CONSTRAINED EXPECTATION-MAXIMIZATION (EM), DYNAMIC ANALYSIS, LINEAR QUADRATIC TRACKING, AND NONLINEAR CONSTRAINED EXPECTATION-MAXIMIZATION (EM) FOR THE ANALYSIS OF GENETIC REGULATORY NETWORKS AND SIGNAL TRANSDUCTION NETWORKS

A Dissertation

by

HAO XIONG

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

DOCTOR OF PHILOSOPHY

December 2008

Major Subject: Computer Science

CONSTRAINED EXPECTATION-MAXIMIZATION (EM), DYNAMIC ANALYSIS, LINEAR QUADRATIC TRACKING, AND NONLINEAR EXPECTATION-MAXIMIZATION (EM) FOR THE ANALYSIS OF GENETIC REGULATORY NETWORKS AND SIGNAL TRANSDUCTION NETWORKS

A Dissertation

by

HAO XIONG

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

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Yoonsuck Choe
Committee Members,	Ricardo Gutierrez-Osuna
	Dezhen Song
	Juergen Hahn
	Ruzong Fan
Head of Department,	Valerie E. Taylor

December 2008

Major Subject: Computer Science

ABSTRACT

Constrained Expectation-Maximization (EM), Dynamic Analysis, Linear Quadratic Tracking, and Nonlinear Constrained Expectation-Maximization (EM) for the Analysis of Genetic Regulatory Networks and Signal Transduction Networks. (December 2008)

Hao Xiong, B.S.,University of Houston
Chair of Advisory Committee: Dr. Yoonsuck Choe

Despite the immense progress made by molecular biology in cataloging and characterizing molecular elements of life and the success in genome sequencing, there have not been comparable advances in the functional study of complex phenotypes. This is because isolated study of one molecule, or one gene, at a time is not enough by itself to characterize the complex interactions in organism and to explain the functions that arise out of these interactions. Mathematical modeling of biological systems is one way to meet the challenge.

My research formulates the modeling of gene regulation as a control problem and applies systems and control theory to the identification, analysis, and optimal control of genetic regulatory networks. The major contribution of my work includes biologically constrained estimation, dynamical analysis, and optimal control of genetic networks. In addition, parameter estimation of nonlinear models of biological networks is also studied, as a parameter estimation problem of a general nonlinear dynamical system. Results demonstrate the superior predictive power of biologically constrained state-space models, and that genetic networks can have differential dynamic properties when subjected to different environmental perturbations. Application of optimal control demonstrates feasibility of regulating gene expression levels. In the difficult problem of parameter estimation, generalized EM algorithm is deployed, and a set of explicit formula based on extended Kalman filter is derived. Application of the method to synthetic and real world data shows promising results.

To My Family

ACKNOWLEDGMENTS

During the process of completing my dissertation, there have been many people without whom I would not have finished the process. My advisor, Dr. Yoonsuck Choe, has always guided me with patience, wisdom, foresight, and kindness. His example in scholarship, painstaking attention to detail, and his devotion to science is no less eloquent than his words. He is ever ready to help me, no matter how insignificant the question. His outstanding scientific ability, combined with his willingness to help, makes him the perfect advisor. I have learned a lot from him, and I know I still have a lot to learn from him.

I also would like to express my thanks to Dr. Ricardo Gutierrez-Osuna, Dr. Dezhen Song, Dr. Juergen Hahn, and Dr. Ruzong Fan for spending their valuable time to serve on my committee and offering me their comments and suggestions for improvement.

My friends at A&M gave me tips about everything from how to conduct research to daily living matters. Their friendship made the long and arduous process of completing my degree enjoyable. Yingwei Yu, Yueping Zhang, Nan Zhang, and Jacob Foshee are those with whom I had more frequent contact and most to thank.

