developed by project supervisor Assist. Prof. Dr. Hakan Oktem and Nilufer Caliskan.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Mathematical Modeling of Dynamical Systems

A dynamical system describes the evolution of the variables interacting with each other and the 'environment' over time. Here, the term *evolution* implies position and velocity. To explain the underlying laws governing the system dynamics, mathematical models have been proposed. For example, a system of first order ordinary differential equation (ODE), $dx/dt{=}F(x{,}t)$, is a model that explains the velocity in space($dx/dt$) as a function($F$) of the position($x$) and the time($t$). This type of abstraction -well defined mathematical relations- allows;

- studying the dynamical system's behavior under different conditions
- predicting the dynamical system's future evolution for given initial conditions
- planning the necessary interventions to bring the dynamical system to a desired state

Therefore, mathematical modeling of dynamical systems is a fundamental problem in understanding and/or controlling the physical, chemical and biological phenomena. In this study, the focus will be on the switched hybrid systems at which the regulatory relations are defined by mimicking the switch in a circuit that can determine the state of a variable in the system, either ON or OFF. The intervariable regulations occur subject to the threshold phenomena and we are after finding a reliable, robust and computationally efficient algorithm to infer the rules of switching; in other words for any given variable $x_j$, we try to find

- the variable $x_i$ that regulates the switching in $x_j$
- the type of effect $x_i$ has on $x_j$ (repression or activation)
- the value of each threshold and the corresponding state space partition

The aspects mentioned up to here will be explained in the succeeding sections of this study. The objectives explained above represents that the method is applicable only for a particular problem; the efforts explained in detail throughout this work are concentrated for solving the problem of inferring a particular sub-class of nonlinear dynamical systems. It is reasonable to specify the domain of the proposed method since every problem should be undertaken according to its own considerations and necessary modifications should be adopted considering the nature of the problem.

## 1.2 Problem Statement and Scope

Every mathematical model is in fact an abstraction which is incapable of representing the original problem with complete accuracy; however it captures the essential features of the original problem [2]. Inferring the dynamics of a system from empirical findings is an important problem that have grabbed the attention of many researchers from different fields. A few exemplary fields include -but not limited to- gene network modeling [28, 4], climate change investigation [3] and population dynamics studies [10]. The steps succeeding the inference of model parameters is different depending on the goal to be accomplished. Intervention planning focuses on planning the required interventions to bring the system to a desired state whereas inverse theory aims to reconstruct the history. The common point in these studies is the requirement of a mathematical model for describing the observed features of the system in a quantitative manner. In this aspect, the first critical step of building a model is the model class selection. Choice of the appropriate model class based on a priori knowledge or empirical evidence requires close attention since poor model selection would lead to incapable models even the model parameters are selected optimally. A criterion that can be useful at this stage is the qualitative behavior matching. According to this criteria, “a model class C is considered only if it is capable of exhibiting the qualitative behavior of primary interest” [26].

In this study, the focus is on a particular type of nonlinearity in the evolution, in which the variables regulate each other via threshold phenomena and display piecewise linear evolution where the linearity is violated by switching occurrences only.

Thus, as qualitative behavior matching criterion suggests, we restricted our attention to a model class which is capable of representing switching behavior regulated by threshold phenomena. For that reason, the piecewise linear hybrid system formalism with single threshold state space partitioning is adopted in this work. Hence, the dynamical systems whose regulatory relations can be explained by threshold phenomena -according to analytical or empirical evidence-, can be considered as the domain of the modeling approach that is proposed here. Detailed discussion on the importance of this model class and the particular systems that can be modeled by this formulation is given in next sections. Moreover, basic ideas in related works on classification of nonlinear systems depending on the way they violate linearity is also provided.

With the predetermined model class for the particular type of phenomena, the efforts are concentrated on the inference of model parameters in this work. These parameters can be grouped in two parts referred here as the switching parameters and the continuous evolution parameters. The so-called switching parameters include the parameters defining the regulation on ordered variable tuples (effect of $var_i$ on $var_j$ in terms of activation, repression or none) and the corresponding threshold values. The continuous evolution parameters govern the in-state movement of system and they are defined separately for each state.

This thesis studies a modified parameter inference method based on using derivative estimation in switching detection which we consider as the original contribution of the study. The piecewise linearity of evolution refers to the linearity of evolution in a state between two consecutive switchings and correspondingly, the switchings are the instances between two different linear parts where the smoothness and linearity fails. After detecting the switchings, the best fit values of continuous evolution parameters are found by L2 error minimization individually for each state. On the other hand, the inference of regulatory relations among the variables in the system requires some further work and certain assumptions to be satisfied. This process is explained in detail as well.

In the second chapter, preliminaries of the referred topics are mentioned. Firstly, the hybrid system formalism is introduced. The fundamental concepts in the theory of hybrid systems is briefly explained and the important points relevant to the problem in consideration are outlined. Then, theoretical background of inference problem is given and key points of inferential modeling are discussed. This chapter also contain preliminary information on networks.

The piecewise linear models are discussed in the third chapter. Switching behavior and threshold phenomena are explained in detail and the specifications of suggested piecewise linear models are given. Having stated the specifications of the problem, the fourth chapter is reserved for the solution; the inference algorithm is introduced. The critical steps of the method are explained together with the computer codes developed for each task. The demonstrations on hypothetical examples are also given to clarify the rationale and implementation of each step.

In the fifth chapter, the implementation of the method is performed on simulated data generated for an exemplary dynamical system. Advantages and shortcomings of the method are discussed and possible improvements are mentioned. The final remarks and conclusions on the entire work are made in the final chapter.

# Chapter 2

# Background

## 2.1 Hybrid Systems

Hybrid systems stand for the class of dynamical systems in which continuous and discrete variables regulate each other. In general, we can consider two types of occurrences in hybrid systems; *the flow*, that describes the change of continuous variables in time and *the jump*, which refers to any change in the discrete variables. Coexistence of continuous and discrete variables in hybrid systems allows modeling flows incorporated with jumps. After this general introduction, the formal definition of a hybrid system can be made as follows:

A hybrid system H is a collection H = (Q, X, Init, f, Inv, E, G, R), where

- $Q = \{q_1, q_2 ...\}$ is a set of **discrete states**
- $X = \mathbb{R}^N$ is a space of **continuous variables**
- $Init \subseteq Q \times X$ is a set of **initial states**
- $f : Q \times X \rightarrow \mathbb{R}^N$ is a **vector field**
- $Inv : Q \rightarrow P(X)$ is an **invariant set**
- $E \subseteq Q \times Q$ is a set of **edges**
- $G : E \rightarrow P(X)$ is a **guard condition**
- $R : E \times X \rightarrow P(X)$ is a **reset map**

In this formalism, the vector field $f$ is the function governing the continuous dynamics in a given invariant set. Accordingly, invariant sets, $Inv$, can be considered as a partitioning of the state space with respect to different continuous evolution characteristics. The edges, $E$, correspond to crossings between invariant sets; they represent the transitions among discrete states, $Q$. A guard condition, $G$, is defined for every edge in terms of well defined mathematical relations on the discrete and continuous variables in the system. The state transition represented by each edge occurs when the corresponding guard conditions are satisfied.

Many systems we encounter in everyday life can be considered as hybrid systems. In electrical control circuits, continuous phenomena are interrupted by switches opening and closing, or diodes going on or off. Likewise, in chemical process control the continuous evolution of chemical reactions is controlled by valves and pumps[22]. In general, almost every continuous time feedback control systems operating with discrete feedback signals are simple and widely used examples of hybrid systems. In these systems, the observations on system in continuous time are used and when certain conditions are satisfied, the discrete feedback signal initiates the application of the necessary action. The application of this principle can be observed in diverse fields, a fuse protecting the circuit from fluctuations in voltage, a thermostat keeping the temperature in the specification limits or an industrial quality control system employing continuous time measurements for ensuring the manufacturing specifications are all developed on this idea.

Thermostats used in almost every sort of heaters and coolers are simple hybrid systems where the change of temperature in continuous time, -*the flow*-, determines the switching of system on or off, -*the jump*-, according to certain conditions. The discrete events, the edges, are the ON/OFF and OFF/ON switchings of the heat source in the system. No switching occurs while the temperature is within the specification limits. When the lower or upper limit is exceeded, the system switches the operation mode to take the temperature to the desired level. The operation mode of the system controls the inflow/outflow of heat to the system and in turn the temperature level determines turning the heat source ON or OFF.

The thermostat system is explained here since it is useful in exemplifying the basic concepts in hybrid systems. The temperature is the continuous variable that is desired to be kept in a certain interval. The operation mode of the heat source (can be a heater or cooler) is the discrete variable in the system. Hence the operation state can be represented as a binary value -0 or 1- and the value of this variable at any time can be determined by observing the system temperature. Thus, the temperature has a direct regulatory effect on the operation state. This effect of continuous variables on the system state is described by guard conditions in the hybrid system formalism. In this control problem, the guard conditions are introduced via certain threshold values which discretize the continuous variables into Boolean variables representing whether the continuous variables value is above or below the threshold. This type of thresholds can be the design specifications of the man-made control systems or they can arise from the inherent characteristics of other dynamical systems in nature. Threshold phenomena is discussed in further detail in Section 3.2.

On the other hand, the change of temperature in time displays a continuous evolution that can be explained by a system of differential equations. As long as the operation mode is not changed, the instantaneous temperature change can be represented by the same system of differential equation(s). The state space partition in which the evolution of continuous variable remains unchanged is called the invariant space. As the operation state controls the heat inflow/outflow provided by the heat source to the system, the binary variable representing the operation state regulates the temperature change in continuous time by changing the coefficients of the differential equation.

The hybrid system formalism is also applicable to dynamical systems in nature. Although the regulatory relations among the variables may not be defined that straightforward, certain qualitative features of the dynamical system can make hybrid system formalism favorable. For example, threshold phenomena appearing in different biological and physical systems (action potential threshold in cells [6], switching behavior in gene networks) can be that sort of motivation for employing hybrid systems. In the simplest scheme, threshold phenomena refers to the type of regulatory relation at which a variable attaining a certain value -the threshold- results in a change in the evolution of one or more variables in the system. In the hybrid system formalism, the value of the variable relative to the threshold can be defined as a discrete variable and these discrete variables would define the guard conditions determining the state transitions. In this way, invariant spaces would be defined as a state space partition in the classical hybrid system formalism. Each threshold can be considered as a hyperplane in the separating the state space into subspaces, e.g. if there are $n$ variables in the system and one threshold for each variable, there will appear $2^n$ invariant subspaces. In this particular case, the state of the system can be represented as a binary string of length $n$ consisting of zeros and ones denoting whether the corresponding variable is above or below the threshold.

## 2.2 Inference Problem

The problem of understanding the nature of a dynamical system has a broad perspective. We can consider different subclasses for this problem based on the approach of learning the governing dynamics. A predictive learning approach aims to predict future states of the system from observations of the present and past states of the system whereas a diagnostic learning approach aims to infer the probable past states of the system that might have led to the present state of the system. On the other hand, the objective may not be to predict the future or explain the past, but to provide a theoretical basis for any specific physical phenomena and this can be considered as an application based learning approach [8]. The possible ways of learning mentioned here are built on learning from observations and building a mathematical model would be useful for understanding the physical phenomena in all three aspects. In this type of empirical approach, inferential modeling would be a suitable approach to construct the mathematical model.

The complexity of the inference problem is mostly determined by the qualitative dynamical features of the system in consideration. For example, linear behavior is easily tractable and predictable. On the other hand, when the nonlinearity appears, it becomes much harder to extract useful information from empirical findings. It can even become impossible to infer system dynamics depending on the type of nonlinear behavior; inference for a chaotic nonlinear system can be an example. It can be possible to build up mathematical models for chaotic systems through analytical findings using the laws governing the system and a priori knowledge on probable variable interrelations. On the other hand, inference is the method of using empirical findings and in chaotic systems it is hard to interpret the underlying dynamics displaying the observed qualitative features.

As it is a challenging problem, inferential modeling of nonlinear dynamical systems has received growing interest. Development of instrumentation technologies providing high throughput data, increased computational power and more efficient inference algorithms have enabled inference of dynamical features of those systems from empirical observations. Some exemplary challenges where the inference methodology can suggest promising results in the inference of nonlinear dynamics can be given as:

- Inference of gene expression dynamics: After development of expression microarray technology, the availability and quality of data on gene expression measurements made it possible to study the dynamics of genes controlling biological phenomena. As a results, the problem of gene network inference has grabbed the intention of researchers [28, 4, 9]. Gene network inference suggests potential use in intervention planning and drug discovery [31].

- Calibration of ecosystem models: Accurate predictions on ecosystems are important to predict the future of the species in the ecosystems. information on population. For marine ecology, satellite ocean color data [18] can be utilized for longer term forecasts.

- Parameter estimation in flux balance models: Fermentation is a biochemical process used in manufacturing of different goods. Understanding the dynamics of batch fermentation could be utilized for finding the optimal parameters for fermentation processes [29].

