MODELING GENE REGULATORY NETWORKS FROM TIME SERIES DATA USING PARTICLE FILTERING

A Thesis

by

AMINA NOOR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2011

Major Subject: Electrical Engineering

MODELING GENE REGULATORY NETWORKS FROM TIME SERIES DATA USING PARTICLE FILTERING

A Thesis

by

AMINA NOOR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Co-Chairs of Committee,	Erchin Serpedin
	Mohamed Nounou
Committee Members,	Ulisses Braga-Neto
	Shuguang Cui
	Radu Stoleru
Head of Department,	Costas N. Georghiades

August 2011

Major Subject: Electrical Engineering

ABSTRACT

Modeling Gene Regulatory Networks from

Time Series Data Using Particle Filtering. (August 2011)

Amina Noor, B.E., M.S., National University of Sciences and Technology

Co–Chairs of Advisory Committee: Dr. Erchin Serpedin
Dr. Mohamed Nounou

This thesis considers the problem of learning the structure of gene regulatory networks using gene expression time series data. A more realistic scenario where the state space model representing a gene network evolves nonlinearly is considered while a linear model is assumed for the microarray data. To capture the nonlinearity, a particle filter based state estimation algorithm is studied instead of the contemporary linear approximation based approaches. The parameters signifying the regulatory relations among various genes are estimated online using a Kalman filter. Since a particular gene interacts with a few other genes only, the parameter vector is expected to be sparse. The state estimates delivered by the particle filter and the observed microarray data are then fed to a LASSO based least squares regression operation, which yields a parsimonious and efficient description of the regulatory network by setting the irrelevant coefficients to zero. The performance of the aforementioned algorithm is compared with extended Kalman filtering (EKF), employing Mean Square Error as the fidelity criterion using synthetic data and real biological data. Extensive computer simulations illustrate that the particle filter based gene network inference algorithm outperforms EKF and therefore, it can serve as a natural framework for modeling gene regulatory networks.

To my family

ACKNOWLEDGMENTS

Praise be to God, the Cherisher and Sustainer of the worlds. I am very grateful for His infinite mercy and blessings.

I extend my special thanks to my advisors Dr. Erchin Serpedin and Dr. Mohamed Nounou, for their guidance and support. I am particularly indebted to Dr. Serpedin for his constant motivation and encouragement. I am also grateful to Dr. Ulisses Braga-Neto, Dr. Robert Cui and Dr. Radu Stoleru for serving on my committee.

I would like to thank my colleagues at the Department of Electrical and Computer Engineering particularly, Sabit, Bilal, Fang-Han, Sang-Woo, Kwadwo, and Yi who provided a nice and friendly work environment. In addition, I want to thank my friends in College Station, Eman, Sevgi and Amina who have made my stay here a wonderful experience.

In the end, I want to thank my family for their continued support, encouragement and prayers. I am highly indebted to my parents for their immeasurable sacrifices. I am specially thankful to my brother, Umar, and my sister, Mariyum, for their love. I extend my immense gratitude to my husband, Aitzaz, for his unconditional support and for always standing by me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Gene regulation is one of the most intriguing processes taking place in living cells. With hundreds of thousands of genes at their disposal, cells must decide which genes to express at a particular time. As the cell development evolves, different needs and functions entail an efficient mechanism to turn the required genes on while leaving the others off. Cells can also activate new genes to respond effectively to environmental changes and perform specific roles. The knowledge of which gene triggers a particular genetic condition can help us ward off the potential harmful effects by turning that gene off. For instance, cancer can be controlled by deactivating the gene that causes it. Fig. 1 gives a brief description of the gene regulation. Receptors located outside of the cell-membrane receive signals from the environment which are passed through the cytoplasm into the nucleus. This activates a protein called transcription factor in the nucleus which on binding to the promoter region of the gene triggers the enzyme, RNA polymerase. This enzyme transcribes the DNA into mRNA, which is then translated into protein.

The amount of mRNA produced tells us how active or functional a gene is. The level of gene functionality can be measured using microarrays or gene chips to produce the gene expression data. Intelligent use of this data can help us get an understanding of how the genes are interacting in a living organism. While the theoretical applications of gene regulation are extremely promising, it requires a thorough understanding of this complex process. Different genes may cooperate to produce a particular reaction while a gene may repress another as well. The potential benefits

---

The journal model is *IEEE Transactions on Automatic Control*.

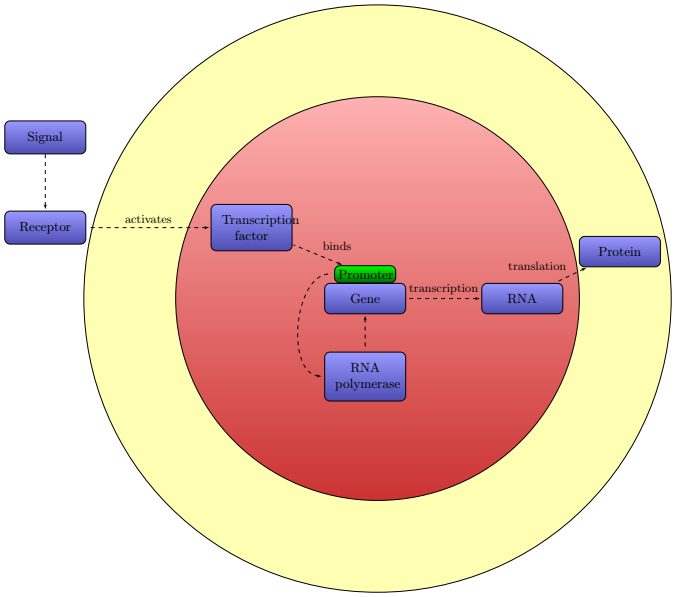

Fig. 1. Gene regulation mechanism.

of gene regulation can only be reaped if a complete and accurate picture of gene interactions is available. A network specifying how different genes are interconnected can go a long way in helping us understand the gene regulation mechanism.

A. Gene Regulatory Networks

A particular way to describe gene interactions is through a *gene regulatory network*. Gene regulatory networks are a class of graphical models that serve to capture the control and interactions taking place among biological components including mRNA, proteins and DNA sequences. Genes constitute the nodes in this graphical network while the relations between interacting genes are modeled by edges connecting the related nodes. Such a network depicts various interdependencies among genes. A typical gene network is shown in Fig. 2. The correlation among various genes in the network is then determined by using the gene expression data. This quantitative anal-

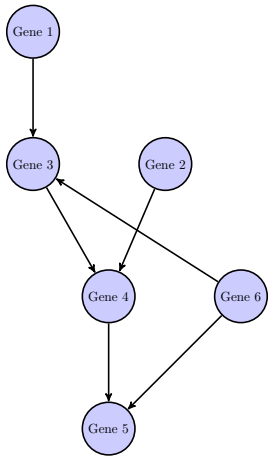

Fig. 2. A typical gene regulatory network.

ysis represents the extent to which a gene is affected by other genes in the network. A key ingredient of this approach is an accurate and representative modeling of gene networks. Precise modeling of a regulatory network coupled with efficient inference algorithms can potentially pave the way for curing genetic diseases, improving diagnostic procedures and producing drug designs with greater impact.

Recent advances in DNA microarray technology coupled with their tremendous applications has spurred significant research efforts in inferring gene regulatory networks. A natural consequence of this improvement in technology is a deluge of biological data obtained through simultaneous monitoring of gene expression profiles of large sequences. It is imperative that simple and computationally efficient algorithms are developed which provide an insight into the regulatory and physical interactions taking place among living cells. Inferring useful information from large amounts of biological data by utilizing experimental techniques alone can incur significant costs as well as consume time. With experimental resources coming at a premium, it is precisely this idea of their economic use that advocates computational biology as a

viable and attractive alternative [1], [2].

B. Literature Review

Modeling of gene networks is, in essence, a set of decision rules that describe the activation and repression of each gene via various proteins. The proteins produced by transcription and translation might serve as activators or repressors for other genes resulting in feedback loops which can be both positive and negative. Depending upon our prior knowledge about possible interactions and the amount of gene expression data available, there can be different levels of gene network models.

Several methods of modeling gene networks have been reported in literature, which vary from being very simple to very sophisticated [3], [4], [5]. The simplistic class of models can be called a Parts list which includes models specifying various components of gene network [6], [7]. Some modeling techniques establish the interactions among the genes based on statistical dependence, e.g. clustering [8], [9]. Information-theoretic criteria have also been proposed to quantify the extent of correlation between genes and infer the corresponding regulatory network [10]. This approach can, however, model static relations only. A more precise and insightful construction of a gene regulatory network can be obtained by incorporating the random effects caused by perturbations and the evolution of gene reactions in time. To facilitate the extraction of useful information from expression profiles time series data, various dynamical models have been employed [11]. In particular, Bayesian networks [12], factor graphs [13] are frequently used to model conditional dependencies among genes in a network. Boolean networks [14], [15], neural network [16] etc., have also been proposed as potential candidates to model the control of various components in the cell and external factors. All of the aforementioned algorithms come with their

respective advantages and short comings. A lot of research is being devoted to introduce improvements in the working of these algorithms and enhance our understanding of gene interactions.