My family gave me more than support; they gave me encouragement in the darkest hours. My father is unfailing in his steadfast belief in me and in transmitting that belief. My mother is a paragon of kindness and silent assistance. And above all, my tender, supportive wife is one I turn to for my intimatest thoughts and most far-reaching plans, and has supported and comforted me. I dedicate this dissertation to her and everyone who has made this possible.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

A. Research Objectives

A living individual is more than a conglomeration of cells, more than a sum of its parts; it is a complex biological system. It consists of many interrelated, interacting, and interfering components. It should be studied as it exists, as a whole, for it can be studied not only in terms of the mechanistic, molecular-level components but also in terms of integrated operations [1]. Phenotype, the outwardly visible and measurable characteristics of individuals, is often the emergent property of complex interactions between genes, proteins, and other biochemical elements. Most phenotypic variations, including those involved in complex diseases and differential drug responses, are generated by integrated actions of multiple genetic and environmental factors that are organized into a hierarchical biological network through dynamic, epigenetic, and regulatory mechanisms [2]. Concentrations of metabolites, genes or proteins, and their dynamics largely determine the phenotype of the cells and are the fingerprints of cellular physiology. Germline or somatic mutations lead to subsequent transcriptional and translational alterations which will affect the phenotype of the cells and cause diseases. Therapeutic interventions such as radiation, drug, and gene therapy try to alter metabolites, proteins, and expressions of genes from an undesired state or abnormal state to a desired or normal state. Designing efficient drugs that have few side-effects and that can improve production of food, animal feed, enzymes, advanced biopolymers, chemicals, and pharmaceuticals require us to perform systems identification on dynamic biological networks, study their dynamic behaviors

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

such as stability and controllability, analyze their steady-state and transient response, and to develop rules for optimally controlling the response and behavior of biological systems.

The goal of this dissertation is to develop computational models and algorithms for systems identification of genetic regulatory networks and signal transduction networks, to investigate the networks' dynamic properties, and to design strategies for control systems to optimally alter gene expressions. The specific objectives are as follows:

1. Apply linear state-space models to the study of genetic regulatory networks, and develop new system identification methods that produce biologically motivated constrained models of genetic regulatory networks. The reason for performing system identification first is to firmly ground all my subsequent studies in real world data in order to have real world significance.

2. Apply differential dynamical analysis of genetic networks under different environmental perturbations as a means to uncover the differential dynamical behaviors of genetic regulatory networks. The working hypothesis is that diseased cells are fundamentally different from healthy cells in their dynamic behaviors.

3. Formulate genetic-regulatory-network therapy as a linear quadratic optimal tracking problem and apply optimal control to the problem.

4. Develop and apply nonlinear parameter estimation methods to models of signal transduction networks and/or genetic regulatory networks, which could be subjected to time-delays, in order to obtain hard-to-meausure reaction rates and constants.

B. Significance of the Proposed Objectives

Because genes code proteins and proteins largely determine cells' functioning, a genome-wide expression profile is a nice window into the biochemical state of cells. With the completion of human genome project and the availability of an increasing number of organisms' genome maps, microarrays can offer genome-wide expression profiles for an increasing number of organisms. Gene regulations can be seen as offering a bird's-eye view of the biochemical state of cells. Therefore, a lot of effort in systems biology has concentrated on studying genetic regulatory networks, particularly the dynamical models of genetic networks.

Costrained estimation: I will discuss the rational for my choice of modeling framework in the next chapter, so I will just note here that it is driven by our limited knowledge of mechanisms of gene regulation and biochemical reactions of cells in general. But even for coarse-grained models, there still are not enough data to definitely learn a predictive model, even for a small system. One way we can deal with this is by augmenting available data with additional biological knowledge to restrict search space. The biologically motivated constraint I chose is the known structure of genetic networks, which is usually gathered through direct experiments. This forces dynamical models to be sparse and therefore a large number of parameters become constants. This work also fits into the general structure learning framework of dynamical Bayesian networks, where the parameter estimation is predicated on the current structure, and the best structure is the one that generates best estimated parameters. In other words, my constrained parameter estimation method can serve as the inner loop of any structure learning algorithm for dynamical Bayesian networks.