Some of the challenging nonlinear dynamical system inference problems, including but not limited with the gene regulatory networks exhibit a switching nature due to the threshold phenomenon. In other words they are typical hybrid systems and governed by switching differential equations. Dynamical properties of those types of networks and their simplified versions were already studied as abstract gene network models [11]. Besides, some dynamical systems which do not fit to global models expressed by elementary mathematical expressions can be approximated by fitting each portion of the model locally to a simple expression.

### 2.2.1 Key Concepts on Inference

Inference is the method of estimating the system dynamics from empirical observations. The continuous evolution can be represented perfectly only with a continuous trajectory, i.e. with infinitely many observations. Naturally, storing the observations in continuous time is not possible with finite bits of storage. Hence the empirical data are available as a finite sample. With sufficient measurement accuracy and sampling frequency, it is still possible to extract the underlying dynamics from discrete sample. In other words, we can mimic the continuous system with a finite sample and employ the inference method of discrete systems on that sample. In the following equations, $X$ denotes the set of values of independent variables over time. $y_k$ denotes the vector containing the values of dependent variables at time $k$ and $\widehat{y}_k$ denotes the corresponding estimates. The inference problem can be interpreted as a discrete optimization problem as follows:

$$\widehat{y}_k = f_p(X),$$

$$\widehat{p} = arg \min_p \left( \sum_{k=1}^{n} C(y_k, \widehat{y}_k) \right). \tag{2.1}$$

In this equation, the function $f$ represents the mathematical relation used to estimate the values of dependent variables. $C$ is a cost (or penalty) function, which would typically be as a distance function in this case and the objective of the optimization problem is to find the set of parameters $p$ that minimizes the total penalty. Hence, to define an inference problem as an optimization problem, a well defined mathematical relation $f$ should be defined in its parametric form first. Once we have this type of function relating the independent variables to dependent variables, the problem turns into curve (or in higher dimensions surface) fitting, i.e. finding the optimal parameters that provide the best fit among observations and estimations.

In the mathematical sense, once the appropriate model class is determined, inference becomes finding the 'best' model parameters. The inferential modeling strategy followed in this work proceeds in this manner; first the appropriate model class is introduced and then the parameters of the system are found. The goodness of model parameters is their ability to bring results that are similar to observations the system yields under the same conditions. The maximum similarity can be expressed as the minimum error between estimates and observations, in this aspect the best parameters would be the ones minimizing the total error. Different error measures can be used, but the most popular one is L2 error minimization and this method is adopted in this study as well.

Having transformed the problem into a minimization problem, the most critical aspects are the complexity of the selected model and the amount of empirical data to be used in optimization. If we consider the model building problem as a curve fitting problem, the model class determines the specifications on the *shape* of the curve. More flexibility in the choice of curve allows higher approximation accuracy, however increased flexibility would correspond to a broader class of possible models and this would increase the complexity of the problem. The amount of data is another factor regarding the complexity of the optimization problem in consideration. Therefore, at the model class selection stage, the amount of data should be taken into account. If the data set is large, the model complexity should be kept low to compensate the complexity arising from the data amount.

## 2.3 Networks - An Overview

In the classical theory of dynamical systems, the interactions between the variables and system environment is analyzed. In this aspect, networks can be considered as a subclass in which the primary emphasis is on the intervariable effects only. The theoretical background of network theory coincides with the graph theory in mathematics and it has application in a varied range of natural sciences and engineering disciplines including computer science, electronical engineering, physics, biology etc. Furthermore, it has applications in social sciences like economics, and sociology as well.

A network (or a graph) is a set of arcs and nodes. Considering the analogy with dynamical systems, the nodes represent the variables or entities in the system and the arcs represent the interaction between two nodes. In a communication problem the nodes will be the servers and customers in the system and server-customer interactions will be represented by arcs. In a logistics problem, the nodes may represent the demand and supply points of certain goods and the arcs will denote the possible routes of transportation among them; in this case arcs can be attributed with certain capacities restricting the interaction at a certain level. In general, a network can be considered as a map of interactions among entities. Directions should be attributed to relations describing cause-effect relations or inflow-outflows. Further specifications can be displayed on a network as well; the particular type of cause-effect relation can be shown (a plus sign on the arc denoting activation and a minus sign denoting repression) or in the case of flows, capacities of arcs (possibly due to physical limitations) can be introduced. Hence, the networks can be considered as alternative representation schemes for dynamical systems.

With the recent technological developments, various models have been employed in mathematical biology and bioinformatics to describe gene regulatory systems. The gene networks are of particular importance for this study since the results found here about inference of piecewise linear models would be used in a project with priority on functional genomics. The gene network modeling is a promising field offering long term targets of innovations in intervention planning for genetic diseases and drug discovery.

The metabolic functions inside an organism are regulated by genes via protein synthesis mechanism. The regulatory relations in a gene network, which is often described symbolically by interaction graphs, can give idea about the processes occurring within a cellular system. Gene networks are finite oriented graphs where nodes represent the genes involved in the biological system of interest and arrows describe their interactions: a positive (resp. negative) arrow from a gene to another represents an activation (resp. inhibition) of the expression of the latter gene by some product of the former [32]. Thus, estimating the underlying dynamical relations among genes is crucial in understanding the biological phenomena. However, gene networks are large and complex network structures and our knowledge on the interactions within these networks is limited [15]. With the recent advances in microarray technology, huge amounts of gene expression data can be obtained for analysis. The availability of data stimulated the interest in modeling cellular networks and understanding the gene interactions. This provides a strong motivation for deriving new methods or improving current methods used for inferring the regulatory dynamics of gene networks.

One of the widely accepted ideas in gene network modeling is the Boolean definition of state of a gene. Many researchers defines two states for each gene as active or inactive (1 or 0 respectively). This approach is parallel to biochemical facts concerning the metabolic reactions. Moreover, this definition also allows use of logical Boolean functions, such as "AND" or "OR" that control the response of a component to a set of inputs. Further specifications on the characteristics of the network is the basis of difference between gene network models. For example, Kauffman used NK models to model genetic regulation. An NK model defines a system considering of N components with K interactions between them and each component can have any number of states. Weber [41] emphasizes that these models are in themselves neither biological, nor restrictively physical, but essentially mathematical. Considering the dynamical system as a network, K can be interpreted the connectivity and N as the number of nodes. Hence, an advantage of using NK models is the ability of making estimation on the complexity related issues for the problem in consideration. In their work, Perkins et al. [28] reached at an upper bound for the amount of data required to infer the regulatory relations in a randomly generated NK network under certain assumptions.

In the literature, there are different approaches proposed for inferring the regulatory dynamics in gene networks. The Boolean networks, difference equation models and piecewise linear equation models are among the popular frameworks for inferential modeling. For the available models, finding a reliable and efficient inference methodology is a commonly studied problem. [28] presents a method for inferring models of gene expression dynamics by transforming the time derivatives of gene expression values into Boolean states, ON and OFF and deriving logical rules for activation and inhibition relations. [4, 23] suggest methods that employ gene perturbations for understanding causal relationships between genes, [20, 19] use Bayesian networks which model causal relationships between variables based on probabilistic measure. The gene networks can be abstracted by complex system equations. Recently, the approach of approximating the complex system equations with a piecewise linear model has been studied [12, 26]. The piecewise linear formulation is preferred due to existence of inherent switchings in the gene network.

# Chapter 3

# Piecewise Linear Models

## 3.1 Properties of Piecewise Linear Models

A general deterministic system can be described by a functional $F$ that maps the input $x(t)$ as a function of $t$ to an output $y(t)$. According to this definition, the system defined by $F$ is linear if and only if the following condition is satisfied for any two given inputs $x_1(t)$, $x_2(t)$ and respective outputs $y_1(t) = F(x_1(t))$, $y_2(t) = F(x_2(t))$:

$$\alpha_1 y_1(t) + \alpha_2 y_2(t) = F(\alpha_1 x_1(t) + \alpha_2 x_2(t)). \tag{3.1}$$

In the aspect of mathematical modeling, linear systems have favorable properties [25]. For instance, a typical linear dynamical system is given in the equation 3.2 with a linear ordinary differential equation. Unlike most of the nonlinear dynamical systems, this class of systems has analytical closed form solution, which is given in the equation 3.3,

$$\frac{dy}{dt} = My, \tag{3.2}$$

$$y(t) = y_0 \exp{(t - t_0)M}, \qquad \forall \quad y_0, t_0. \tag{3.3}$$

Linearity offers advantages in input-output analysis as well. The behavior of the response for a complex input can be analyzed by reducing the input as a linear combination of simpler known ones. This representation of input would provide better understanding for the output in the linear case since the output would also be a linear combination of responses to known inputs. Due to their analytical advantages, the linear models are commonly used for approximating nonlinear systems by linearization.

Nonlinearity is a widely observed behavior in various real world dynamical systems and it appears in many different ways. Without using any approximations or simplifications, a very limited range of nonlinear dynamics can be modeled due to immense complexity. The utilization of these simplificatory methods or approximations is a critical issue since upon application of these methods, the abstraction may fail to represent the phenomena with the desired level of similarity. The suitability of an approximation method would depend mostly on the nature of the phenomena. Thus, a characterization of nonlinearity based on the way of violation of linear intuition can provide guidance for the selection of certain model features, possible approximations etc. Note that, the concept of linearity (or linear evolution) refers to any sort of continuous evolution displaying the characteristics of linear ordinary differential equations, i.e. the relation between derivatives and observed values of variables at a given instance being linear.

Pearson identifies certain qualitative features for the system in terms of input responses and considers the occurrence of these features as an implication of nonlinearity. He suggested a characterization for system's nonlinear behavior based on the level of nonlinearity [27]. The nonlinear behavior is classified into three classes, namely **mildly nonlinear behavior**, **strongly nonlinear behavior** and **intermediate nonlinearity**. In this way, not only the nonlinearity in the system nature is detected empirically but also the feasibility of a probable approximation or simplification method can be measured. Pearson's study is focused on control problem. According to him, the four important measures of model utility are, approximation accuracy, physical interpretation, suitability for control and ease of development. The first two are the common objectives of any modeling problem, thus the idea of understanding the level of nonlinearity can be useful in the existence of nonlinearity for other modeling problems. The suggested method of classification is utilized for modeling issues in control applications; certain modeling features -e.g. special feedback, feedforward structures- and model subclasses are assigned to different levels of nonlinear behavior. In a more general aspect, the level of nonlinearity can give idea about the adequacy of replacing the nonlinear system with a substitute which is sufficiently simplificatory to work on.

It is mentioned that the linearization is used for modeling nonlinear systems to preserve the analytical advantages offered by linearity in the model. However, it might be improper to approximate a nonlinear system with a linear model if the level of nonlinearity in the observed behavior is high, i.e. the accuracy of a linear model built for strongly nonlinear behavior would be less than the case of a mildly nonlinear model.

Piecewise linear models have been proposed for modeling nonlinear systems that cannot be approximated accurately by linear models. A system is piecewise linear if its state space $P$ can be partitioned into disjoint subspaces ($P = P_1 \cup P_2 \cup ... \cup P_n$) such that linearity holds within each subspace. A piecewise linear dynamical system can be represented by the switching differential equations as shown in equation 3.4:

$$\begin{aligned} \frac{dy}{dt} &= M_{s(t)}y + b_{s(t)}, \quad where \\ s(t) &= s_i \quad if \quad y(t) \in P_i. \end{aligned} \tag{3.4}$$

In the literature, the idea of using piecewise linearity have been considered for different problems involving nonlinear dynamical systems [30]. Rewienski studied trajectory piecewise linear approach to model order reduction problem of nonlinear dynamical systems. In that study, he points out that the largest group of model order reduction algorithms applies to linear systems, or more precisely linear time-invariant (LTI) systems. The advantage of linearity can partially be preserved in nonlinear dynamical systems with a piecewise-linear approach and this fact is introduced in the explanation of the motivation of using trajectory piecewise linearity for model order reduction problem in nonlinear systems.

The rationale of using piecewise linear approximations for representing the nonlinear dynamics of regulatory systems exhibiting threshold phenomena is the suitability of piecewise linear approximations for this particular nonlinear behavior. Considering the mentioned critical measures, approximation accuracy and physical interpretation, the piecewise linear approximations are satisfactory. The approximation accuracy of piecewise linear models can be adjusted depending on the desired level of accuracy and the parallelity of switching points in piecewise linear models and threshold regulated switching in discussed dynamical systems provides a basis for physical interpretation. Piecewise linear models are also favorable in the aspect of qualitative behavior matching. A satisfactory model should be capable of exhibiting the essential features of the real world system. In that context, piecewise linear models with single threshold per variable were investigated and many features supported by them was demonstrated [26, 11]. Thus, we know that they can model systems exhibiting fixed point stable, periodic, quasiperiodic, chaotic, multistationary behavior, computation etc. Secondly, piecewise linear models were studied and in the literature there exist known methodologies for analysis and synthesis of them [36]. Therefore, piecewise linear approximations of switching networks can be used for applications like long term forecasting, intervention planning and control wherever possible.