Of the statistical techniques currently applied to model gene networks, dynamic Bayesian networks have received the most widespread attention [17], [18]. Dynamic Bayesian networks offer significant advantages in terms of incorporating our prior belief about the system structure in stochastic modeling to identify the system parameters. State space models [19], [20], [21] and Kalman filter, which are specific instances of dynamic Bayesian networks, have also been employed to model gene regulatory networks [3], [22]. Kalman filter suffers from an inherent drawback of being applicable to linear Gaussian models only. However, the interdependencies among various genes are rarely, if ever, linear [3]. In order to capture complex gene interactions efficiently, it is crucial to alleviate the assumption of a necessarily linear model and develop algorithms that produce desired results even in the presence of possible nonlinearities in the system model [3]. Extended Kalman filter (EKF) is one such method for estimation and prediction in case of nonlinearity and has been frequently used to perform inference in gene networks [23], [24]. This approach works well in the presence of steady state data and if the system presents slow dynamics. While it offers some advantages in terms of simplicity and small data needs, it is, at best, only an approximation since it relies on linearization of the nonlinearity. There is a considerable degradation in the performance of EKF if either the initial estimate of the state is wrong, or there are deficiencies in the modeling of the system. Clearly, this loss in acceptable performance is a direct consequence of the linearizing operation. Therefore, advanced techniques that preserve any inherent nonlinear structures in the state evolution and deliver performance guarantees with desired fidelity are required.

C. Major Contributions

To cope with nonlinearities, this thesis proposes the usage of particle filtering techniques. A generalization of Kalman filter, particle filter can accurately model the evolving dynamics of a system by catering for any possible nonlinearity, thus removing the sub-optimality caused by linearization approximations. The noise impairing the physical system can arise due to intrinsic factors, such as translation and transcription taking place in the cell, or due to extrinsic factors. In this scenario, particle filtering offers another distinct advantage over extended Kalman filter in that it can be used in the presence of arbitrary noise distributions whereas extended Kalman filter assumes the noise to be Gaussian. This work proposes a method to reverse engineer gene regulatory networks using time series data. The microarray data is assumed to obey a linear model. To obtain a more accurate and precise picture of gene interactions, a nonlinear model for gene expressions and a discrete time state space system of equations are considered to model possible time variations. Our major contributions in this work can be summarized as follows [25],[26].

1. A particle filter based approach is presented to model nonlinearities in a gene network instead of relying on first order approximations. The gene regulatory network is expressed as a state space model and a sigmoid squash function is used to model the nonlinearity. The states are recursively estimated using particle filter whereas the system parameters required in this estimation are estimated online using a Kalman filter operation. This approach helps to create a more accurate representation of the network by alleviating the sub-optimality introduced by approximations used in contemporary methods [24].

2. A key observation in modeling a gene regulatory network is that a particular gene interacts with a few other genes only and as such, many of the system

parameters signifying these 'weak' relationships are irrelevant. The parameter vector is thus, expected to be sparse. To capture this sparsity, the particle filter is augmented with the well known *Least Squares Shrinkage Selection Operator* (LASSO) based least squares regression operation. LASSO helps us to identify the 'relevant' subset of system parameters of the network. This yields a parsimonious and concise description of the gene regulatory network.

3. The performance of the aforementioned algorithm is rigorously evaluated for synthetic data as well as real biological data sets for Malaria and Worm time series gene expression profiles. The results are contrasted with those reported in [24]. It is demonstrated that particle filtering followed by LASSO outperforms the nonlinear approximation based method proposed in [24] for synthetic as well as real data. Our proposed algorithm, therefore, can serve as a natural framework for modeling gene regulatory networks.

D. Organization

The remainder of this thesis is organized as follows.

- CHAPTER II outlines the underlying nonlinear state space system used to model the gene regulatory network.
- CHAPTER III presents our main algorithm using a particle filter based state estimation followed by LASSO operation to estimate the sparse parameter vector.
- CHAPTER IV tests the performance of this algorithm for synthetic as well as real data and results are compared with the extended Kalman filter proposed in [24].

- CHAPTER V concludes the thesis along with some directions for future research.

CHAPTER II

SYSTEM MODEL

A. State Space Model

The dynamical gene system is modeled using a standard state space approach. Assuming a system consisting of N genes, the model for the evolution of states at the ith time instant can be expressed as

$$\mathbf{y}_i = g(\mathbf{y}_{i-1}, \mathbf{w}_{i-1}) \tag{2.1}$$

where the function $g(.)$ characterizes the regulatory relationship among various genes and is not constrained to be linear in order to allow a complete generalization of the model. The state vector $\mathbf{y}_i$ represents the gene expression values at a particular time instant i and the noise $\mathbf{w}_i$ impairing the system is assumed to be $i.i.d$ Gaussian such that $w_{i,n} \sim \mathcal{N}(0, \sigma_w^2)$. The microarray data is represented in terms of the variables $\mathbf{z}_i$ which also constitute a set of noisy observations. At the ith time instant, the states $\mathbf{y}_i$ are assumed to be related to the gene expression levels $\mathbf{z}_i$ as

$$\mathbf{z}_i = h(\mathbf{y}_i, \mathbf{v}_i) \tag{2.2}$$

where $\mathbf{v}_i$ is considered Gaussian such that $v_{i,n} \sim \mathcal{N}(0, \sigma_v^2)$. The system model is depicted in Fig. 3 at the ith instant.

As discussed before, in order to capture the inherent nonlinearity relationships existing among genes, a linear restriction of the function $g(.)$ is alleviated. In partic-

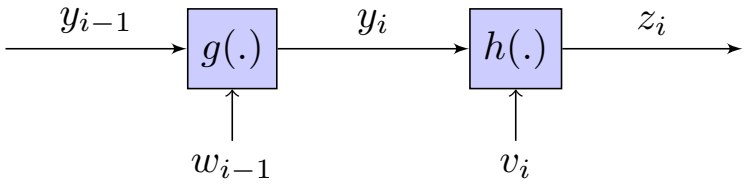

Fig. 3. System model block diagram.

ular, the following model is considered for the evolution of states [25]

$$y_{i,n} = \sum_{m=1}^{N} b_{nm} f(y_{i-1,m}) + w_{i,n}$$
$$i = 1, ..., I, \quad n = 1, ..., N \tag{2.3}$$

where the unknown constants b_{nm} model the nonlinear regulatory relations among various genes. The nonlinear function $f(y_{i-1,m})$ is the sigmoid squash function given by

$$f(y_{i-1,m}) = \frac{1}{1 + e^{-y_{i-1,m}}} \tag{2.4}$$

which is illustrated in Fig. 4. This function enables the conditional distribution of the state to remain Gaussian, although the mean is now a nonlinear function of the parents [3].

In most of the current literature, the microarray data is assumed to be fully correlated with the gene expression. This assumption is maintained for simplicity and ease of inference. However, microarray experiments are known to be noisy and it is very important to incorporate the stochasticity in the microarray data model. In this thesis, a linear Gaussian model for the microarray data is considered which can

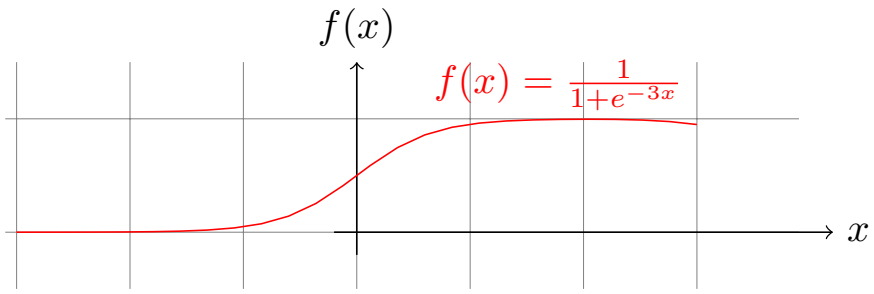

Fig. 4. Sigmoid squash function.

be expressed at the ith time instant as [22], [24]

$$\mathbf{z}_i = \mathbf{y}_i + \mathbf{v}_i. \tag{2.5}$$

The system model outlined above is complete in the sense that it captures all important features of the gene regulatory network, e.g., nonlinearity, noise and dynamics.

B. Problem Statement

Given a set of noisy observations $\mathbf{z}_i$ at various time instants, which is assumed to have evolved through the state space model described in (2.3) and (2.5), our goal is to infer the gene regulatory network by determining the unknown constants b_{nm}. Accurate estimates of b_{nm} enable us to quantify the degree of interactions among genes.

CHAPTER III

METHOD TO INFER GENE REGULATORY NETWORKS

"An algorithm must be seen to be believed".

-Donald Knuth [27].