Dynamical analysis: Although model learning garners the attention of many researchers, dynamical analysis of learned models is also attracting attention [3, 4, 5, 6]. However, comparative study of dynamical gene regulation has not drawn the same degree of interest as its static counter-part, the differential expression of genes and genetic networks. Differentially expressed genes are a valuable tool in genetic study. Much of what we know about contributions of genes to diseases or physiological traits comes from the comparative study of healthy and sick cells' gene expression profiles; it is a giant field in genetics. It is expected that dynamical differential analysis can offer even more insights. People have used dynamical property for dynamical analysis [4, 7], and I propose we analyze differential dynamical properties to uncover different dynamical behaviors of sick and healthy cells. We know that healthy and sick cells are fundamentally different, and that the difference would manifest itself dynamically. Uncovering differential dynamical behaviors is one more step toward understanding the role of gene regulation dynamics in pathogenesis.

Optimal control: Even with dynamical analysis, treatment through genes is a daunting challenge, as the recent high-profile failures of gene therapy demonstrated [8, 9]. The big problem is that gene regulation is complex, interacting, and interfering. Targeting one or few genes is usually not efficacious and is often riddled with debilitating side-effects. Instead, we should target a genetic network. I propose to apply optimal tracking (manipulation of inputs in order to have outputs track a reference trajectory) to the manipulation of genetic networks. If a sick genetic network can be made to behave like a healthy one, that should be an effective treatment. I chose a particular framework, optimal control, because it has many tools so that we can adapt mathematical formulation to the particular situation and its peculiar requirements. If we want to minimize dosage, we can incorporate that into the performance index; or if we have hard limits on dosage, that can be considered a constraint on feasible inputs. The flexibility and versatility of optimal control makes it a potentially powerful tool in systems biology.

Nonlinear parameter estimation: Linear models of gene regulation have the virtue of simplicity and analytic tractability, but they lack fine-grained details and offer poor mechanistic insights. As our biological knowledge grows, our available data become more plentiful, we inevitably want to have detailed, mechanistic models of genetic regulatory networks, as well as other biological networks such as signal transduction networks or metabolic networks. In fact, signal transduction networks tend to be modeled as nonlinear ordinary differential equations [10, 11, 12, 13]. Nonlinear state-space dynamic models can model the nonlinear biochemical reactions when there are many hidden variables difficult or impossible to measure directly. Nonlinear state-space models can also make possible more detailed models of genetic regulatory networks, such as those that take into account transcriptional factors binding. However, because functional activities of cells tend to be compartmentalized and ordinary differential equations (ODEs) presume all elements are well mixed, time-delays sometimes need to be introduced into models to account for the transport process. For example, transcription takes place in the nucleus for eukaryotic cells, but translation takes place in the cytoplasm, so mRNAs need to travel out of the neucleus before translation can take place. Nonlinearity and time-delay are hurdles we must face as we seek ever more detailed, mechanistic understanding of biological networks.

CHAPTER II

BACKGROUND

Biology and mathematics are no strangers. Wiener considered his cybernetics to be a unifying work of engineering and biology, and Kalman alluded to biological analogies in his work [14]. The field of mathematical biology has been laboring on metabolic pathways, heart electrical patterns, and neural and circadian oscillations [14]. As for the current wave of quantitative studies in systems biology, the tipping point was genome sequencing, which enabled high-throughput experimental methods like microarrays [15], protein arrays [16], mass spectrometry [17], and nuclear magnetic resonance (NMR) [18, 19], while yeast two-hybrid assay or chromatin immunoprecipitation allowed the detection of protein-protein or protein-DNA interactions [20, 5]. Then there are promising breakthroughs such as nanopore gene sequencing [21] or nanotechnology that enable experimental methods to examine the physiological functions [22].

A. Background on Gene Regulation

In this section, I will briefly discuss some basic terms about genes and gene regulation, since they will be used repeatedly throughout this proposal.

DNA can be seen as a string of symbols (A, T, G, C) that carry genetic information. To decode this genetic information, cells translate different parts, or regions, of DNA into different proteins; the sequence of the amino acids which make up the end-product, the protein, depends on the sequence of symbols in the generating region of DNA. An intermediate product of DNA as it is translated into protein is RNA, and the process of generating RNA from DNA is called transcription. For our purpose, we will consider the specific region of DNA that is involved in producing a protein to be one gene. In the human genome, there are about 20,000 to 25,000 genes.