## 3.2 Threshold Phenomena

The regulatory relations in dynamical systems can be interpreted as cause-effect or action-reaction relationships among the system elements. The evolution of the affecting element(s) in time would be a factor determining the evolution of the responding element. A classification of the type of relation between affecting and responding elements in the system can be useful at this point. The first type of relation will be referred in this work as *homogeneous*, referring to the existence of response from the responding element to every change in the affecting element at every level. For example, in a predator-prey system, any increase or decrease in the population of one of the species would affect the rate of change of both species no matter how small the change is. In general, systems of differential equations are appropriate for modeling homogeneous relations.

On the other hand, in certain cases, the response to the change in the affecting element is observed just when a certain threshold value is attained. Thus the response is not homogeneously observed at every level, it takes place after the certain value is reached. The formation of action potential on the membrane of a nerve is an example for the case that the response requires a certain level of electrical charge difference. Prior to any stimuli, the nerve membrane is at resting potential. When the excitatory stimulus is received, the sodium and potassium ions are carried by the active transport, this continuous process is named as depolarization. As the depolarization carries on, no response is given until the threshold for action potential is attained. When the depolarization reaches this level, the action potential is formed and carried through the nerve cell. If the excitatory stimulus is not powerful enough to reach to the depolarization threshold, no message will be carried through the nerve cell. In this example, the occurrence of the response at the threshold level is similar to an ON-OFF switch in an electrical circuit which allows or blocks the current. The nerve cell example demonstrates a particular class of threshold phenomena where the threshold decides between response and no response. In a different case, the threshold of affecting variable can be considered as a switch that changes the characteristic of the response of responding variable.

Threshold phenomenon is observed in various dynamical systems and its relation with switching systems has been studied in the literature [6, 39]. There are different dynamical systems in nature that the physical, chemical or biological occurrences display this characteristic. The switching hybrid systems -hybrid systems governed via switching differential equations- are appropriate for modeling those phenomena.

## 3.3 Suggested Piecewise Linear Model

In section 2.2, the possible objectives in the studies about understanding the nature of dynamical systems were introduced; the aim should be providing a theoretical basis for the observations or creating a model and infer its parameters or predicting the future states of the system for the given conditions. The focus of this study is the inference problem. Some of the challenging nonlinear dynamical system inference problems, including but not limited with the gene regulatory networks, exhibit a switching nature due to the threshold phenomenon. In other words they are typical hybrid systems and governed by switching differential equations. Dynamical properties of exemplary networks and their simplified versions are studied as abstract gene network models [11]. The idea of piecewise linear models and inference of locally linear system parameters has been studied in gene network literature [15, 1] and these ideas are adopted in formulation of suggested model. Besides, some dynamical systems which do not fit to global models expressed by elementary mathematical expressions can be approximated by fitting each portion of the model locally to a simple expression. Taking all these into consideration, this work aims to propose an inference method nonlinear dynamical systems displaying switching property regulated by threshold phenomena by using piecewise linear approximations.

Nonlinear behavior is observed in many systems in different ways. While using more complex models that would capture the nonlinear behavior of these systems, there is always a trade off between complexity and model fit quality. Piecewise linear formulation can approximate complex systems with an adjustable approximation accuracy. Moreover, the advantages of linear systems can be preserved with this formulation [26]. Hence, the piecewise linear models have favorable properties in the aspect of this trade off. In this study, we focused on inferring regulatory relations in linear switched hybrid systems which are governed by threshold phenomenon. The specifications of these systems can be abstracted as follows:

$$\dot{x}_t = M_{s(x_t)}(x_t) + b_{s(x_t)}. \tag{3.5}$$

Here, the parameters $M$ and $b$ are determined by the system state at time $t$ and $x$ denotes the continuous variable(s). Having introduced the state dependence of piecewise linear model, we should explain what the states are. As mentioned previously, we focused on systems in which the regulatory relations are explained by a single threshold per variable. The state of the system $s(x_t)$ in the suggested model is a vector formed by 0 or 1 values where each 0 means that the corresponding variable is below its threshold and 1 means the corresponding variable is above its threshold at that instant. It is obvious that, there are $2^n$ states for a system composed of $n$ variables. With this definition, a state transition can be interpreted as the change in a single element of the Boolean sequence that represents the state. Let $s(t)$ be the state of the system at time $t$ and $B_i(t)$ be the Boolean state of the $i^{th}$ variable ($var_i$) in the network. If $var_i$ exceeds or falls below the threshold in the time interval $[t-1, t)$, the state transition can be written explicitly as follows:

$$\begin{aligned} s(t) = (B_1(t), B_2(t), .., B_i(t), .., B_n(t)), &\quad where \\ B_j(t) = B_j(t-1), &\qquad \forall \quad j \neq i \\ B_j(t) = 1 - B_j(t-1), &\quad \forall \quad j = i. \end{aligned} \tag{3.6}$$

In this formulation, a state transition triggered by the threshold crossing of a variable in the system would be observed as a switching in the activity of other variable(s) in the system. So it is assumed that, every switching occurrence in the evolution of a variable is due to another one crossing the threshold, which we call activator or repressor depending on the type of effect. Furthermore, it is assumed that this regulation relation is always the same for the same transition occurred at different times. The repeatitivity of the identical regulatory relation with same outcome enables characterization and identification of switching occurrences from empirical data. In the empirical approach, it is a commonly used way to group likely occurrences together based on some measure of similarity and compare the conditions at these instances to find some clue about the possible reasons of the observed phenomena. That is, for a repeated event, certain parameters are recorded to keep track of the conditions and the existence of the same condition at every occurrence of the event in consideration is a statistical evidence supporting a causal relationship between the event and the observed condition. This is quite similar to accepting that smoking is one of the factors leading to cancer based on the observations on smoking habits of cancer patients. At this point, it should be noted that likelihood based statistical evaluations are necessary to differentiate between causal relationships and accidental generalizations.

The state space partitioning defined by the thresholds is discussed in section 3.2. Geometrically, we assume that state transitions are allowed between neighboring subspaces, i.e. no more than one threshold crossing is possible in the time interval [t-1,t). The possibility of two variables to attain the threshold value of each at exactly the same time is very close to 0 and we can avoid contradictory observations by increasing the sampling frequency in the experiment design. Hence this assumption is also reasonable. Following remarks can be made for these assumptions:

1. No variables regulate itself by a threshold mechanism; when a variable crosses the threshold, a switching occurs in another.
2. The same threshold crossing always causes the same switching.
3. A direct transition is not allowed between two states if there are more than one different elements in the sequences. See Figure 1 for the possible transitions in a 3 variable network.

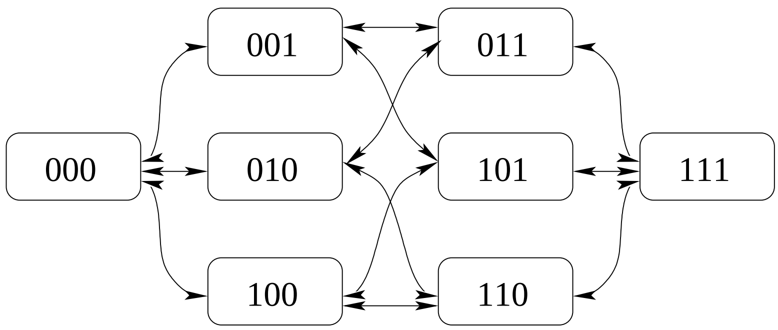


Figure 3.1: Possible state transitions for 3 variable network

In the figure, all possible state transitions are shown for a 3-variables single threshold system. Double-sided arrows represent the transitions could occur in both directions. A transition between states (000) and (110) is not depicted because for such a transition to occur, $var_1$ and $var_2$ should exceed the threshold at the same instant and this contradicts with the third assumption. However, that kind of a state change may occur indirectly in two steps.

The self regulation of variables is not allowed because it would result in zeno effect, infinitely many transitions in a finite time with the current formulation which suggests single threshold for each variable. However, this problem can be prevented by introducing refractory periods, the minimum time delays between consecutive state transitions. (See [25] for further information on zeno effect and refractory periods).

# Chapter 4

# Inference Algorithm

## 4.1 Outline of the Algorithm

Up to this point, the model selection process and the parameter inference problem has been explained in detail. Regarding the specifications of the systems in consideration, the method we propose can be summarized in the following steps:

1. **Detection of state transition instances:** The observed dynamics is approximated by a piecewise linear model. Once we detect the state transition instances, we can break observations from the piecewise linear model into phases in which linearity is preserved.
2. **Inferring regulatory relations:** By observing the similarities at state transition instances, the regulatory relations (in terms of activation and repression) can be inferred according to the assumptions on system nature.
3. **Inferring system parameters:** With the assumption of piecewise linearity, we interpret the system behavior as consecutive linear behaviors with different parameters. Once we know the regulation mechanism of state transitions, the parameters of the *observed* states can be inferred easily.

Detection of the state transition instances is the first step in the inference method that is proposed here. In section 3.3, the rationale of tracking the occurrence of same -or at least similar- events to predict the underlying factor triggering this particular occasion is explained. In the suggested framework, only the interactions among the variables are taken into consideration. To sum up, the triggering mechanism for a switching is assumed to be a state transition or in other words a threshold crossing.

Table 4.1: Values of All Variables in the System at Switching Times First Variable

| Switch Time | Var1 | Var2 | Var3 | Var4 |
|---|---|---|---|---|
| t=t1 | 0.612 | 0.388 | 0.498 | 0.232 |
| t=t2 | 0.471 | 0.212 | 0.501 | 0.272 |
| t=t3 | 0.154 | 0.746 | 0.5 | 0.621 |
| t=t4 | 0.522 | 0.343 | 0.506 | 0.506 |
| t=t5 | 0.312 | 0.156 | 0.503 | 0.850 |
| t=t6 | 0.014 | 0.545 | 0.498 | 0.214 |

Thus, if the same type switchings observed in the evolution of the a given variable can be identified, then there will be the chance of doing a comparative analysis for investigating the regulatory variable(s) controlling the evolution of that given variable. Therefore, after the same type of switching instances are located on the trajectory of a given variable, the values of the other variables should be compared at these instances. For the sake of consistency, the same switching observed in different times should be initiated by the same factor. Hence, if the values of a variable measured in two distinct switching instances are significantly different, then this variable cannot have a regulatory effect on this variable. If any variable takes very close values at all occurrences of the same type of transition for a given variable, then it is reasonable to assume that it is a regulator variable.

Consider the case demonstrated in the following table. The tabular data denotes the values of all four variables in the hypothetical system recorded at the switching instances of variable1. If one of the given variables initiate the switching in first variable by a threshold crossing, its value should be almost the same in all six switching instances. When the values in the columns are checked, variables 1,2 and 4 takes quite dissimilar values at state transition instances whereas variable3 takes values very close to 0.5. Under the assumptions given in section 3.3, we can conclude that the third variable triggers the switching in the first variable. Moreover, as the regulatory relations are defined in terms of threshold crossings, we can conclude that the threshold of variable 3 triggering the switching in variable1 is 0.5.

Having found the regulatory variable, it is quite easy to determine the type of effect, whether it is activation or repression. In the hypothetical example introduced above, the switching times are the instances when the third variable exceeds or falls below its threshold. When the derivative estimates at those instances is available for third variable, it can be stated that the third variable exceeded (corr. fell below) the threshold if the derivative of the third variable around the switching instance is positive (corr. negative). Then, we conclude that the third variable activates (corr. represses) the first one if the first variable undergoes an upwards (corr. downwards) trend when the third variable exceeds the threshold and a downwards (corr. upwards) trend when the third variable fells below the threshold.

The operations carried out for detection of the regulatory relation affecting variable1 can be repeated for all variables in the network and when all the regulatory relations are identified in this way, the behavior of the system at any given time can be determined by the thresholds. Hence, the observations of the system over time which could be approximated by a piecewise linear model can be split into separate phases in which the evolution can be approximated by a linear relation. This is the third step in the algorithm outlined above. Once the linear parts of the piecewise linear model are distinguished, it is easy to infer the parameters of the linear evolution by regression.

The main frame of the inference algorithm is explained up to here but there are still remaining challenges in the achievement of these steps. In the first step, the detection of state transitions from empirical data is required. In the second step, the comparisons will be carried out for all variables and there is still ambiguity in the notion of values at certain time points to be *very close* to each other. In the remaining parts of this section, the techniques employed for overcoming these problems will be introduced and the adequacy of these techniques for the inference problem in consideration will be discussed. The programs used for each task and the links among them are explained. The validity of the techniques employed in these programs to achieve certain tasks is also discussed to validate the conformity of the programs with the tasks they are involved.

## 4.2 Estimating Continuous Dynamics from Discrete Observations

Inferential modeling is based on building up a model that would display desired characteristics of a system. It is possible to track and store all information in discrete event systems over a finite time period since there are finitely many events (*jumps*) to observe. Once the type of relation governing the dynamics of the discrete event system is known, the parameters of the discrete system can be inferred. On the other hand, the continuous events (*flows*) can partially be observed since the observations on the desired measures of the system can be done on a discrete time basis and there is limited storage for the data gathered from the measurements. The continuous evolution should be approximated by interpolation through discrete observations.