In this section, the methodology proposed to infer the system parameters in (2.3) is described [25]. Our approach is best illustrated in Fig. 5. The algorithm is presented in detail below.

A. Particle Filtering

Particle filtering, also known as Sequential Monte Carlo method, is a suboptimal algorithm which uses point masses to approximate the probability densities [28], [29], [30]. Particle filtering serves as a natural candidate for making inference in gene regulatory networks since it is not restricted to linear state evolution models. In addition, particle filter is also suitable for scenarios where the noise corrupting the system can assume arbitrary probability distributions.

Let $\mathbf{d}_i$ denote the set of all observations up to time i, i.e., $\mathbf{d}_i \triangleq [\mathbf{z}_1, \ldots, \mathbf{z}_i]^T$. Based on the measurements $\mathbf{d}_i$ and past state estimates $\mathbf{y}_{1:i-1}$, our objective is to estimate the current state $\mathbf{y}_i$. This requires the posterior density $p(\mathbf{y}_i|\mathbf{d}_i)$ of the state $\mathbf{y}_i$. The process is carried out in two steps which involve predicting the posterior density given the past observations and updating it given the current observation [1].

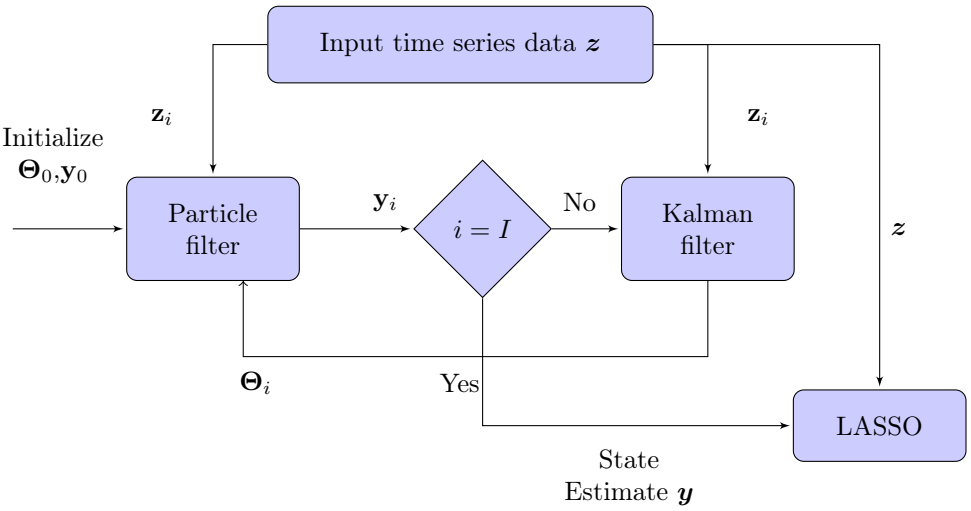

Fig. 5. Block diagram of gene regulatory network inference methodology.

The prediction step is given by

$$p(\mathbf{y}_i|\mathbf{d}_{i-1}) = \int p(\mathbf{y}_i|\mathbf{y}_{i-1})p(\mathbf{y}_{i-1}|\mathbf{d}_{i-1})d\mathbf{y}_{i-1} \tag{3.1}$$

where the posterior density $p(\mathbf{y}_{i-1}|\mathbf{d}_{i-1})$ is assumed available from the previous iteration. Based on the Markov model in (2.3), the conditional distribution of the state $\mathbf{y}_i$ can be expressed as

$$p(\mathbf{y}_i|\mathbf{y}_{i-1}; b_{nm}) = \frac{1}{(2\pi\sigma_w^2)^{N/2}} \exp\left\{ \frac{-||\mathbf{y}_i - \sum_{m=1}^{N} b_{nm} f(\mathbf{y}_{i-1,m})||^2}{2\sigma_w^2} \right\}. \tag{3.2}$$

The constants b_{nm} are assumed to be available through an online estimation using Kalman filter as described in the next section. At this point, we can utilize the observation $\mathbf{z}_i$ available at time i. The update step can be written as

$$p(\mathbf{y}_i|\mathbf{d}_i) = \frac{p(\mathbf{z}_i|\mathbf{y}_i)p(\mathbf{y}_i|\mathbf{d}_{i-1})}{p(\mathbf{z}_i|\mathbf{d}_{i-1})} \tag{3.3}$$

where the normalization constant is conveniently expressed as

$$\begin{aligned} p(\mathbf{z}_i|\mathbf{d}_{i-1}) &= \int p(\mathbf{z}_i|\mathbf{d}_{i-1}, \mathbf{y}_i) p(\mathbf{y}_i|\mathbf{d}_{i-1}) d\mathbf{y}_i \\ &= \int p(\mathbf{z}_i|\mathbf{y}_i) p(\mathbf{y}_i|\mathbf{d}_{i-1}) d\mathbf{y}_i \end{aligned} \tag{3.4}$$

and $p(\mathbf{z}_i|\mathbf{y}_i)$ is given by

$$p(\mathbf{z}_i|\mathbf{y}_i) = \frac{1}{(2\pi\sigma_v^2)^{N/2}} \exp\left\{ \frac{-||\mathbf{z}_i - \mathbf{y}_i||^2}{2\sigma_v^2} \right\}. \tag{3.5}$$

The prediction and update stages for state $\mathbf{y}_i$ are succinctly described in Algorithm 1. The process is initiated by drawing K particles $\{\mathbf{y}_0\}_{k=1}^K$ for an initial state $\mathbf{y}_0$ from a known prior density $p(\mathbf{y}_0)$. At the ith iteration, particles $\{\mathbf{y}_{i-1}\}_{k=1}^K$ sampled from the posterior density $p(\mathbf{y}_{i-1}|\mathbf{d}_{i-1})$ are assumed to be available.

In the *prediction* step, the particles $\{\mathbf{y}_{i-1}\}_{k=1}^K$ and $\{\mathbf{w}_{i-1}\}_{k=1}^K$, sampled from $p(\mathbf{w}_{i-1})$, are used to predict the state $\mathbf{y}_i$. This is accomplished by using

$$\mathbf{y}_i^* = g(\mathbf{y}_{i-1}, \mathbf{w}_{i-1}) \tag{3.6}$$

for all K particles. The unknown system parameters Θ needed in this prediction are supplied by an online estimation using Kalman filter and its details are deferred to the next section.

The observation $\mathbf{z}_i$ available at time instant i necessitates an *update* of the state estimate $\mathbf{y}_i$. The normalized likelihood for kth prior sample can be written as

$$\xi_k = \frac{p(\mathbf{z}_i|\mathbf{y}_i^{*k})}{\sum_{k=1}^K p(\mathbf{z}_i|\mathbf{y}_i^{*k})}. \tag{3.7}$$

The updated estimate for state $\mathbf{y}_i$ can now be obtained by drawing particles $\{\mathbf{y}_i\}_{k=1}^K \sim p(\boldsymbol{\xi}_i)$. At the termination of the update stage, the particles $\{\mathbf{y}_i\}_{k=1}^K$ are good approximations of samples from the posterior distribution $p(\mathbf{y}_i|\mathbf{d}_i)$. This process is depicted

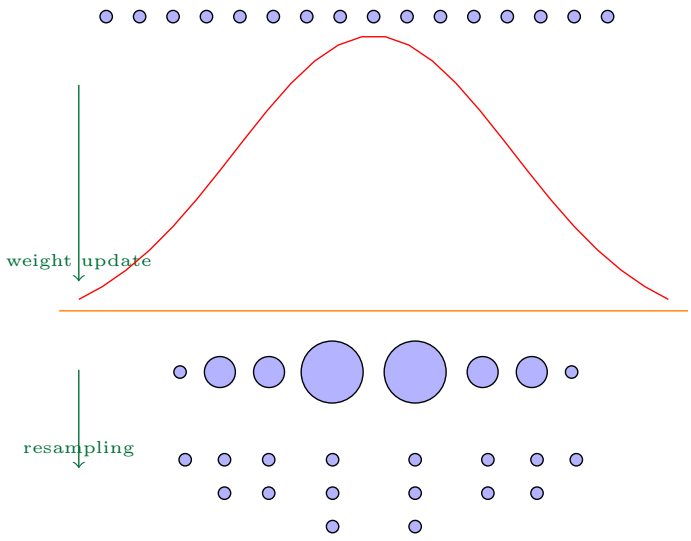

Fig. 6. Weight update and resampling in particle filter.

in Fig. 6

Particle filtering suffers from the well-known problem of sample attrition. As the algorithm proceeds, the variance of the importance weights can only increase. Hence, after some iterations, there will only be a single particle carrying most of the weight. This phenomenon is called the *degeneracy effect*. This problem is resolved by resampling in which the particles with small weights are removed and those with higher weights are replicated in proportion to their weights.