In a living cell, if a gene is generating RNA, or being transcribed, then it is being expressed. Whether a gene is expressed or not depends on the biochemical needs within the cell and external signals from outside the cell. When and how much a gene is expressed is tightly controlled by an intricate cellular machinery, and this tight control is called gene regulation. Microarrays are one method to measure how much a gene is expressed, in terms of concentration levels of RNAs, also called expression levels.

A gene can regulate another gene by producing proteins that facilitate or inhibit the other gene's transcription and/or translation, by being in close proximity of each other, or by competing for key proteins that are involved in transcription or translation. Some genes are auto-regulatory, that is, they regulate themselves. Except for auto-regulatory genes, gene regulation has direction, so that if gene A regulates gene B, there is no automatic guarantee that gene B also regulates gene A. In addition to direction, there is also intensity to gene regulation, where one gene can strongly or weakly impact another gene's expression levels. The result of gene regulation is fluctuation in the expression levels of regulated genes, which means we can consider a gene to be a time-varying variable whose values determine the gene's expression levels. If we represent the regulatory relations of a group of genes as a directed graph, where each gene is a vertex and each directed edge represents gene regulation with the regulatory intensity as the edge's value, we get a transcriptional network, also called a genetic regulatory network.

B. Two Approaches to Modeling

There are many approaches to dynamic modeling of biological systems, but they can be broadly classified into two classes: the top-down or the bottom-up approach [23, 24]. The bottom-up approach starts from the bottom with basic biochemical reactions that are often represented as reaction kinetics. This approach is popular in modeling signal transduction networks, some of which have been studied for decades and are relatively well characterized. The reaction equations are usually ordinary differential equations (ODEs) that approximate the discrete, stochastic, biochemical reactions by assuming that there are enough molecules for a continuous approximation and that the reactants are well mixed to have near uniform concentration. Kinetic parameters are best directly measured, and when direct measurements are impossible, the parameters can be estimated, although the estimated values are not entirely trusted by some scientists. For example, one project, the Silicon Cell initiative, does not accept estimated parameters for their models [25]. But regardless of the source of kinetic parameters, the objective of the bottom-up approach is to discover novel systematic properties and to predict system behavior. As more experiments are performed and more data are generated, mechanistic models of subsystems can be merged into larger systems, where the ultimate goal is to derive a mechanistic, experimental model of biological systems.

There are two principle drawbacks to the bottom-up approach that prevents its application everywhere in biology. One is that the bottom-up approach requires a lot of information which is not available for most biological systems. To begin, it requires a complete list of biochemical elements participating in the system, and then it needs complete stoichiometric information and the kinetic parameter values. Second, the number of reaction equations increase so dramatically as the modeled system gets larger, that simulating even a fraction of the approximately 25,000 genes and 100,000 proteins in a human cell would overwhelm the most powerful computer today and in the foreseeable future. However, as Bornholdt noted [26], “we do not have to retrace all the details of the biochemistry. ... a clever way to throw away details may be the most important part of model building.” Bottom-up approach does not model every detail of biochemical reactions, and the top-down approach throws out even more.

If the bottom-up approach to biological modeling is considered to be mechanistic, then the top-down approach should be considered to be causal modeling or “phenomenological,” that is, only causal relations and the observed phenomena are modeled, not the underlying mechanisms. The advent of the top-down approach can be associated with the availability of high-throughput data and the increasing availability of complete genome sequences for diverse organisms. Starting with genome-wide measurements, often concentrations of mRNAs, the bottom-up approach seeks to discover novel correlations or causal relations between genes and/or proteins, or insights about the relations. There is no pretense that the causal relationships represent any kind of mechanistic, direct molecular reactions. Often the causal relationships are modeled as networks, as gene regulations are modeled as genetic regulation networks. The top-down approach also differs from the bottom-up approach in the choice of the underlying models. While the bottom-up approach often uses ODEs, the top-down approach has seen many kinds of models from Boolean networks to Bayesian networks to nonlinear ODEs. The particular choice depends on the kind of questions the researcher wants to ask and the kind of available data the researcher has, but among models using differential or difference equations, “the linear functions have proved to be the most versatile . . . because they reduce the number of parameters and avoid problems with overfitting” [24]. For the exact reason as quoted above, the genetic networks models in this dissertation are all linear (the nonlinear parameter estimation chapter uses only signal transduction networks or artificial systems).