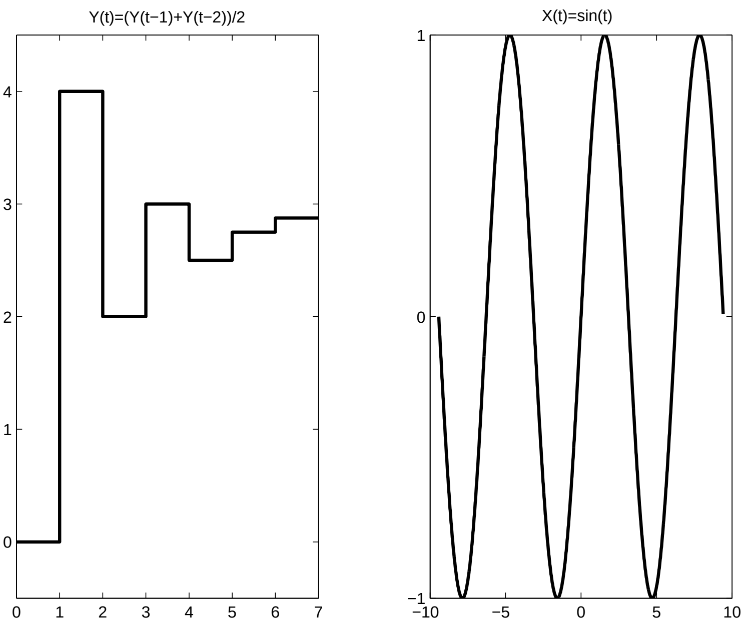


Figure 4.1: Discrete vs. Continuous Events

The information content of the discrete event shown in the figure 4.1 is the data containing the sizes of the equally-spaced jumps. Using these information, the model parameters explaining these jumps can be inferred if it is known that the value of the discrete variable at a given time $t$ depends on the values at time $t-1$ and $t-2$. On the other hand, the continuous event in the figure cannot be stored as a series of discrete observations with full accuracy; it could be discretized and then stored as a discrete series.

The empirical data obtained from experiments and observations on the system would be used to measure the concordance of the abstraction with the observed system dynamics. Since the procedure is data-driven, the quality and reliability of data plays a critical role in the success of the procedure. In other words, even if an appropriate inference methodology is successfully implemented, if the data is not *good* enough the abstraction will not be capable of representing system's behavior. The reliability of data is related to the accuracy of the measurements. The measurements should be unbiased and the noise arising from experiment setup should be kept at a reasonable level. The term quality refers to capability of the discrete data in representing the continuous evolution. The continuous evolution in the system is approximated with an interpolation through the discrete data. The discrete data is generally recorded with successive measurements repeated with a fixed frequency. At this stage, attention should be paid in the selection of the measurement frequency. A continuous signal could become distorted due to poor sampling, that is the interpolation of the discrete data gives a signal displaying different characteristics from the original one.This problem is known as aliasing in the signal processing terminology.

In the section 3.3, it is mentioned that the continuous evolution within an invariant subspace (i.e. a given system state) is determined by a linear first order differential equation (See equation 3.5). To infer the unknown parameters of the ODE governing the evolution in an invariant subspace, linear regression can be employed on the discrete data that belongs to the given subspace since there is a linear relation between the observed values($x_t$) and time derivatives($\dot{x}_t$) of the variables in the system. Although the observed values are known, the time derivatives for the sampling instances should be estimated from the observed values prior to the inference of the parameters. When these estimates are found, the system of ODEs would be transferred into an overdetermined system of linear equations and the best-fit parameters of this system could be found by regression.

The importance of estimating the switching times accurately has been discussed in the previous sections. First of all, the derivative estimates are required for both discretizing the linear ODE system. Moreover, derivative estimates can also be employed for detecting the switchings from the empirical observations. The details of the method used in switching detection is explained hereafter.

## 4.3 Derivative Estimation

Up to here, the importance of estimating time derivatives for system discretization have been discussed. After the system is converted into an overdetermined linear system, if the boundaries between different linear behaviors can be detected efficiently, the piecewise linear data set can be split into linear pieces and then the parameter inference problem can be reduced into a simple linear regression problem.

As its name suggests, the piecewise linear behavior can be recognized by the existence of different linear behaviors in different phases of the evolution. If two consecutive phases are governed by two systems ODEs with different parameters, this would be observed as a change in the pattern on the trajectory. The different patterns are expected to result in the occurrence of non-smoothness at the transition instance between two consecutive phases. Thus, for a piecewise linear model, detecting the nonsmoothness on the trajectory is equivalent to detecting the switching in the evolution of one of the variables.

From calculus, we know that a function is smooth at a point if the left- and right-hand derivatives are equal for the function at that point. Thus, the points of nonsmoothness on the trajectory of function values would be transformed into jumps after differentiation. Jump detection problem has been studied deeply in signal processing [21, 42, 40]. By transforming the problem from nonsmoothness identification on observed values to jump detection on derivatives, we take advantage of the possibility to adopt the methodology used for similar problems in a different field. The details about jump detection will be discussed further in the next section.

Up to now, the point of estimating the time derivatives from empirical observations has been discussed. At the first level, the derivative estimates are used in reducing the complexity of inferring the piecewise linear model by providing us a tool for splitting the system into linear parts. Afterwards, the derivatives are used in the discretization of ODEs. Apparently, the derivative estimates play a critical role in the inference method proposed in this work and that is a strong motivation to discuss the key concepts about numerical differentiation in this part.

### 4.3.1 Numerical Differentiation

For a given smooth function, integral of the function over an interval is found by the limit of a sum of infinitesimal quantities and the derivative at a given point is obtained by the limit of a ratio of infinitesimal quantities. The elementary methods of numerical differentiation and integration are based on these definitions and they are quite alike. Let $f$ be a continuous function whose values are known only at $n+1$ time points in the interval $[a, b]$. Then, for the function $f$, the integral and derivative estimates can be numerically approximated as follows:

$$\begin{aligned} \int_a^b f(t)dt &\approx \sum_{i=1}^{n} (f(t_i) * (t_i - t_{i-1})), \\ \frac{df}{dt}(t_i) &\approx \frac{f(t_i) - f(t_{i-1})}{t_i - t_{i-1}}. \end{aligned} \tag{4.1}$$

These definitions are not unique but any estimate derived from the limit definitions would have a similar form. Here, numerical differentiation is analyzed on a comparative basis with numerical integration to explain the noise sensitivity of derivative estimation. In the integral approximation in equation 4.2, the empirical values are summed up which means the possible error is the term found by the noise factors multiplied with the differences. Adding noise terms allows cancellation and it suppresses the effect of noise automatically. On the other hand, in the derivative estimation with difference equations, the noise term changes the nominator of a quotient whose denominator can be small as well. So, if the differences are small, the effect of noise would be amplified and this will harm the reliability of derivative estimates. Even if the noise is not amplified by the division, the advantage of cancellation occurred in integration does not work for difference equations in general. This drawback of difference equations motivated use of other techniques for derivative estimation.

The notion of noise is frequently emphasized throughout this work since existence of noise is inevitable in almost all empirical data and to generate a robust algorithm, the effects of noise should be suppressed. For example, gene networks are widely studied systems which display the switching behavior and [20] points out that in realistic situations, gene expression measurements are noisy. This particular example points out the importance of robustness for an inference method. A similar approach employing piecewise linear model formulation for inference of gene networks suggested difference equations for discretizing the system of ODEs [1]. Unlike numerical integration, estimating the derivative of a function at a discrete set of points with the formula given in equation 4.2 is a sensitive problem as small perturbations can cause large changes. Thus, the inherent noise in gene expression data makes finite difference approximation unfavorable in the aspect of robustness. Instead of using difference equations, fitting some continuous function to the given discrete data and then differentiate this function to estimate the derivative is expected to yield more robust results.

At this point, the performance of two derivative estimation methods, difference equations and curve fitting will be compared on examplary discrete data. The first graph in the figure 4.2 depicts a continuous spline defined over the time interval $[-\Pi, 2]$ with continuous first order derivatives (it composed of sine function on $[-\Pi, 0]$, identity function on $[0, 1]$ and exponential function $e^{x-1}$ on $[0, 1]$). The discrete data obtained from observations on this system are simulated by sampling the continuous function every 0.01 seconds and adding random noise on it. The second graph depicts that empirical data.

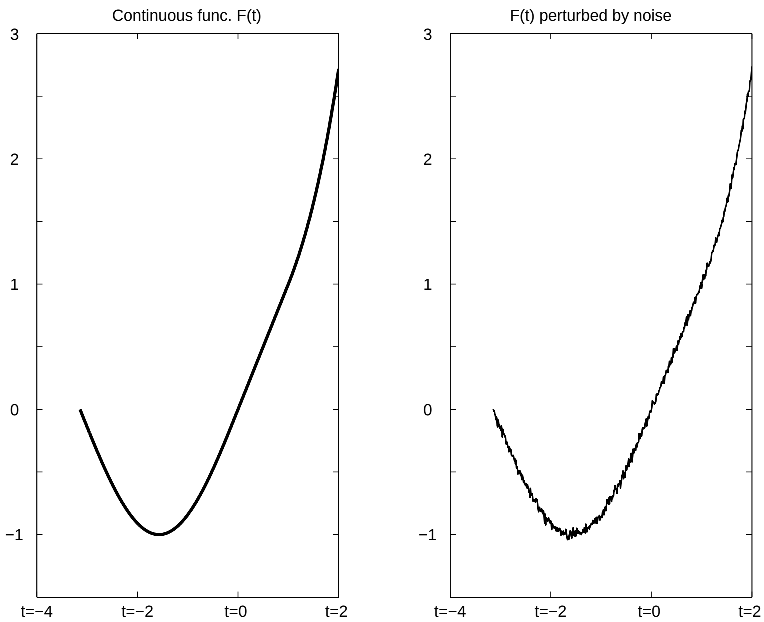


Figure 4.2: The Actual Continuous Pattern and Simulated Empirical Discrete Data

The derivative of the original spline can be found by differentiating the continuous functions composing the spline analytically. The difference equation estimates are obtained by the equation 4.2. The Matlab code used for the evaluation of derivative estimates of local curve fitting method can be seen in Appendix 1. The graphs in the figure 4.3 belong to actual derivatives, local curve fitting estimates and difference equation estimates respectively. It is clearly seen that local curve fitting method significantly outperform difference equations in terms of approximating accuracy.

For numerical derivative estimation, curve fitting to the discrete data is recommended in the literature also [17]. The theoretical frame of this method is simple, the unknown continuous evolution function is approximated by a function whose derivatives can be found analytically. Finding the best fit parameters of the approximating function is an optimization problem whose complexity is determined by the type of approximating function and the number of discrete data along which the curve fitting will be done. After evaluating the approximating function, the derivatives of this function at given time points can be used as estimates of the derivative of the unknown evolution function. The strongest motivation of using curve fitting is finding robust derivative estimates. The perturbations arising from noise in empirical data are smoothened by curve fitting and this yields more robust derivative estimates.

Although the derivative estimation with curve fitting method gives more robust

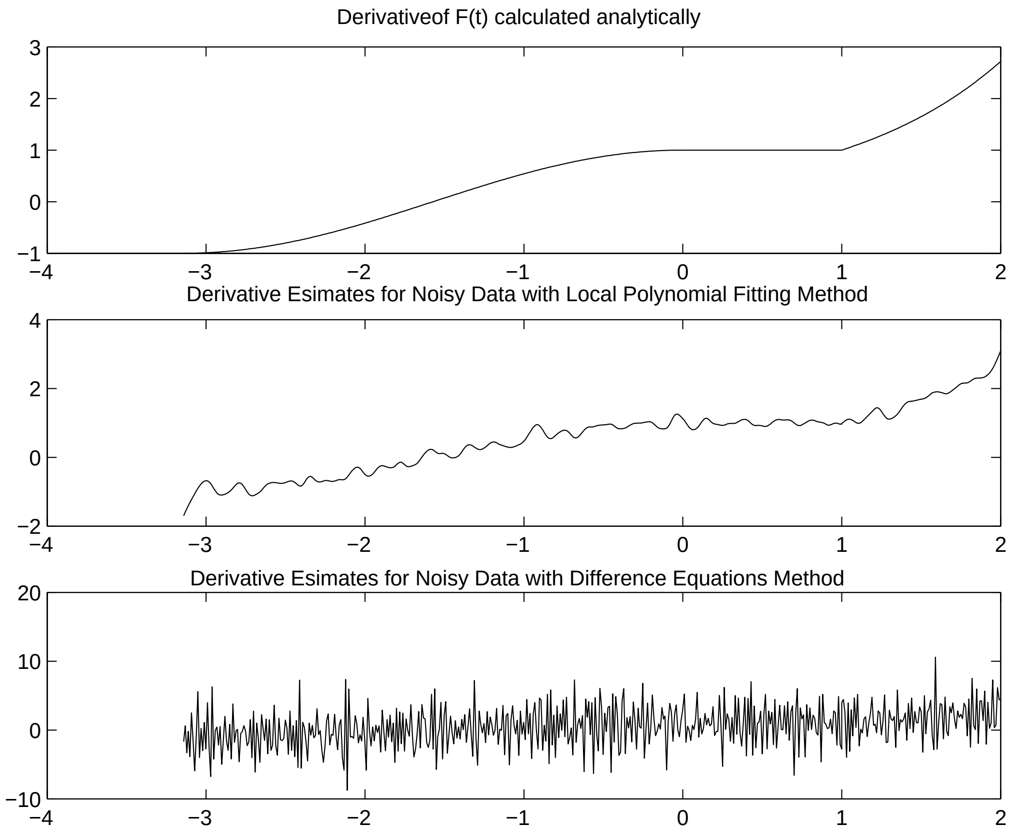


Figure 4.3: The Actual Derivative Values and Derivative Estimates Obtained by Two Different Methods

results, it has the drawback of increased computational complexity in comparison with difference equations, which means an increase in the cost of computation. To have a robust inference algorithm with a reasonable computational cost, efficient ways of using curve fitting should be found to keep this complexity at a reasonable level.