B. Kalman Filter

Kalman filter is an algorithm which is frequently employed to estimate the hidden variables in a linear state space model observed in Gaussian noise. It is an online algorithm which uses the noisy observations at each time instant to predict the unknown states as depicted in Fig. 7. In our framework, particle filter works in conjunction with the Kalman filter, with the former predicting the unknown states $y_{m,i-1}$ and the

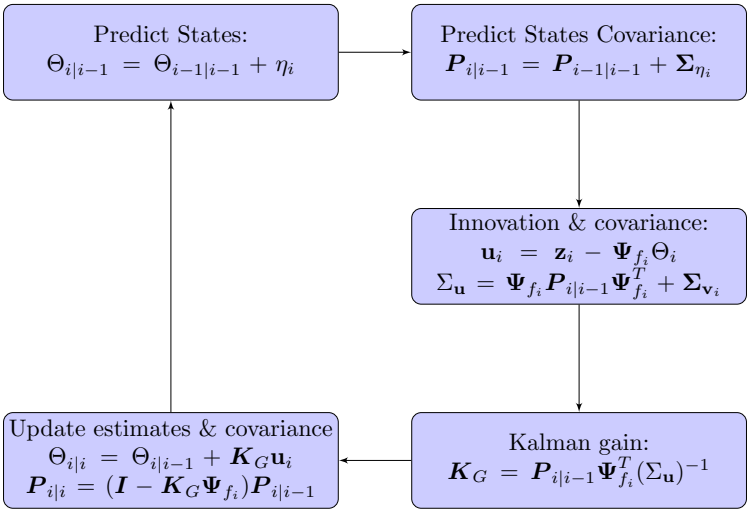

Fig. 7. Kalman filter flow chart.

latter estimating the constant system parameters [25]. It can be observed from the system model described in (2.3) and (2.5) that given the states $y_{m,i-1}$, the state space model becomes linear in the unknown parameters. The linearity of this model and Gaussian noise impairment makes Kalman filter a natural candidate for estimating b_{nm}. Define

$$\Theta \triangleq [b_{11}, \ldots, b_{1N}, b_{21}, \ldots, b_{2N}, \ldots, b_{N1}, \ldots, b_{NN}]^T$$

$$f' \triangleq [f(y_{i,1}) \ldots f(y_{i,N})] \tag{3.8}$$

$$\boldsymbol{\Psi}_{f_i} \triangleq \begin{bmatrix} f' & 0 & 0 & 0 \\ 0 & f' & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & f' \end{bmatrix} \tag{3.9}$$

Then the state and output equations can be expressed as

$$\begin{aligned}\Theta_i &= \Theta_{i-1} + \eta_i \\ \mathbf{z}_i &= \boldsymbol{\Psi}_{f_i}\Theta_i + \mathbf{v}_i\end{aligned} \tag{3.10}$$

where the parameters Θ_i are assumed to evolve from a Gauss-Markov process. The noise η_i denotes the uncertainty in the unknown parameters and its variance is assumed low so that the parameters are almost constant. The update equations for the Kalman filter are as follows:

$$\begin{aligned}\Theta_{i|i-1} &= \Theta_{i-1|i-1} + \eta_i \\ \boldsymbol{P}_{i|i-1} &= \boldsymbol{P}_{i-1|i-1} + \boldsymbol{\Sigma}_{\eta_i} \\ \mathbf{u}_i &= \mathbf{z}_i - \boldsymbol{\Psi}_{f_i}\Theta_i \\ \boldsymbol{K}_G &= \boldsymbol{P}_{i|i-1}\boldsymbol{\Psi}_{f_i}^T(\boldsymbol{\Psi}_{f_i}\boldsymbol{P}_{i|i-1}\boldsymbol{\Psi}_{f_i}^T + \boldsymbol{\Sigma}_{\mathbf{v}_i})^{-1} \\ \Theta_{i|i} &= \Theta_{i|i-1} + \boldsymbol{K}_G\mathbf{u}_i \\ \boldsymbol{P}_{i|i} &= (\boldsymbol{I} - \boldsymbol{K}_G\boldsymbol{\Psi}_{f_i})\boldsymbol{P}_{i|i-1}\end{aligned} \tag{3.11}$$

where $\boldsymbol{K}_G$ is the standard Kalman gain. It must be emphasized that estimate of Θ_i is determined online. Once this estimate becomes available, it is fed back to the particle filter which uses them to calculate the state $\mathbf{y}_i$. This process is depicted in Fig. 5

C. Parameter Selection Using LASSO

The *Least Absolute Shrinkage Selection Operator* (LASSO), proposed by Tibshirani in [31], is a solution to a least squares regression problem with an additional L_1 norm penalty on the parameter vector. The idea of introducing an L_1 norm penalization stems from some inherent short comings associated with an L_2 regularization. While L_2 regularization provides numerical stability and high fidelity, it does not encourage

a parsimonious description and interpretation of the system parameters. The L_1 norm regularization, while retaining many of the useful properties of L_2 norm regularization, provides an additional benefit in that it selects only a subset of parameters that matter in system description. Unlike L_2 penalization where all coefficients generally have non zero values, the L_1 norm regularization of the system parameters employed in LASSO yields a sparse model consisting of only a subset of parameter values. In addition, the subset of parameters so produced comes as close as subset selection schemes do to an ideal subset selector [31]. Hence, LASSO provides an efficient means of system selection. This algorithm has found increasing applications in areas where the parameter vector to be estimated is expected to have a sparse structure e.g., compressed sensing.

In general, a LASSO based least squares regression problem with an L_1 norm penalization can be expressed as

$$\min_{\mathbf{x}} \frac{1}{2}||\mathbf{r} - \mathbf{\Phi}\mathbf{x}||_2^2 + \lambda||\mathbf{x}||_1$$

where $\mathbf{r} \in \mathbb{R}^{N\times 1}$ denotes the received outputs, $\mathbf{\Phi} \in \mathbb{R}^{N\times m}$ is the matrix of regressors and $\mathbf{x} \in \mathbb{R}^{m\times 1}$. The regularization parameter λ affects the trade-off between desired fidelity and sparsity. Using the L_1 norm regularization, the LASSO *shrinks* the unconstrained least squares estimate and therefore, yields a subset of system parameters while setting the irrelevant coefficients to 0. If $\mathbf{\Phi}^T\mathbf{\Phi}$ is invertible, the objective function in the above minimization is convex and finding the LASSO based solution $\mathbf{x}$ amounts to solving a standard convex optimization problem. The selection of the regularizer λ is critical. Several methods have been identified in literature to determine suitable values of λ [31].

After the particle filtering stage delivers the state estimates $y_{i,n}$ for $i = 1, \ldots, I$ and $n = 1, \ldots, N$, they are fed to a LASSO based regression operation Fig. 5. The

LASSO identifies the system parameters b_{nm} using the estimated states and the observed data. A key rationale behind our LASSO based least squares data fitting is that for a particular gene in question, it is related to only a few other genes and as such, many of the constants b_{nm} signifying the regulation relationship among various genes are 0. LASSO allows us to identify only a subset of the system parameters forcing the other irrelevant (or 'weak') interactions to zero. As a result, a more parsimonious and efficient description of the gene regulatory network is obtained [25]. For the nth gene, its observations and estimated states can be stacked using (2.3) and (2.5) as

$$\begin{bmatrix} z_{n1} \\ z_{n2} \\ \vdots \\ z_{nN} \end{bmatrix} = \begin{bmatrix} f(y_{0,1}) & \dots & f(y_{0,N}) \\ f(y_{1,1}) & \dots & \vdots \\ \vdots & \ddots & \\ f(y_{I-1,1}) & & f(y_{I-1,N}) \end{bmatrix} \begin{bmatrix} b_{n1} \\ b_{n2} \\ \vdots \\ b_{nN} \end{bmatrix} + \begin{bmatrix} v_{n1} \\ v_{n2} \\ \vdots \\ v_{nN} \end{bmatrix} \tag{3.12}$$

which can be compactly expressed as

$$\mathbf{z}_n = \mathbf{\Phi}\mathbf{b}_n + \mathbf{v}_n \tag{3.13}$$

LASSO operates on this overdetermined system of equations for the nth gene and produces a parameter vector $\mathbf{b}_n$ by solving

$$\min_{\mathbf{b}_n} \frac{1}{2}||\mathbf{z}_n - \mathbf{\Phi}\mathbf{b}_n||_2^2 + \lambda||\mathbf{b}_n||_1 \tag{3.14}$$

The invertibility of the matrix $\mathbf{\Phi}^T\mathbf{\Phi}$ defined in (3.13) ensures that the objective function is strictly convex and a globally optimal solution is guaranteed by using standard convex optimization techniques [32]. The subset of system parameters so obtained, highlight the relevant gene regulatory relationships among interacting genes while setting others to zero, thus yielding a concise system description.

D. Inference Algorithm

The operation of our algorithm to infer the gene regulatory network is graphically depicted in Fig. 5 and the corresponding pseudocode formulation is summarized in Algorithm 1 [25]. In essence, a particle filtering approach to estimate the states coupled with an online Kalman filter based parameter estimation delivers the estimated states to the LASSO operator. Since genes interact with only a few other genes, the parameter vector is expected to be sparse for a particular gene. LASSO helps us in identifying this subset by solving the constrained optimization problem (3.14).

Algorithm 1 `Gene Network Inference`

Input time series data set z

Initialize I, K, Θ_0

for $k = 1, ..., K$ **do**

    Draw $y_0^k \sim p(y_0|z_0)$

end for

for i = 1,...,I **do**

    for k = 1,...,K **do**

        Draw $w_{i-1}^k \sim p(w_{i-1})$

        Predict $y_i^{*k} \leftarrow g(y_{i-1}, \Theta_{i-1}, w_{i-1})$

    end for

    for k = 1,...K **do**

        Calculate normalized weight ξ_k using (3.7)

    end for

    for k = 1,...,K **do**

        Update $y_i^k \sim \xi_k$

    end for

    Update Θ_i using Kalman filter (3.11)

end for

LASSO: Estimate parameters **b** from **y** and **z** using (3.14)

return

CHAPTER IV

RESULTS

A. Application on Synthetic Data

In this part of the simulation, a 4-gene network is assumed. The data is generated using the model given in (2.3). Table 1 gives the true values of the parameters b_{nm}. The values of the parameters contain various zeros which is also true for a real gene network as we know that the gene network is sparse. The variance of the system noise $v_n \sim \mathcal{N}(0, \sigma_v^2)$ is taken to be 10^{-4}. The gene interactions are estimated using the proposed method and compared with the extended Kalman filter approach used in [24]. $K = 200$ number of particles is used for particle filter simulation.

The values of estimated parameters using both the algorithms is shown in Table I. It is observed that the proposed algorithm achieves a good estimate on the constant parameters. The simulation is repeated for varying values of the system parameters and number of genes and our approach gives better estimation performance than EKF. To keep the thesis concise, the figures are not shown here. The estimated values of synthetic time series data are compared using the Mean square error criterion for both the algorithms and the results are shown in Fig. 8 It can be seen that particle filter gives lower MSE than EKF for a wide range of values of measurement noise variance.

B. Application on Real Biological Data

1. Network Modeling for Malaria Time Series Data

The proposed algorithm is now tested on real data using gene expression time series data for plasmodium falciparum. This data set consists of 48 time points for 530

Table I. True Parameters and Estimated Values Using EKF and PF+Lasso

Θ	True Values	EKF	PFL
b_{11}	3.2	2.8395	3.1471
b_{12}	-4.13	-2.6341	-4.1460
b_{13}	0.02	-10.2405	0.1036
b_{14}	0.02	9.6628	-0.0304
b_{21}	0.01	-1.5092	0.1911
b_{22}	4	2.2126	4.0443
b_{23}	-1.2	11.2065	-1.4836
b_{24}	1.1	-10.4415	1.3312
b_{31}	4.2	4.1983	4.2315
b_{32}	0.02	-0.3805	0.0184
b_{33}	0.01	4.0591	-0.0116
b_{34}	-3	-7.5617	-2.9653
b_{41}	4.05	5.3747	3.9680
b_{42}	0.01	-0.4277	-0.0457
b_{43}	0	2.5036	0.3070
b_{44}	-5.5	-7.7780	-5.7107

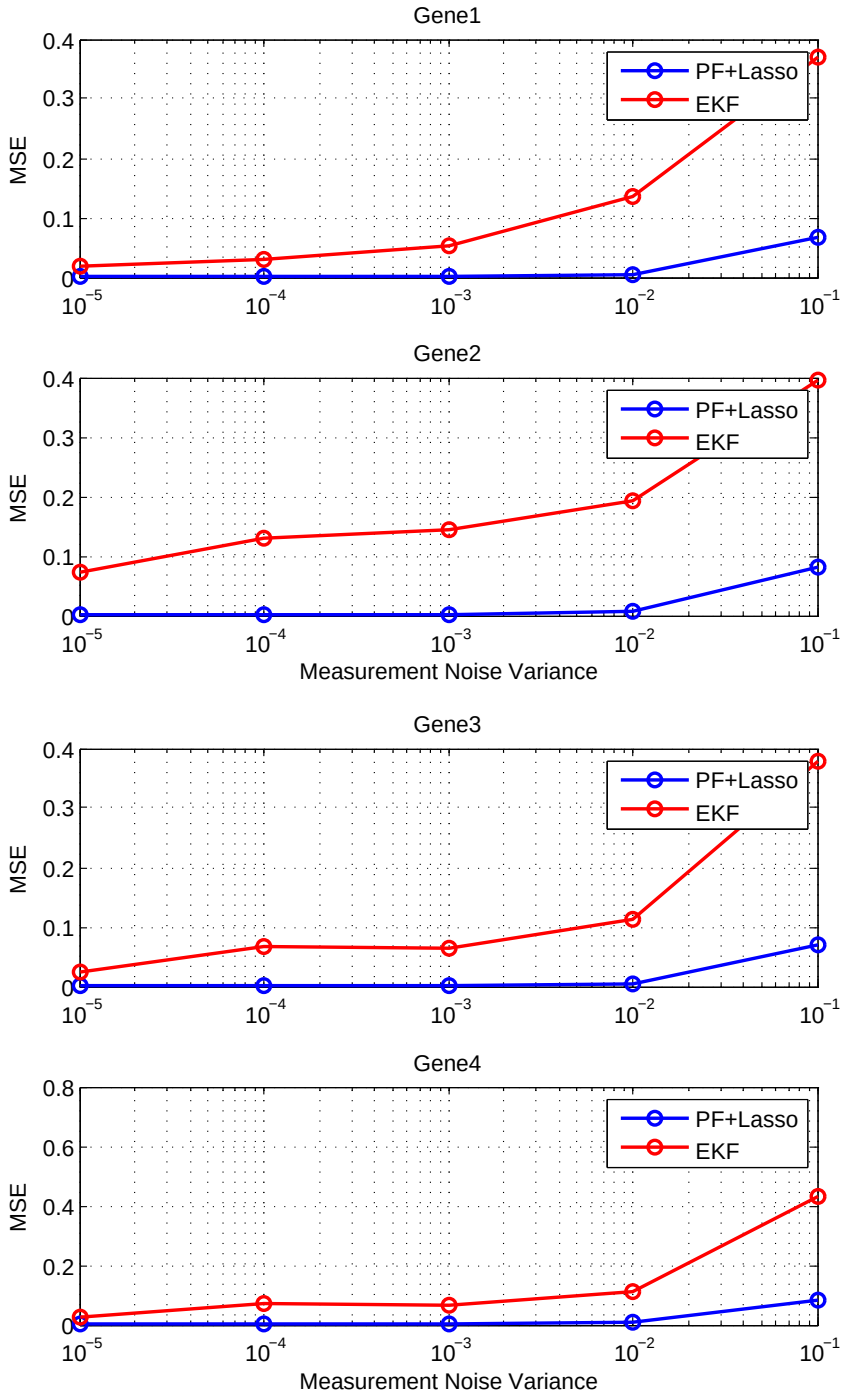

Fig. 8. MSE performance comparison between extended Kalman filter and particle filter using synthetic data.

genes [33]. For the purpose of this simulation, six genes data is considered. The system noise is taken to be $v_n \sim \mathcal{N}(0, 10^{-4})$. By using the assumed system model, the unknown states and parameters are estimated using the proposed algorithm and the EKF approach [24]. The estimation of the observed values by both the algorithms are compared using the Mean Square Error criterion. The observation noise $w_n \sim \mathcal{N}(0, \sigma_w^2)$ variance ranges from 10^{-5} to 10^{-1}. It is found that as the noise variance increases, MSE for EKF starts increasing. Particle filter, however, shows very low MSE for the entire range of observation noise variances as shown in Fig. 9 It can be inferred that our method models the network efficiently and is robust to changes in noise.

2. Network Modeling for Worm Time Series Data

The time series data obtained during C. Elegans embryo development is used in this comparison [34], [35]. The data set consists of 123 time points. Eight genes are considered for this simulation which are pal-1, tbx-8, elt-1, elt-3, nhr-25, cwn-1, nob-1 and vab-7. The performance evaluation criterion is the same as before i.e. Mean square error and it shown in Fig. 10,11. The variation of observation noise $w_n \sim \mathcal{N}(0, \sigma_w^2)$ is from 10^{-5} to 10^{-2}. The system noise variance is kept the same as in the previous simulation. Fig. 12,13 show the gene expression time series data and its predicted values for observation noise variance of 10^{-3}. It can be seen that particle filter provides a very nice fit to the data.

The algorithm proposed in this thesis was tested on synthetic data and real biological data. It shows promising results on all these data sets. We can thus conclude, that particle filter can provides a viable alternative to EKF for modeling gene regulatory networks while giving a better performance.