The complexity of curve fitting process heavily depends on the complexity of the function to be fitted and the number of discrete points along which the curve fitting will be done. In many cases, thousands of discrete time observations are made to understand the continuous phenomena. Fitting a single function to all the data and trying to find the optimal parameters for this function would be inefficient. Instead, the curve fitting procedure could be done repeatedly on smaller intervals including reasonable number of data points. For example, consider a case where the number of observations, $m = 1000$, and the derivative estimates at these 1000 data points are required. In this case, local curve fitting could be applied by fitting approximating functions to small data sets, e.g. consisting of 20 observations each. In this case, 50 different curves would be fitted to 50 different groups of data and the derivative estimates of the points in each group can be approximated with the derivatives of corresponding approximating function. This idea is closely related with the notion of windowing in signal processing. A window function is a function that vanishes outside of some chosen interval. If the value of windowing function is taken 1 inside the interval, the multiplication of the window function with any discrete or continuous function would be equivalent to restricting the domain of the function on the windowing interval. The small groups for local fits can be obtained with this type of multiplication by choosing the size of the windowing interval (*window-size*) such that desired number of observations would stay inside. By shifting the interval of local fit along the entire discrete empirical data set, a family of windowing functions would be obtained and the data can be partitioned into small groups with these functions.

The rationale of using local curve fitting for robust derivative estimation has been discussed up to this point. Proper attention should be paid on the further specifications on the choice of approximating function and the window-size. From this point on, the focus of discussion is shifted to the derivative estimation in piecewise linear models or switching dynamical systems. The issues that should be taken into consideration for choice of approximating function and determination of window-size for this particular problems is explained thoroughly.

1. **Appropriate function type for curve fitting:** As mentioned above, the

motive for fitting a function to discrete data is to estimate derivatives by differentiating the approximating function. Finding the parameters of the approximating function is an optimization problem and efficiency of the algorithm in finding these parameters is very important since this problem would be solved over and over for each window. After finding the approximating function, its derivatives should be calculated. Taking these aspects into consideration, the approximating function should be chosen so that the computation cost of evaluating the function values and its derivatives should be low. Due to analytical advantages in differentiation and low cost of computation, polynomials are commonly suggested as the function family to be fitted [17, 7, 33, 13]. For a polynomial of degree $n$, finding the coefficients of the derivative function, which is again a polynomial of degree $n-1$ at most, requires n multiplications. For the same polynomial, the computation of function values at a given point requires $n$ additions and $n$ multiplications. Polynomials outperform the function families containing exponential and/or sinusoidal terms with respect to these measures.

The family of polynomials are considered to be appropriate for curve fitting. The final specification of approximating function is the degree of the polynomial. For better interpretation of the importance of polynomial's degree, some particular choices should better be discussed. As an extreme case, if a first degree polynomial is selected as fitting function, the problem turns into best line approximation. Since the resulting approximation will be a line, it will yield constant derivatives on the chosen window. Since the continuous evolution is assumed to be governed by switching differential equations, the linear approximation to this function is too simplistic; especially in the windows containing a switching point. This kind of oversimplification may result in inability in finding the jump occurrences in time derivatives which indicate the switching instances.

On the other hand, if we choose the maximum degree, which is the number of data points in the window, the problem would be transformed into interpolation. Full degree polynomial interpolation is not favorable in the interpolation methodology since it is likely to yield undesired oscillatory behavior in the interpolant. Moreover, the first order derivatives would be sensitive to perturbations in that case [17]. Hence polynomial interpolation is not suitable for numerical derivative estimation with noisy data. Thus, depending on the number of points to which the polynomial would be fitted, an appropriate degree should be selected in between. When the degree increases, the noise sensitivity would increase as well. On the other hand, when the degree is too low, the best fit polynomial would be a loose approximation. This trade off between approximation accuracy and robustness should be taken into consideration in the determination of degree.

2. **The window of curve fitting:** The amount of empirical data required for a realistic and accurate model inference is quite high. Fitting a polynomial to huge amounts of data points would be very inefficient computationally or maybe impossible. Thus, instead of fitting a single curve to the entire data set, local polynomial fitting should be employed. This approach is quite similar to use of splines in the polynomial interpolation through large number of data points. Whether the problem is interpolation or curve fitting, use of piecewise polynomials offers lower degree polynomials which decrease the oscillations and increase the accuracy significantly.

   The most important parameter of the windowing function is the window-size. In our formulation, the window-size refers to the number of discrete data points to which a local polynomial would be fitted. In the next chapter, a hypothetical switching hybrid system example is simulated and the inference methodology is implemented. In that part, the degree of the polynomial and window-size are determined experimentally. Both parameters are manipulated and the resulting plots for derivative estimates and curves fitted to discrete data are analyzed in the aspects of fitting quality to discrete data and the noise sensitivity.

   The shifting of window along the discrete data ensures estimating the derivatives at all data points. If the shift is chosen to be equal to window size, one derivative estimate for each point would be obtained. However, if only at each step, more than one derivative estimates would be found. Assume the window size $w = 5$. If the window is shifted by five data points at each step, the first polynomial would be fitted to data points 1 to 5 and the derivative estimate for data point 5 would be found by the approximating polynomial in this window. On the other hand, if the window is shifted by one data point, the first five windows would include the data points 1 to 5, 2 to 6, 3 to 7, 4 to 8 and 5 to 9 respectively. All first five windows would include the fifth data point, hence five different derivative estimates from each approximating polynomial can be obtained for that point. Averaging the derivative estimates from different windows containing the same point is expected to yield more robust results, therefore this type of window shifting is implemented in the computer algorithms codes written for the proposed inference method.

The cost of computation is an important criteria for the choice between different methods. The cost of derivative estimation can roughly be interpreted in terms of the parameters determining the computational complexity. Let $D$ denote the size of discrete data set, $m$ denote the window-size of local fitting and $n$ denote the degree of locally approximating polynomial. The shifting by one data point in each step requires solving $D-m+1$ polynomial fitting problem. Fitting a polynomial of degree $n$ to $m$ data points requires solving an overdetermined linear system of dimension $m*(n+1)$. For each window, the derivative of the fitting polynomial should be found analytically and the derivative function, which is another polynomial of degree $(n-1)$, should be evaluated at all data points in the window. For a given time point $t$, $m$-many derivative estimates would be found from different windows intersecting at $t$ and a weighted average of these estimates would be used as the derivative estimate at $\dot{x}_t$. Using these information on the iterations, the computational cost of finding the derivatives in terms of elementary operations (additions, multiplications etc.)can be estimated.

There are studies in the literature where local polynomial fitting is employed in estimation problems. Recently, the polynomial fitting approach in nonparametric regression estimation has become popular and this approach is studied by many researchers [7, 33, 13]. Masry points out that local polynomial fitting approach is superior to the Nadaraya Watson estimator in the context of estimating the derivatives of the regression function [24].

## 4.4 Jump Detection

Finding the boundaries between different linear behaviors is the most critical part of the inference algorithm since it is the fundamental tool used for finding intervariable regulatory relations and the thresholds governing these relations. It has been explained that the switchings in the continuous evolution appear as jumps in the estimated time derivatives. Thus, detecting the jumps is used for finding the switchings. A very similar technique is used in data mining for event detection from time series data. The aim of employing event detection is to divide the continuous time observations into different *episodes* separated from each other by certain *events*. Afterwards, the function to be used in curve fitting in the interval between successive change points is decided and model parameters are inferred [16]. In this frame, the switching instances can be considered as the events to be detected and the piecewise linearity can be interpreted as the linearity within the episodes violated with switching events at the boundaries.

Pictorial explanation of jump detection in time derivatives can provide better understanding. Figure 4.4 depicts the synthetic data generated according to the piecewise linear evolution in which each component has repressilatory effect on another ($var_1$ inactivates $var_2$, $var_2$ inactivates $var_3$ and $var_3$ inactivates $var_1$). The repression designated here can be interpreted as the decay of the affected variable after the repressor variable exceeds its threshold; the threshold for all variables is taken as 1 in the simulation. The particular choice of regulatory relations among the variables resulted in successive growth and decay phases in the evolution of all three variables. Figure 4.5 depicts the time derivatives estimated from discrete data via local polynomial fitting.

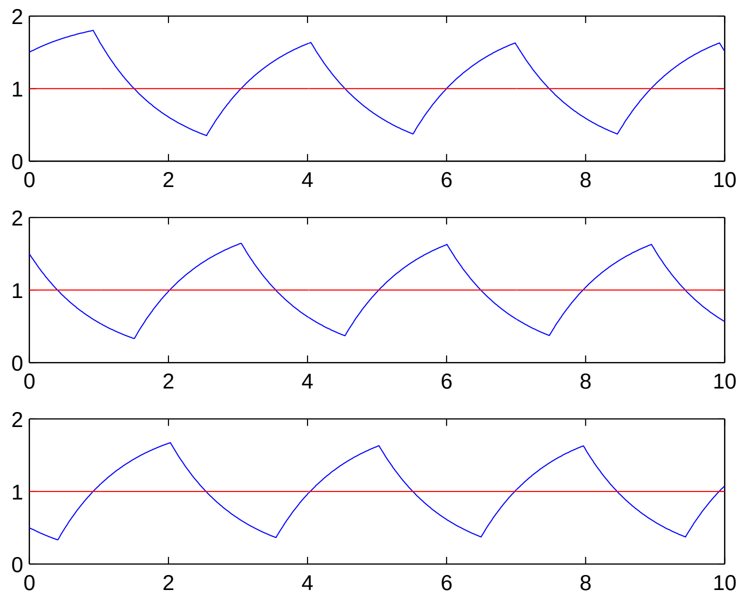


Figure 4.4: Synthetic data for 3-variable network

Finding the switching instances of variables is the key point for inference based on piecewise linear approximation. However, there is no straightforward way of detecting the instances when the evolution is nonsmooth since the empirical observations are available as a discrete time series. At the switching instances of affecting variables,

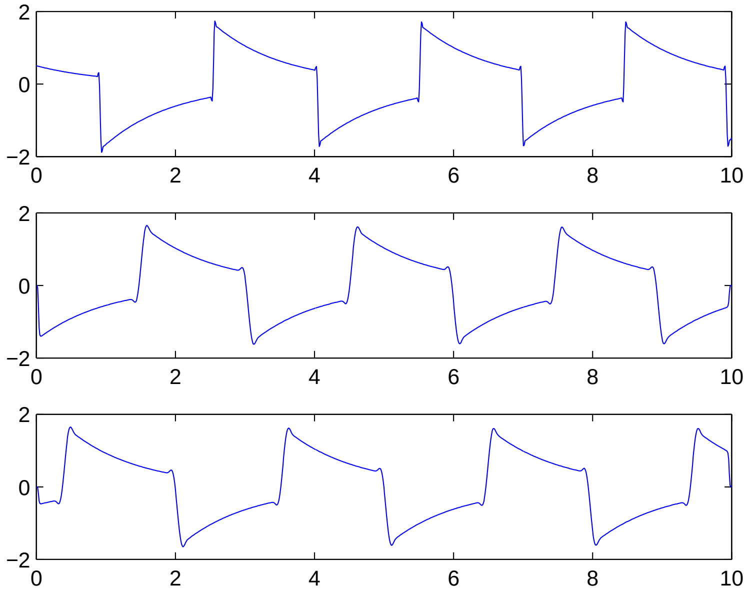


Figure 4.5: Estimated time derivatives of data in Figure 2

the simulated continuous evolution trajectory of the responding variables fail to be smooth as expected. For the time derivatives, those instances corresponds to jumps. From calculus, we know that nonsmoothness of function at a point $p$ corresponds to jump in the first derivative at the same point. Thus, each jump occurrence in the estimated time derivatives indicates a switching provided that the estimates are reliable.