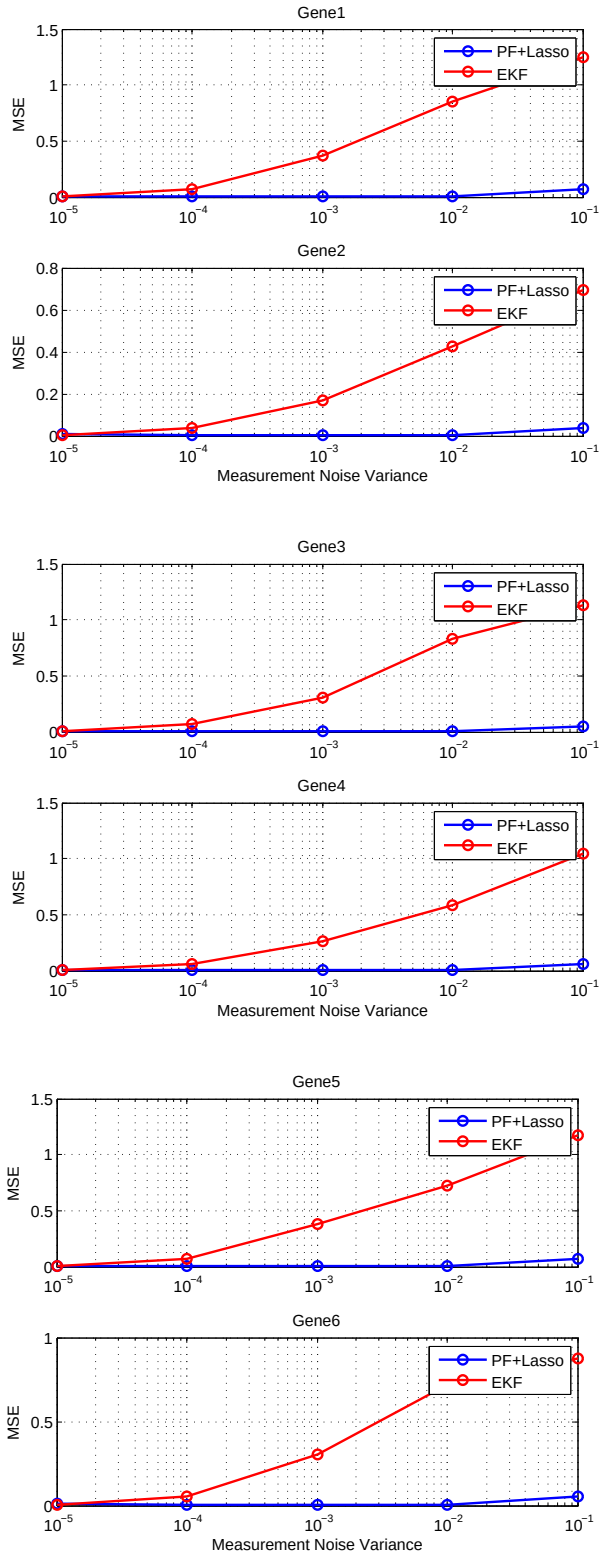

Fig. 9. MSE performance comparison between extended Kalman filter and particle filter using Malaria time series data.

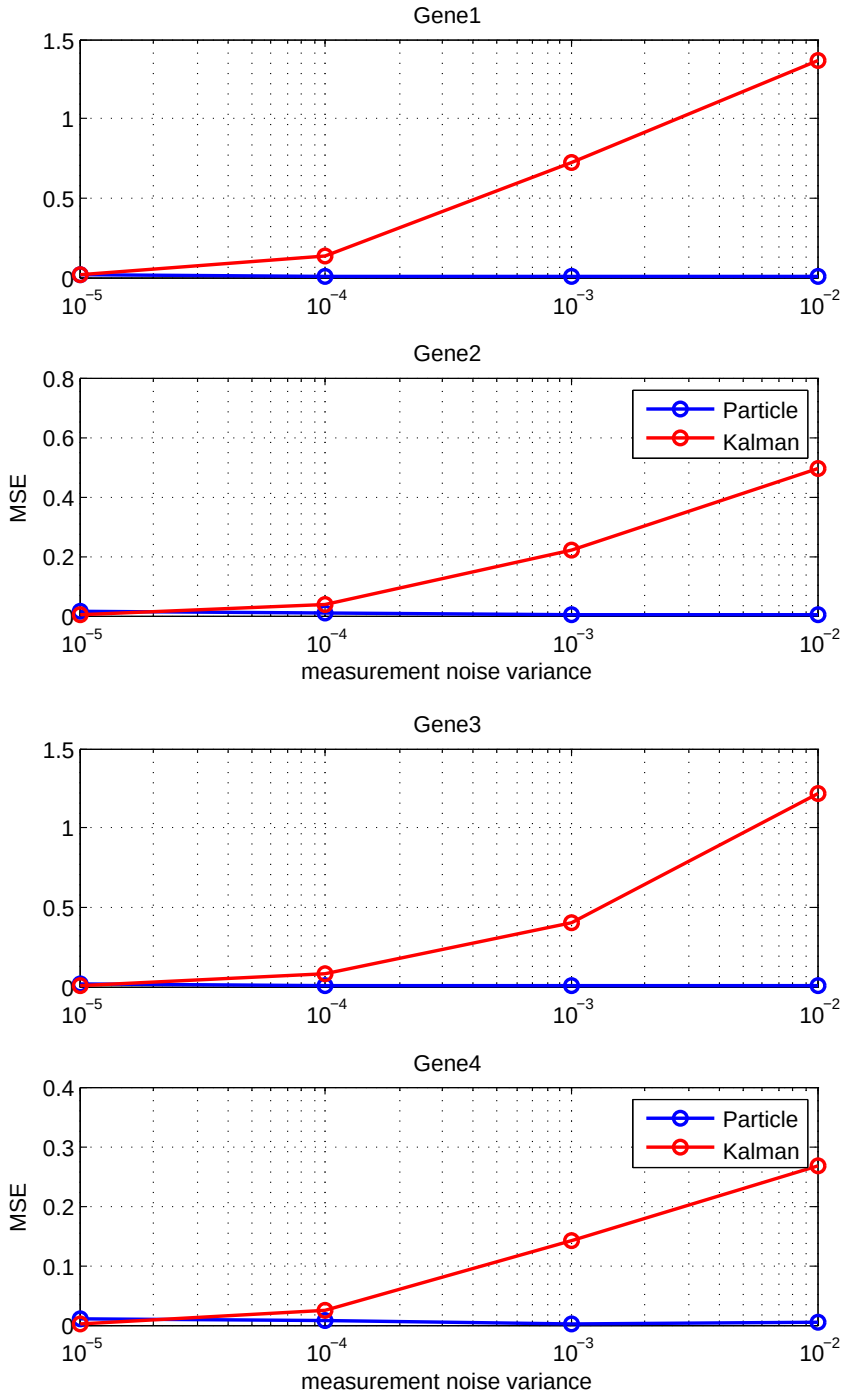

Fig. 10. MSE performance comparison for gene 1-4, between extended Kalman filter and particle filter for C. Elegans time series data.

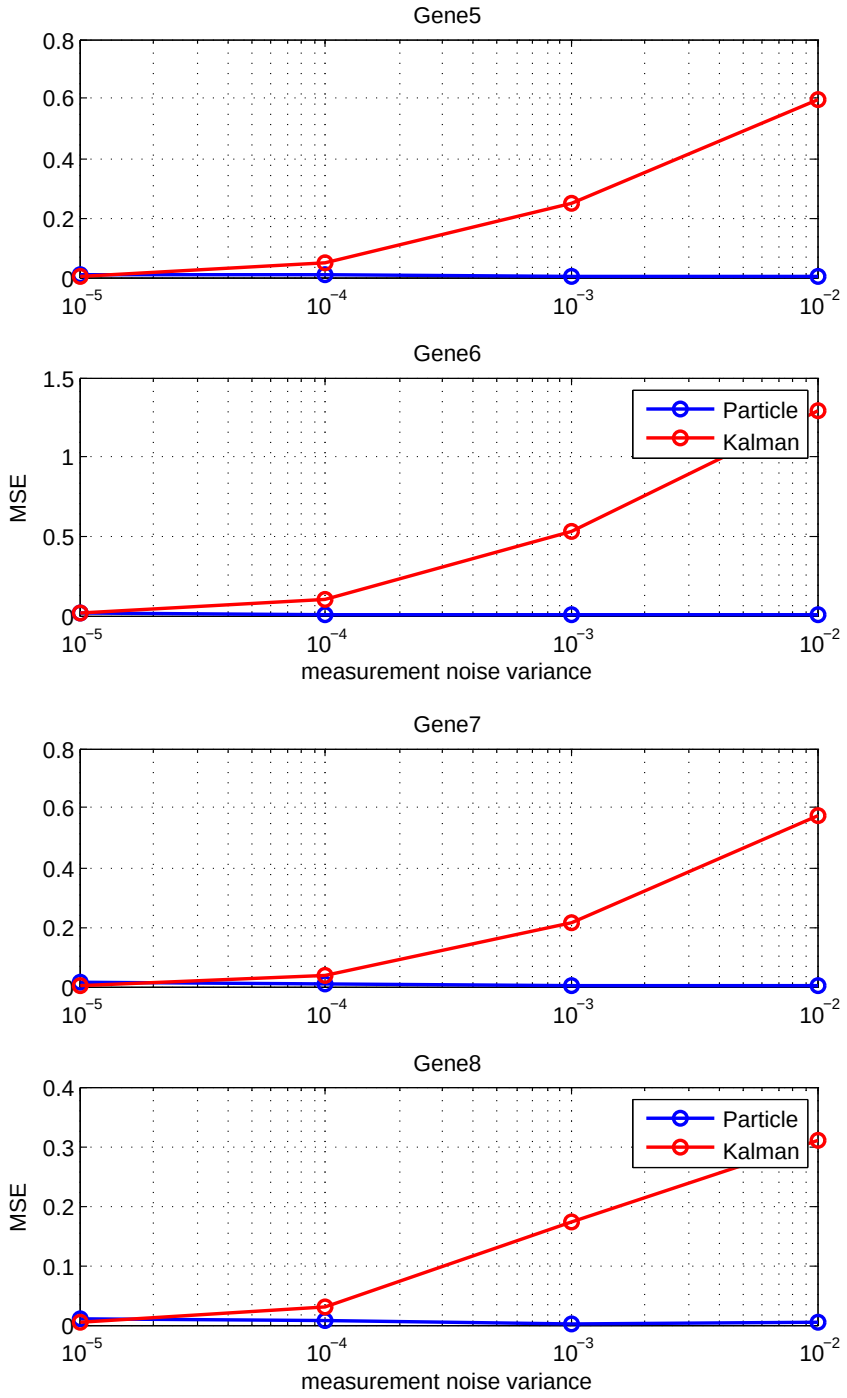

Fig. 11. MSE performance comparison for gene 5-8, between extended Kalman filter and particle filter for C. Elegans time series data.

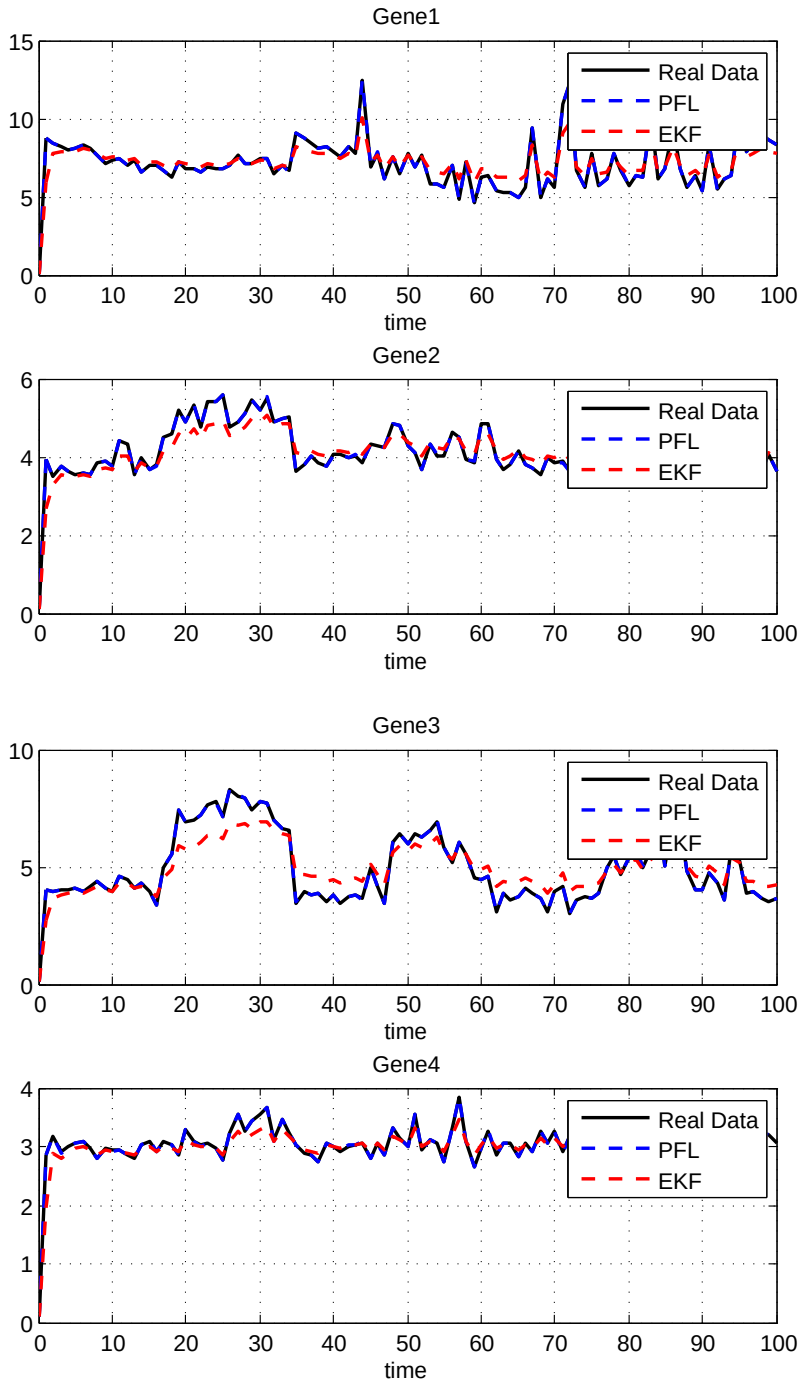

Fig. 12. Observed and predicted gene expression for gene 1-4 of C. Elegans time series data.

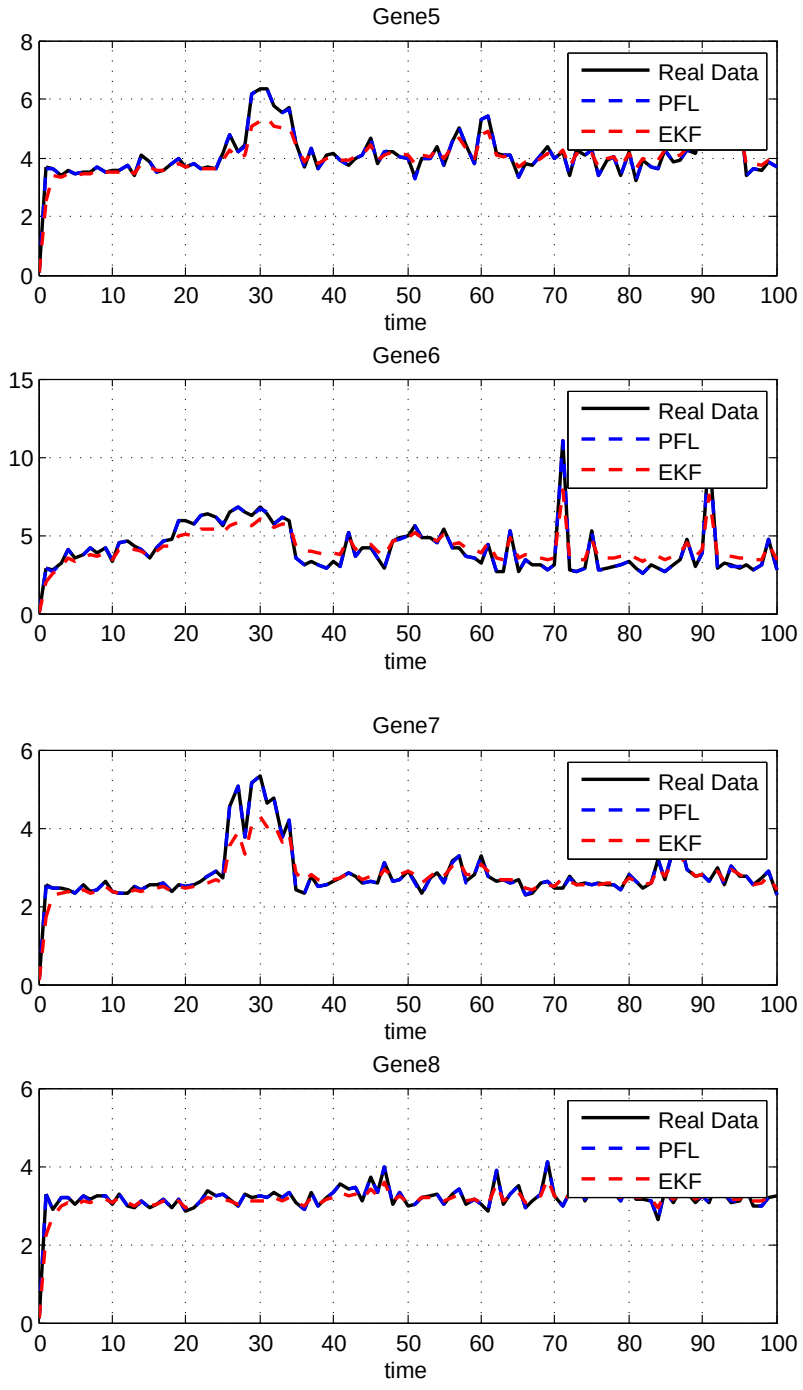

Fig. 13. Observed and predicted gene expression for gene 5-8 of C. Elegans time series data.