To detect the jumps in the time series, the jump detection techniques studied in signal processing are adopted. Wavelet analysis has been implemented to various problems in the literature including detection of image edges[35] and extraction of objects from complex backgrounds [34] in image processing, tracking-based estimation of support boundaries[5] and sharp cusp detection [40]. In the method implemented here, discrete high pass filters are employed to convert jumps into impulses. The convolution of time derivatives with the appropriate high pass filter yields another time series that have values close to zero at points where the approximation is smooth. At the points of switching instances, the convolution yields a positive or negative impulse depending on the direction of jump. The first graph in 4.6 depicts an exemplary discrete high pass filter that can be used for this type of problem. The second graph belongs to the estimated time derivatives of a variable in a simulated 3 variable repressilatory piecewise linear model. In the last graph, the convolution of derivatives with the suggested filter is given.

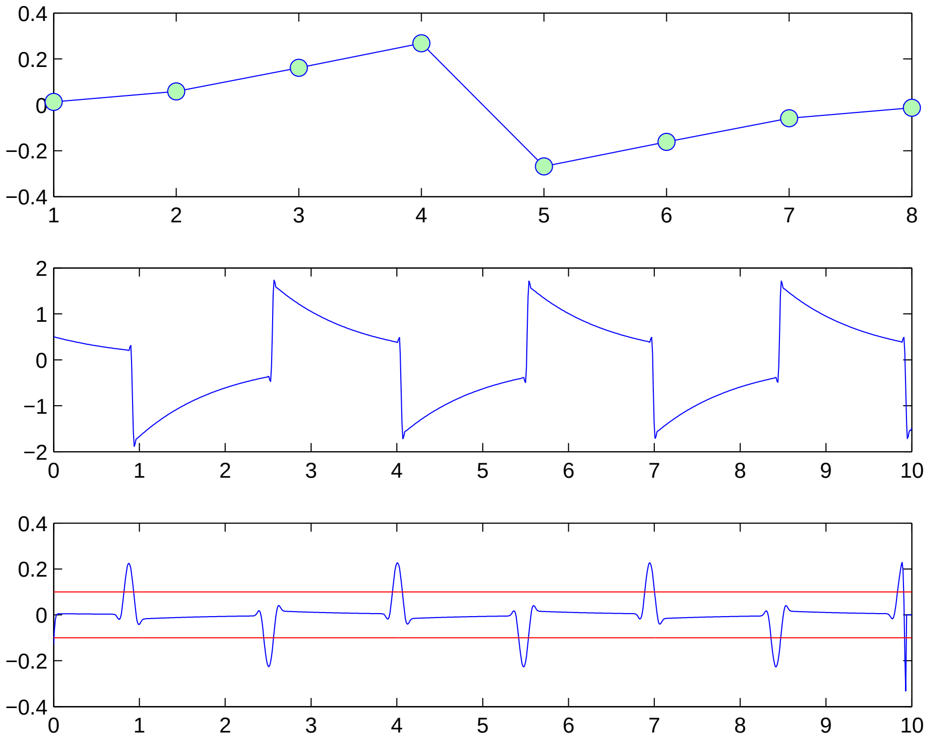


Figure 4.6: Jump detection using convolution with high pass filter

The method of switching detection for a given discrete time-series data can be interpreted as a series of transformations. Replacing the derivatives with observed values ensures the transformation from a piecewise smooth approximation function disturbed by sharp edges to another piecewise smooth function separated with jumps. At the latter step, convolution with a high pass filter allows representation of the series of jumps as an impulse train. The positive (correspondingly negative) impulses appearing in the convolution of derivative estimates and high pass filter, correspond to a down-jump (corr. up-jump) in the derivatives. For the bi-state (ON or OFF) switchings, this up-jump(corr. down-jump) indicates to an OFF-to-ON(corr. ON-to-OFF) switching. Impulse detection from discrete data is an easier problem, the significantly large values (in the absolute sense) can be tracked via comparison with a benchmark. This reference level is depicted as the straight lines in the last graph in figure 4.6.

The importance of robustness of derivative estimates can also be understood better with the interpretation of switching detection along these steps depicted in figure 4.6. If there is noise in the data, this will cause perturbations in the first graph. If these perturbations could not be suppressed in the second step, artificial jumps generated by random noise would occur in the second graph which would make the identification of original jumps corresponding to switching detection harder. At the last step, the clearly interpretable impulse train structure would be replaced by frequent oscillations and the method would possibly fail to detect the switchings in the designated way.

## 4.5 Summary of the Inference Method

The problem studied in this work is inferential modeling of hybrid system evolving according to switching differential equations governed by threshold phenomena. The rationale of piecewise linear approximations has been discussed in chapter 3 and the steps to be followed after the detection of switchings has been discussed.

- **Step 1.***Estimate the derivatives of the responding variable from discrete empirical data.* A Matlab code is developed for estimating derivatives from time series discrete evolution data (See derest.m in Appendix 1). The function has two more input parameters other than discrete data, the window-size and degree of the local approximating polynomial. These two parameters should be set carefully since they determine the approximation accuracy of the locally fitting polynomials and the computational complexity of derivative estimation procedure. In the hypothetical example discussed in this work, these parameters are found experimentally taking these issues into consideration. The function returns the denoised data obtained by fitted polynomials with the derivative estimates and these can be used in the inference of the parameters governing the continuous evolution.

- **Step 2.***Convert the jumps in estimated derivatives into impulses.* A Matlab code is developed for detecting the jumps in time derivatives [See jump-time.m in Appendix 2]. This function evaluates the derivative estimates by calling the function *derest.m* and used a gaussian high pass filter. This filter can be interpreted as the complement of gaussian function and it is derived from gaussian low pass filter by multiplying the second half of the gaussian filter with $-1$ (this filter is depicted in first graph of figure 4.6). The gaussian filter has two input parameters. These values are found experimentally as well. The critical issue about the filter is that the first parameter determining the size of filter should be even for the sake of symmetry of the high pass filter. The convolution of derivative estimates with the high pass filter yield the desired impulse structure.

- **Step 3.***Detect the switching instances.* The impulses correspond to switching occurence. Detection of an impulse from a discrete time series can be achieved by using comparison with a benchmark level (the benchmark is determined experimentally). The benchmark level for the hypothetical example is depicted as the straight lines in last graph of figure 4.6). In this part of the algorithm, the instances at which the values exceed the benchmark level are recorded. In this way, finding the peak of the impulse is not possible in general; an interval around each impulse would be found. Since the impulses are expected to be symmetrical, the midpoint of each interval is taken as the switching instance.

- **Step 4.***Infer the affecting variable and the threshold.* It is explained in section 4.1 that the function values of the affecting variable should be very close to each other at all switching occurences in the responding variable. Having detected the switching instances, the values of all variables can easily be tabulated and the mentioned comperative analysis can be carried to infer the regulatory relation.

- **Step 5.***Infer the all regulatory relations in the system.*The first four steps should be repeated for all variables so that the all the regulatory relations would be inferred and the corresponding thresholds would be found.

- **Step 6.***Infer the parameters of continuous evolution.* When all the thresholds are known, the invariant subspace partition in the hybrid system formalism can be determined. Within each invariant subspace, the continuous evolution is governed by a system of linear ordinary differential eqeuations. The parameters of the discretization of ODEs in each subspace can be inferred from empirical data and these parameters can be used as estimates of the parameters of continuous evolution.

# Chapter 5

# Application of the Method

## 5.1 Application on Simulated Example

The proposed inference method is explained in detail in the previous chapter. Demonstration of the method on exemplary data can be useful to make implementation procedure more clearly understood. In this part of the thesis, the method would be implemented on the discrete data generated by a simulation. The main Matlab functions employed in every step will be explained. For the sub-functions called by main functions in the iterations, the Matlab files in the Appendices should be viewed.

The simulated discrete data belongs to 1000 observations from a 3 variable switching dynamical system evaluated by equally-seperated sampling with a sampling rate 100 samples per unit time. The desired qualitative behavior is created by introducing and the possible state transitions, initial states, initial values and the threshold levels for each variable. It has been discussed that the empirical observations on the dynamical systems are subject to noise; therefore gaussian noise is added to the simulated data to have a more realistic demonstration. The results are depicted in the Figure 5.1.

The detection of switching instances for each variable is the first task. The M-file **jump-time.m** is developed for switching time estimation for a given variable. The first input is the discrete observations for the chosen variable as a row vector. The following input parameters are the *window size of averaging* and *degree of approximating polynomial* mentioned in derivative estimation and these parameters belong to the sub-function derest.m. The length of the high pass filter that will used for transforming jumps into impulses is the third input to be determined by the user accompanied with window size and polynomial degree. The last input parameter for

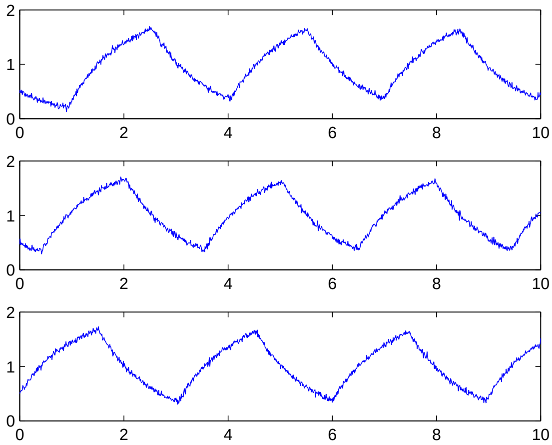


Figure 5.1: Simulated Discrete Data for Switching Dynamical System

**jump-time.m** is the time interval between consecutive observations, which is 0.01 for this problem.

As mentioned previously, the determination of *window size*, *polynomial degree* and *filter length* has been done experimentally. The important point that should be taken into consideration at this step is that window size should be an odd number and filter length should be an even number for the purposes of symmetry. The code **jump-time.m** is designated so that the impulse structure is represented graphically with a figure after each run (an example is given in Figure 5.2). After a few trials with different parameters, these figures should guide the user about how to manipulate each parameter.

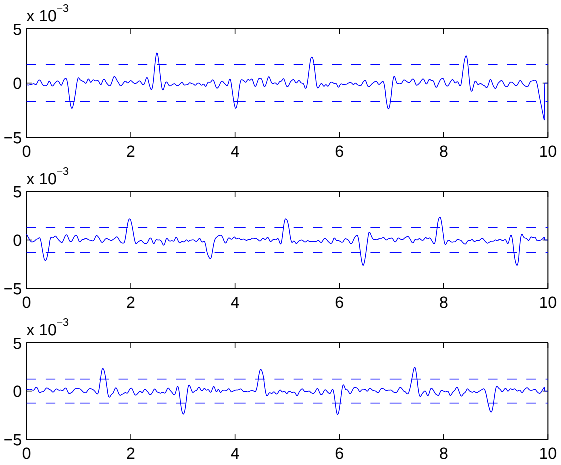


Figure 5.2: The Graph of Impulse Structure for First Variable

The output of the Matlab function does not directly provide the switching occurence times. Each switching occurence is detected in an interval instead (See table of outputs). Consider the values found for the first variable. The first switching is detected in the interval [1.22,1.31]. For locating the exact switching instance, the midpoint of each interval can be used since the impulse structures are expected to be symmetrical around the peak. Using this idea, the switching instances for each variable are obtained.