CHAPTER V

CONCLUSIONS

"I can't be as confident about computer science as I can about biology. Biology easily has 500 years of exciting problems to work on. It's at that level".

-Donald Knuth [27].

Precise modeling of a gene regulatory network is a critical component in understanding complex gene interactions. As a clearer picture of this interactions emerges, several potential applications can be realized. In particular, knowing which gene triggers a specific genetical disorder can help us control it by deactivating this gene. It can also have a tremendous impact on diagnosis and drug designs. Gene regulatory networks are graphical models used to depict gene interactions. Quantifying the correlation among genes is the next logical step that requires efficient network inference algorithms. In order to extract useful information from large amounts of biological data, it is important that computationally efficient algorithms are devised.

This thesis considers the modeling and learning of gene regulatory networks using a nonlinear model which is a more general characterization of gene interactions. The gene network is modeled using a state space approach and particle filtering is used for state estimation. The parameters regulating the interaction among genes are supplied by an online Kalman filter. Since the parameter vector is frequently sparse, a subset of these parameters signifying only the relevant system coefficients are identified via a LASSO based least squares regression process. Extensive performance

evaluations demonstrate that this particle filtering based approach outperforms the extended Kalman filtering in terms of MSE Criterion. The results are proved using synthetic data as well as Microarray data for Malaria and C. Elegans gene expression time series. Our algorithm can, therefore, serve as a natural framework for modeling gene regulatory networks.

A. Future Work

Several avenues for further research can be identified.

- In future, we intend to use our algorithm to infer gene regulatory interactions with the true experimental results reported in literature for both the malaria and worm data.
- More sophisticated pruning techniques should be considered to infer gene connections.
- It can also be interesting to study other nonlinear modeling techniques. In addition, different nonlinear modeling functions can be investigated besides the sigmoid squash function e.g., polynomial functions.

REFERENCES

[1] H. Kitano, "Computational systems biology," *Nature*, vol. 420, pp. 206-210, Nov 2002.

[2] X. Cai, and X. Wang, "Stochastic modeling and simulation of gene networks," *IEEE Signal Processing Magazine*, pp. 27-36, Jan 2007.

[3] Y. Huang, I. M. Tienda-Luna, and Y. Wang, "Reverse engineering gene regulatory network," *IEEE Signal Processing Magazine*, pp. 76-97, Jan 2009.

[4] H. de Jong, "Modeling and simulation of genetic regulatoy systems: A literature review," *J. Computational Biology*, vol. 9, no. 1, pp. 67-103, 2002.

[5] H. Hache, H Lehrach, and R. Herwig, "Reverse engineering of gene regulatory networks: a comparative study," *EURASIP Journal on Bioinformatics and Systems Biology,* Article ID 617281, pp. 1-12, 2009.

[6] F. Markowetz and R. Spang, "Inferring cellular networks - a review," *BMC Bioinformatics*, vol. 8, p. S5, Sept. 2007.

[7] T. Schlitt and A. Brazma, "Current approaches to gene regulatory network modeling," *BMC Bioinformatics*, vol. 8, p. S9, Sept. 2007.

[8] X. Zhou, X. Wang, and E. R. Dougherty, "Construction of genomic networks using mutual-information clustering and reversible-jump Markov-cain-monte-carlo predictor design," *Signal Processing,* vol. 83, pp. 261-274, 2002.

[9] H. Toh, and K. Horimoto, "Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling," *Bioinformatics*, vol. 18, pp. 287-297, 2002.

[10] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring connectivity of genetic regulatory networks using information-theoretic criteria," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 6, no. 3, pp. 410-419, Apr-Jun 2008.

[11] W. Zhao, E. Serpedin, and E. R. Dougherty, "Inferring gene regulatory networks from time series data using the minimum description length principle," *Bioinformatics,* vol. 22, pp. 2129-2135, 2006.

[12] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *J. Computational Biology*, vol. 7, pp. 601-620, 2000.

[13] N. Friedman, "Inferring cellular network using probabilistic graphical models," *Science,* vol. 33, pp. 799-805, 2004.

[14] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the boolean network model," in *Proc. Pacific Symp. Biocomputing,* 1999, vol. 4, pp. 17-28.

[15] I. Schmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics,* vol. 18, pp. 261-274, 2002.

[16] T. Tian, and K. Burrage, "Stochastic neural network models for gene regulatory networks," in *Proc. 2003 IEEE Congress Evolutionary Computation,* 2003, pp. 162-169.

[17] K. Murphy, and S. Mian, "Modeling gene expression data using dynamic bayesian networks," Technical Report, Berkeley, CA: Berkeley Univ., 1999.

[18] Y. Zhang, Z. Deng, H. Jiang and P. Jia, "Inferring gene regulatory networks from multiple data sources via a dynamic Bayesian network with structural EM," in *Proc. Int'l Conf. Data Integration in the Life Sciences,* pp. 204-214, 2007.

[19] M. Quach, N. Brunel, and F. d'Alche-Buc, "Estimating parameters and hidden variables in non-linear state-space models based on ODEs for biological networks inference," *Bioinformatics,* vol. 23, no. 23, pp. 3209-3216, 2007.

[20] F. Wu, W. Zhang, and A. J. Kusalik, "Modeling gene expression from microarray expression data with state-space equations," *Proc. Pacific Symp. Biocomputing,* 2004, pp. 581-592.

[21] R. Yamaguchi, R. Yoshida, S. Imoto, T. Higuche, and S. Miyano, "Finding module-based gene networks with state-space models," *IEEE Signal Processing Magazine*, pp. 3746, Jan 2007.

[22] L. Qian, H. Wang, and E. R. Dougherty, "Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering," *IEEE Transactions on Signal Processing,* vol. 56, no.7, pp. 3327-3339, July 2008.

[23] A. Corigliano, and S. Mariani, "Parameter identification in explicit sturctural dynamics: performance of the extended kalman fitler," *Computer Methods in Applied Mechanics and Eng.,* vol. 193, pp. 3807-3835, 2004.

[24] Z. Wang, X. Liu, Y. Liu, J. Liang, and V. Vinciotti, "An extended Kalman filtering approach to modeling nonlinear dynamic gene regulatory networks via short gene expression time series," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 6, no.3, pp. 410-419, July-Sep 2009.

[25] A. Noor, E. Serpedin, and M. Nounou, "Modeling gene regulatory networks from time series data using particle filtering," submitted to *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* May 2011.

[26] A. Noor and E. Serpedin, "Modeling gene regulatory network from time series data using particle filtering," *International Workshop on Genomic Signal Processing,* Bucharest, Romania, June 2011.

[27] http://en.wikiquote.org/wiki/Donald_Knuth.

[28] B. Ristic, S. Arulampalam and N. Gordon, *Beyond the Kalman Filter - Particle Filters for Tracking Applications.* Artech House, Boston, MA, USA, 2004.

[29] P. M. Djuric, J. H. Kotecha, J. Zhang, Y Huang, T Ghirmai, M. F. Bugallo and J. Miguez, "Particle filtering," *IEEE Signal Processing Magazine,* pp. 19-38, Sep 2003.

[30] O. Cappe, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential monte carlo," in *Proc. of IEEE,* May 2007, pp. 899-924.

[31] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. Royal Statist. Soc B.*, vol. 58, no. 1, pp. 267-288, 1996.

[32] S. Boyd, and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, The Edinburgh Building, Cambridge, UK, 2004.

[33] Z. Bozdech, M. Llinas, B. L. Pulliam, E. D. Wong, and J. Zhu, "The transcriptome of the intraerythrocytic developmental cycle of plasmodium falciparum, *PLoS Biology,* vol. 1, no. 1, pp. 85-100, 2003.

[34] L. R. Baugh, A. A. Hill, J. M. Claggett, K. Hill-Harfe, J. C. Wen, D. K. Slonim, E. L. Brown, and C.P. Hunter, "The homeodomain protein PAL-1 specifies a lineage-specific regulatory network in the C. Elegans embryo, *Development,* vol. 132, pp. 1843-1854, 2005.

[35] M.F. Maduro and J.H. Rothman, "Making worm guts: the gene regulatory network of the caenorhabditis elegans endoderm," *Developmental Biology,* vol. 246, pp. 68-85, 2002.

VITA

Amina Noor received the B.E. and M.S. degrees in electrical engineering from the National University of Sciences & Technology, Islamabad, Pakistan, in 2006 and 2008, respectively. Her research interests include genomic signal processing, pattern recognition, and algorithm design.

Ms. Noor may be reached at Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX 77843. Her email address is amina@neo.tamu.edu.

The typist for this thesis was Amina Noor.