From this data, the switching instances can be classified as shown in the following table:

Having found the switchings, the regulatory variables and corresponding thresholds can be inferred. This step requires the comparison of values of each variable at the switching instances of the variable whose regulator is to be found. For instance, the values of each variable at the switchings of the first variable, the values of all variables are tabulated in table 5.3. It is previously stated that, the variable which takes close values in all switching times is the regulator. At the switching instances of the first variable, the second variable takes values in a narrow range around 1, hence the second variable is taken as the regulator for the first. The mean of values of second variable is taken as the threshold value. For estimating the type of the regulatory relation, whether it is regression or activation, the derivative estimates of variables should be analyzed around the switching instances. For example, the first switch of variable 1 occured at $t = 0.93$. Right after the switching, the first variable has negative derivatives, indicating the start of a downward trend. In the same

Table 5.1: Output of jump-time function applied to all three variables

| up1 | up2 | up3 | down1 | down2 | down3 |
|---|---|---|---|---|---|
| 0,90 | 0,36 | 2,99 | 2,53 | 1,99 | 1,47 |
| 0,91 | 0,37 | 3,00 | 2,54 | 2,00 | 1,48 |
| 0,92 | 0,38 | 3,01 | 2,55 | 2,01 | 1,49 |
| 0,93 | 0,39 | 3,02 | 2,56 | 2,02 | 1,50 |
| 0,94 | 0,40 | 3,03 | 2,57 | 2,03 | 1,51 |
| 0,95 | 0,41 | 3,04 | 2,58 | 2,04 | 1,52 |
| 0,96 | 0,42 | 3,05 | 2,59 | 2,05 | 1,53 |
| 4,02 | 0,43 | 3,06 | 5,50 | 2,06 | 1,54 |
| 4,03 | 0,44 | 3,07 | 5,51 | 2,07 | 1,55 |
| 4,04 | 0,45 | 3,08 | 5,52 | 2,08 | 1,56 |
| 4,05 | 3,50 | 3,09 | 5,53 | 2,09 | 4,49 |
| 4,06 | 3,51 | 5,96 | 5,54 | 4,99 | 4,50 |
| 4,07 | 3,52 | 5,97 | 5,55 | 5,00 | 4,51 |
| 4,08 | 3,53 | 5,98 | 5,56 | 5,01 | 4,52 |
| 6,97 | 3,54 | 5,99 | 8,43 | 5,02 | 4,53 |
| 6,98 | 3,55 | 6,00 | 8,44 | 5,03 | 4,54 |
| 6,99 | 3,56 | 6,01 | 8,45 | 5,04 | 4,55 |
| 7,00 | 3,57 | 6,02 | 8,46 | 5,05 | 4,56 |
| 7,01 | 3,58 | 6,03 | 8,47 | 5,06 | 4,57 |
| 7,02 | 3,59 | 6,04 | 8,48 | 5,07 | 4,58 |
| 7,03 | 3,60 | 6,05 | 8,49 | 5,08 | 4,59 |
| 9,91 | 6,45 | 6,06 | 8,50 | 7,94 | 7,44 |
| 9,92 | 6,46 | 8,91 | - | 7,95 | 7,45 |
| 9,93 | 6,47 | 8,92 | - | 7,96 | 7,46 |
| 9,94 | 6,48 | 8,93 | - | 7,97 | 7,47 |
| 9,95 | 6,49 | 8,94 | - | 7,98 | 7,48 |
| 9,96 | 6,50 | 8,95 | - | 7,99 | 7,49 |
| 9,97 | 6,51 | 8,96 | - | 8,00 | 7,50 |
| - | 6,52 | 8,97 | - | 8,01 | 7,51 |
| - | 6,53 | 8,98 | - | 8,02 | 7,52 |
| - | 9,42 | 8,99 | - | 8,03 | 7,53 |
| - | 9,43 | 9,00 | - | - | - |
| - | 9,44 | - | - | - | - |

Table 5.2: Output of jump-time function applied to all three variables

| | | | | |
|---|---|---|---|---|
| Switch Type 1(Up-jump var.1) | 0.93 | 4.05 | 7.00 | 9.94 |
| Switch Type 2(Up-jump var.2) | 0.40 | 3.54 | 6.49 | 9.43 |
| Switch Type 3(Up-jump var.3) | 3.04 | 6.01 | 8.95 | - |
| Switch Type 4(Down-jump var.4) | 2.56 | 5.53 | 8.47 | - |
| Switch Type 5(Down-jump var.5) | 2.04 | 5.03 | 7.98 | - |
| Switch Type 6(Down-jump var.6) | 1.51 | 4.54 | 7.49 | - |

Table 5.3: Values at Switching Instances

| | | | | |
|---|---|---|---|---|
| Var.1 | (0,18-1,02-1,39) | (1,65-0,99-0,59) | (0,38-1,01-1,38) | (1,61-0,98-0,59) |
| | (0,36-1,01-1,38) | (0,38-0,99-1,38) | (1,62-0,98-0,61) | |
| Var.2 | (0,34-0,32-0,98) | (1,42-1,68-0,99) | (0,61-0,38- 0,98) | (1,38-1,63-1,02) |
| | (0,61-0,39-0,99) | (1,39-1,63-1,00) | (0,62-0,38-0,98) | |
| Var.3 | (1,03-0,61- 0,37) | (0,99-1,45- 1,66) | (1,00-0,63- 0,37) | (0,99-1,40- 1,62) |
| | (1,02-0,62-0,37) | (1,02-1,39-1,64) | | |

time interval, the second variable has positive derivatives, indicating that the second variable exceeded the threshold at the switching instance. The opposite signs of derivatives at the switching instances imply the relation is repressilatory; the affector exceeding the threshold starts a negative trend in the responder. Correspondingly, if the signs are the same, the relation would be activatory.

Using the rules explained up to here, the regulatory relations can be inferred as follows. The so called close values, suggests that variable 2 regulates variable 1, variable 3 regulates variable 2 and variable 1 regulates variable 3. Using the characterization of jumps given in table 5.1 and the derivative estimates at the switching points, it is concluded that second and third variable has activatory effect on the corresponding responders whereas the first variable has repressilatory effect on the third one. The threshold values are inferred as $1, 0.99, 1, 01$.

The estimation of thresholds gives the state space partition into invariant spaces. The observations can be grouped according to the value of each variable with respect to its threshold. Each group would correspond to empirical observations from a different invariant subspace. The evolution in an invariant subspace $P_i$ is defined by linear differential equation $dy/dt = M_i y + b_i$ and the parameters $M$ and $b$ can be

Table 5.4: Estimated Values for $M_{s(t)}$ and $b_{s(t)}$ in Switching Differential Equations

| | $\widehat{M}_{s(t)}$ | $\widehat{b}_{s(t)}$ | $M_{s(t)}$ | $b_{s(t)}$ |
|---|---|---|---|---|
| Q=(0,0,0) | diag[0.99 1.00 0.99] | $[-0.03\ 0.01\ 1.98]^T$ | diag [1 1 1] | $[0\ 0\ 2]^T$ |
| Q=(0,0,1) | diag[1.01 1.03 0.98] | $[0.01\ 1.94\ 2.05]^T$ | diag [1 1 1] | $[0\ 2\ 2]^T$ |
| Q=(0,1,1) | diag[1.00 1.02 1.04] | $[1.91\ 2.01\ 2.00]^T$ | diag [1 1 1] | $[2\ 2\ 2]^T$ |
| Q=(1,0,0) | diag[1.03 0.94 1.00] | $[-0.04\ 0.07\ 0.02]^T$ | diag [1 1 1] | $[0\ 0\ 0]^T$ |
| Q=(1,1,0) | diag[1.04 1.01 0.99] | $[2.02\ -0.01\ -0.03]^T$ | diag [1 1 1] | $[2\ 0\ 0]^T$ |
| Q=(1,1,1) | diag[0.97 0.97 1.01] | $[2.00\ 2.08\ -0.05]^T$ | diag [1 1 1] | $[2\ 2\ 0]^T$ |

estimated using linear regression. At all instances $t_i$ when the observations fall into $P_i$, the differential equation can be approximated by $\dot{y}(t_i) = M_i y(t_i) + b_i$ where $\dot{y}(t_i)$ denotes the derivative estimate at time $t_i$. In the solution, the matrix $M$ is assumed to be diagonal and the parameters M and b are found accordingly for each state using L2 regression.

Having inferred all the parameters for the suggested switching hybrid system, the system parameters inferred by the algorithm are compared with the actual parameters introduced to the simulation in Table 5.4. The results are satisfactory; the regulatory relations are inferred correctly and the threshold values and differential equation parameters are quite accurately obtained.

## 5.2 Capabilities of the Proposed Method

The inference method studied here is designated for inferring the locally linear system parameters with the piecewise linear approximations. General form of piecewise linear approximations are applicable to nonlinear dynamical systems in general; the approximation accuracy can be adjusted with respect to the desired level of accuracy. However, the suggested piecewise linear model is developed for approximating nonlinear dynamical systems displaying switchings triggered by threshold crossings.

The switching detection part contains the innovative ideas of the inference method and that is why this part is examined most intensively in this thesis. In this aspect, local polynomial fitting method is considered satisfactory, especially in comparison with difference equations. However, the signal processing related applications could be studied at an introductory level and basic ideas in the literature are adopted.

The method performed well in switching detection for the simulation example but the robustness could still be improved. Deeper knowledge and expertise in signal processing could yield better results. Adaptive window size selection and use of different high pass filters are exemplary ideas promising significant improvements.

For single threshold partitioning with three variables system, there are $2^3 = 8$ possible invariant subspaces. In the example studied here, 6 of these are observed and parameters governing the vector fields in each invariant subspace were inferred. In case of a simulation, in order to infer the vector fields governing the evolution in invariant subspaces corresponding to discrete states of system $Q = (0, 1, 0)$ and $Q = (1, 0, 1)$, the same dynamics should be simulated with different initial conditions. Accordingly, if the experiment, from which the empirical data from a real event is gathered, can be repeated with different initial or if interventions are possible in the experiment setup, all possible states of the system can be observed and inferred as well.

In the simulation example, a single affecting variable was defined for each variable and these relations were inferred with the proposed algorithm. Comparing the values of candidates at the switching instances of responding variable is employed in finding the regulating variable. This approach in finding the regulator variable is valid only if the system dynamics can be explained by single affector for each variable. Conversely, if none of the variables were found sufficiently close at the switching instances of the responding variable, then more complex regulatory dynamics involving two affecting variables would be considered and more complicated logical rules should be developed accordingly. So, it would be reasonable to start the procedure with the assumption of one regulator variable for a given responding variable and increasing this number until the switching dynamics of that variable can be fully explained.

## 5.3 Discussion

The problems where the suggested inference methodology is expected to yield satisfactory results are briefly mentioned in the previous parts. Among other problems, gene network inference is recently a very popular area since the recent technological improvements increased data availability and quality for this problem. As a result, the previous studies in this field have been investigated carefully to extract useful ideas for development of an the inference method for piecewise linear systems. Moreover, the limitations and shortcomings of those approaches are studied to provide guidance for possible improvement strategies.

In their research, Perkins et al. [28] defined logical rules to infer gene-gene regulatory relations in a gene network abstracted by NK model. They derived the Boolean state of each variable by using the known threshold values. Hence, this type of method demands a priori knowledge of thresholds for all variables. Having defined the logical states, they investigated the switching occurrences of variables to find the regulator variables. With the method proposed here, the fundamental improvement in this aspect is the ability of finding the unknown threshold values directly from empirical data. The method we propose here suggests *going backwards*; finding the switching occurrences from jumps in time derivatives and then using similar logical reasoning to infer the regulatory relations and the corresponding thresholds.

The absence of complete information on threshold levels is one of the primary concerns of this work because if the threshold values for the variables are not given, identification of switchings cannot be straightforward anymore. The steps of inference followed after finding the thresholds have a quite similar basis to previous works. For example, NK models can be useful for obtaining the network structure only after the switchings are found correctly. However their use is restricted to known thresholds case.

The inference of switching dynamics and model parameters provides limited knowledge on the observed system. For obtaining further information on the system, there are also certain alternatives. Thomas et al. [37, 38] defined the notions of multistationarity, memory and stable temporal periodicity in terms of positive and negative feedback circuits in a network. Having identified the regulatory relations in the system, this and similar approaches can be considered as a complementary posterior study for our method to yield detailed results about certain qualitative features of the entire dynamical system. This type of effort will provide better understanding of system dynamics.

One of the primary concerns in the development of inference method is the robustness. In the relevant works cited in this thesis, piecewise linear models are equipped with difference equations and inference methodologies are developed accordingly. However, although this type of approach can be promising in terms of computational efficiency, the difference equations are not robust for estimating the derivatives from discrete empirical data and the results would not be reliable in the existence of noise. Use of local polynomial fitting improves inference algorithm proposed herein in terms of robustness. Another important application that contributes to reliability of the algorithm is in switching detection. Comparison of the values of all variables at every occasion of the same switching reduces the possibility of accidental generalizations at this step.

The original contribution of this work is actually synthesizing the ideas used for different problems. Detecting the jumps in a time series is an idea used in event detection [14] and jump detection techniques are studied for various reasons in signal processing. Nonsmoothness in the evolution of a variable can be a sign of disorder if it arises too frequently. On the other hand, it would indicate a sudden change of behavior, a *switching*, if it occurs between two dissimilar steady patterns. Hence, once the switchings in the continuous evolution are interpreted as events to be detected, the remaining ideas of the proposed method follows accordingly.

It is mentioned that use of derivative estimation by local polynomial fitting improves the robustness of the algorithm. For many dynamical systems, existence of noticeable noise in discrete empirical observations is inevitable; the inherent noise in gene networks can be given as an example. The proposed method offers increased robustness by using local polynomial fitting for derivative estimation instead of difference equations. The trial runs of the algorithm on the synthetic data with different noise levels are also quite promising in estimating switchings accurately in the existence of noise.

Another advantage of using switching detection is the possibility of making feedback on the experiment setup. For example, for the repressilatory 3-variable network -introduced in Chapter 4- for certain initial conditions on the variables, it may not be able to distinguish the activator or repressor variable even if the switchings can be detected successfully. Even in this case, the implementation of method would yield useful suggestions about the candidate regulatory relations. Moreover, appropriate strategies for adding perturbations to experiment setup or restarting the experiment with different initial values would be possible.

# Chapter 6

# Conclusion

In this work, an improved algorithm for inferring piecewise linear models whose switching is governed by threshold phenomena is proposed. The motivation of working on this particular model family is their capability of modeling different quantitative behaviors including chaotic and multistationary behaviors, fixed point, periodic and quasiperiodic attractors. A brief outline of main ideas used in the method is given and the conformity of techniques used in critical steps of the algorithm is questioned in the validation of every step. Afterwards, advantages and limitations of the method are discussed.

Since gene networks are adequate for the application of this methodology discussed here, the gene network inference in the literature is examined to verify the adequacy of inference method for genetic regulation examples. The biological background supports our assumptions on threshold phenomenon and switching property related to piecewise linear models. Considering the specifications of gene networks, the inference method explained in this work is considered applicable for inferring regulatory relations in gene networks from empirical records of gene transcription activity. Considering its strengths summarized in the Discussion section, it is expected to yield promising results for gene regulatory network inference problem.

There are possible extensions that are considered as future work. In the section 3.3, the inference rules based on switching point comparison are derived under the assumption of single activator and single repressor for each variable. However, these rules can be extended for more complex regulatory relations. For example, if we assume that the activity of a variable in the model can be regulated by more than one variable, improved logical mappings should be included in the model and other efforts can be made to infer the structure of these logical rules in the model.

In this study, the ideas about jump detection are studied most intensively since implementation of jump detection is a new idea for piecewise linear model inference. Accordingly, many ideas about jump detection methods in signal processing are adopted. Deeper understanding of these issues can allow modifications and improvements of the inference methodology suggested here and more reliable inference algorithms can be developed on the same fundamental ideas.

The robustness of the inference algorithm is very important. Further improvements on the algorithm should be considered in this aspect also. In the literature, there are examples in which the *window* mentioned in the local polynomial fitting for derivative estimation can be selected adaptively depending on the structure of empirical data. This sort of known methods should be applied to make more robust derivative estimation. Likewise, in the jump detection procedure, use of different wavelets can improve the reliability of the method as well.

To sum up, the methodology studied in this work can be considered innovative since it combines the ideas from different fields for the solution of inferential modeling problem. Hopefully, the ideas presented here will provide guidance for the further work and lead to development of useful methodologies for inference problem.

# REFERENCES

[1] D. Akay. Inference of switching networks by using piecewise linear formulation, 2006.

[2] J. Banks, V. Dragan, and A. Jones. *Chaos-A Mathematical Introduction*. Cambridge University Press, 2003.

[3] H. Beltrami and R. N. Harris. Foreword: Inference of climate change from geothermal data. *Global and Planetary Change*, 29:149–152, 2001.

[4] P. Brazhnik. Inferring gene networks from steady-state response to single-gene perturbations. *Journal of Theoretical Biology*, 237:427–440, 2005.

[5] M.-Y. Cheng and P.r Hall. Methods for tracking support boundaries with corners. *Journal of Multivariate Analysis*, 97:1870–1893, 2006.

[6] J. R. Clay. Axonal excitability revisited. *Progress in Biophysics and Molecular Biology*, 88:59–90, 2005.

[7] W.S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836, 1979.

[8] T. Dean and S. Leach. *Introduction to Learning Dynamical Systems*. http://www.cs.brown.edu/.

[9] M. E. Driscoll and T. S. Gardner. Identification and control of gene networks in living organisms via supervised and unsupervised learning. *Journal of Process Control*, 16:303–311, 2006.

[10] A. J. Drummond, A. Rambaut, B. Shapiro, and O. G. Pybus. Bayesian coalescent inference of past population dynamics from molecular sequences. *Molecular Biology and Evolution*, 22:1185–1192, 2005.

[11] R. Edwards and L. Glass. Combinatorial explosion in model gene networks. *Chaos*, 10:691–704, 2000.

[12] O. Eriksson, Y. Zhou, and J. Tegner. Modeling complex cellular networks - robust switching in the cell cycle ensures a piecewise linear reduction of the regulatory network. In *Proceedings of 43rd IEEE Conference on Decision and Control*, pages 117–123, 2004.

[13] J. Fan. Local linear regression smoothers and their minimax efficiency. *The Annals of Statistics*, 21:196–216, 1993.

[14] C. Feigenwinter and R. Vogt. Detection and analysis of coherent structures in urban turbulence. *Theoretical and Applied Climatology*, 81:219—230, 2005.

[15] J. Gebert, H. Oktem, S.W. Pickl, N. Radde, G.-W. Weber, and F.B. Yılmaz. Inference of gene expression patterns by using a hybrid system formulation an algorithmic approach to local state transition matrices. In *Proceedings of Inter-Symp of The International Institute for Advanced Studies in Systems Research and Cybernetics*, 2004.

[16] V. Guralnik and J. Srivastava. Event detection from time series data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 33–42, 1999.

[17] M. T. Heath. *Scientific Computing: An Introductory Survey.* McGraw-Hill, 1997.

[18] C.P. Hemmings, M. A. Srokosz, P. Challenor, and J.R. Fasham. Split-domain calibration of an ecosystem model using satellite ocean colour data. *Journal of Marine Systems*, 50:141–179, 2004.

[19] S. Imoto, T. Goto, and S. Miyano. Estimation of genetic networks and functional structures between genes by using bayesian networks and nonparametric regression. In *Proceedings of Pacific Symposium on Biocomputing 7*, pages 175–186, 2002.

[20] S. Y. Kim, S. Imoto, and S. Miyano. Inferring gene networks from time series microarray data using dynamic bayesian networks. *Briefings in Bioinformatics*, 4:228–235, 2003.

[21] Y. Luan and Z. Xie. The wavelet identification for jump points of derivative in regression model. *Statistics and Probability Letters*, 53:167–180, 2001.

[22] J. Lygeros. *Lecture Notes on Hybrid Systems.* http://robotics.eecs.berkeley.edu/.

[23] T. MacCarthy, A. Pomiankowski, and R. Seymour. Using large-scale perturbations in gene network reconstruction. *BMC Bioinformatics*, 6, 2005.

[24] E. Masry. Local polynomial fitting under association. *Journal of Multivariate Analysis*, 86:330–359, 2003.

[25] H. Oktem. *Hybrid Systems Lecture Notes*. http://hybrid.iam.metu.edu.tr/.

[26] H. Oktem. A survey on picewise-linear models of regulatory dynamical systems. *Nonlinear Analysis*, 63:336–349, 2005.

[27] R. K. Pearson. Selecting nonlinear model structures for computer control. *Journal of Process Control*, 13:1–26, 2003.

[28] T. J. Perkins, M. Hallett, and L. Glass. Inferring models of gene expression dynamics. *Journal of Theoretical Biology*, 230:289–299, 2004.

[29] A. U. Raghunathan, J. R. Perez-Correa, E. Agosin, and L. T. Biegler. Parameter estimation in metabolic flux balance models for batch fermentation - formulations and solution using differential variational inequalities. Preprint, Carnegie Mellon University.

[30] M. J. Rewienski. *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems*. PhD thesis, Technical University of Gdansk, Poland, 2003.

[31] I. Shmulevich, E. R. Dougherty, and W. Zhang. Control of stationery behaviour in probabilistic boolean networks by means of structural intervention. *Journal of Biological Systems*, 10:431–445, 2002.

[32] C. Soulé. Mathematical approaches to differentiation and gene regulation. *Comptes Rendus Biologies*, 329.

[33] C. J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5:595–645, 1977.

[34] R. N. Strickland and H. I. Hahn. Wavelet methods for extracting objects from complex backgrounds. In *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 7–12, 1996.

[35] J. Sun, Y. Chen D. Gu, and S. Zhang. A multiscale edge detection algorithm based on wavelet domain vector hidden markov tree model. *Pattern Recognition*, 37:1315–1324, 2004.

[36] Z. Sun and S.S. Ge. Analysis and synthesis of switched linear control systems. *Automatica*, 41:181–195, 2005.

[37] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory: I. structural conditions of multistationarity and other nontrivial behavior. *Chaos*, 11:170–179, 2001.

[38] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory: Ii. logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11:180–195, 2001.

[39] P. Waltman and E. Butz. A threshold model of antigen-antibody dynamics. *Journal of Theoretical Biology*, 65:499–512, 1977.

[40] Y. Wang. Jump and sharp cusp detection by wavelets. *Biometrica*, 82:385–397, 1995.

[41] B. H. Weber. Origins of order in dynamical models. *Biology and Philosophy*, 13:133–144, 1998.

[42] H. Wong, W. Ipa, and Y. Li. Detection of jumps by wavelets in a heteroscedastic autoregressive model. *Statistics and Probability Letters*, 52:365–372, 2001.

# APPENDIX 1. derest.m

```
% (C) Ahmet Melih Selcuk
function [out1,out2] = derest (row_data,derest_winsize,derest_polydeg,stepsize);
%INPUTS
%row_data: Discrete time data of selected continuous variable in the PWL system
%derest_winsize: window size of local polynomials in derivative estimation
%derest_polydeg: degree of fitted polynomial in derivative estimation
%stepsize is the time interval between consecutive samples
datalength = length(row_data);
%weig keeps accumulation of weights, Bufff1 keeps polynomial coefficient
%estimates and Bufff2 keeps local derivartive estimates
weig=zeros(size(row_data));
Bufff1=zeros(size(row_data));
Bufff2=zeros(size(row_data));
%iteration would contain accumulation, therefore we initialize the arrays
%by zeros of same length
iter=datalength-derest_winsize+1;
winweig=(kaiser(derest_winsize, (derest_winsize-2)))';
%second parameter of kaiser kernel is chosen EXPERIMENTALLY
for i=1:iter
data=row_data(i:i+derest_winsize-1);
[estims,derests]=loc_fit(data,derest_polydeg);
%derivative estimation function would be determined up to necessity
%of out1 which contains polynomial coefficients
Bufff1(i:i+derest_winsize-1)=Bufff1(i:i+derest_winsize-1)+estims.*winweig;
Bufff2(i:i+derest_winsize-1)=Bufff2(i:i+derest_winsize-1)+derests.*winweig;
weig(i:i+derest_winsize-1)=weig(i:i+derest_winsize-1)+winweig;
```

```
%at each iteration, 'winsize' many entries are updated. In fact, almost
%every parameter is updated 'winsize' times and averaging is achieved
%by entry-wise division of Bufff1 and Bufff2 by weig
end
out1=Bufff1./weig;
out2=(Bufff2./weig)/stepsize;
```

# APPENDIX 2. jump_time.m

```
% (C) Ahmet Melih Selcuk
%This function is called in the function jump_time
function [out]=fillter(derests,filter_winsize)
%INPUTS
%derests: approximate derivative values by obtained function derest
%filter_winsize: the width of high pass filter used
Bufff=zeros(size(derests));
%winsize should be EVEN!!!
winweig = FSPECIAL('gaussian',[1 filter_winsize],1.4);
for i=((filter_winsize/2)+1):filter_winsize
winweig(i)=winweig(i)*-1;
end
for i=1:length(derests)-filter_winsize+1;
Bufff(i)= derests(i:i+filter_winsize-1)*winweig';
%at each iteration, 'winsize' many entries are updated. In fact, almost
%every parameter is updated 'winsize' times and averaging is achieved
%by entry-wise division of Bufff1 and Bufff2 by weig
end
out=Bufff;
%OUTPUT
%out: out is an array of length 'winsize' that is obtained by convolution of derivative estimates with high pass filter
```

# APPENDIX 3. fillter.m

```
% (C) Ahmet Melih Selcuk
%This function is called in the function jump_time
function [out]=fillter(derests,filter_winsize)
%INPUTS
%derests: approximate derivative values by obtained function derest
%filter_winsize: the width of high pass filter used
Bufff=zeros(size(derests));
%winsize should be EVEN!!!
winweig = FSPECIAL('gaussian',[1 filter_winsize],1.4);
for i=((filter_winsize/2)+1):filter_winsize
winweig(i)=winweig(i)*-1;
end
for i=1:length(derests)-filter_winsize+1;
Bufff(i)= derests(i:i+filter_winsize-1)*winweig';
end
out=Bufff;
%out: out is an array of length 'winsize' that is obtained by convolution of derivative estimates with high pass filter
```

# APPENDIX 4. fillter.m

```
% (C) Ahmet Melih Selcuk
%FUNCTION CALL
%This function is called by function derest
function [loc_estims, loc_der]=loc_fit (data,polydeg)
%INPUTS
%loc_data: part of discrete data within the selected window
%polydeg: degree of polynomial to be fitted to loc_data
N=length(data);
loc_der=zeros(1,N);
x=[1:N];
poly_coef=polyfit(x,data,polydeg);
loc_estims=polyval(poly_coef,x);
for i=1:N
for j=1:polydeg
loc_der(i)=loc_der(i)+(polydeg-j+1)*poly_coef(j)*i^(polydeg-j);
end
end
end
%OUTPUTS
%loc_estims: approximate values obtained from fitted polynomial
%loc_der: approximate derivatives obtained from fitted polynomial
```

# APPENDIX 5. pickobs.m

```
% (C) Ahmet Melih Selcuk

function [times]=pick_obs(num_vars,thr,data,dstate)

%This function returns the time instances corresponding to a given discrete state of hybrid system

out=[];

for i=1:length(data)

a=0;

for j=1:num_vars

a=a+power(2,j-1)*(data(j,i)>thr(j));

end

if a==dstate

out=[out i];

end

end

times=out;
```