The next two subsections will briefly discuss some existing models of genetic networks and existing efforts in estimating the parameters of these models.

C. Comparison of Models in Systems Biology

This subsection outlines some popular quantitative models of genetic regulatory networks. They include the state-space model, Boolean networks, Bayesian networks, ordinary and partial differential equations.

1. State-space Models

The state-space model as defined in control/systems theory is different from the state space model defined in computer science or graph theory. The key here is the definition of states. It is defined as the information needed at time t_0 such that, together with all subsequent input for $t > t_0$, the output can be uniquely determined [27]. Define any systems by their inputs, denoted by μ, outputs, denoted as Y, and states, X, and the general form of a state space model can be defined as

$$\begin{aligned}\dot{X}(t) &= f(t, X(t), \mu(t))\\ Y(t) &= g(t, X(t), \mu(t)),\end{aligned}$$

where f and g are functions of time, states, and inputs. A special case where f and g are linear and t only appear as a variable of X and μ is called linear time-invariant system (LTI), which will be treated in chatper III. If the inputs and outputs are both continuous-time, then the model is a continuous-time model; if the inputs and outputs are both discrete-time, then it is a discrete-time system. All the systems in this dissertation are causal, which means all subsequent states depend only on the current and previous inputs.

The state-space model confers great flexibility. The states can be as ethereal as biological influences or as concrete as mRNA concentrations. The level of mechanistic details depends on data and research objective, while the model imposes no arbitrary restraints. Nonlinear ODEs popular in the bottom-up approach are often state-space models, and in theory, a state-space model originally conceived in a top-down approach effort can be expanded and augmented by mechanistic details to become a model useful in a bottom-up effort.

2. Boolean Networks

Boolean networks can be seen as a kind of state-space model or as a dynamical Bayesian network. Each node can only have 0's and 1's as valid states; and state transition, given states at time t determine the states at time $t + 1$, only occur in discrete time intervals using a Boolean function; thus, it cannot be used to model transient behaviors [28, 24]. Assuming the Boolean networks are of a finite size, there should be finite number of possible states. Given enough time steps repetition of states are bound to occur. Those states that repeat in sequence, which includes the steady states such as a sequence of one, are called attractors. States that are not part of the attractors are transient states, and the attractors along with the transient states that lead to the attractors are called the basin of attraction.

Learning Boolean networks also requires a lot of data, 2^N if there are N genes in the network and assuming full connectivity. Sparsity can reduce the data requirement but the requirement remains considerable and this results in large increase in computational complexity because all plausible sparse networks need to be generated in order to pick the optimal network. An equally taxing problem is discretization. Given the noisiness of data it is very hard to come up with a good scheme that balances false positive and false negative rates. This obstacle appears to be poorly addressed by the literature [24].

An interesting application of Boolean networks is discovering global properties of large scale systems. Kauffman [29] randomly generated graphs of n nodes, each of which is connected to k other nodes which are inputs. For low k and certain kind of state transition functions, the number of attractors on average is $\sqrt{n}$. Kauffman argued that this agreed with observation that the number of cell types is roughly square root of the number of genes, if attractors are interpreted as cell types.

Boolean networks make strong assumptions in order to be able to analyze large systems. The assumptions that states are Boolean variables and state transition occur synchronized are not always true in biological systems. In fact, rarely do biological systems change states in lock step [14].

3. Bayesian Networks

Bayesian networks (BN) and its relative dynamical Bayesian networks (DBN) combine the intuition of graphs with the rigor of probability theory. BN cannot cope with the feedback ubiquitous in biology, and that is remedied by DBN. There are many instances of using DBN to estimate a biological systems, and they will be discussed in the next subsection, where systems identification is treated in more details.

4. Ordinary Differential Equations

ODE is detailed, has a good biochemical foundation (for an introduction in its use in metabolic processes see [30, 31, 6]), and nonlinear ODEs in particular are prevalent in the bottom-up modeling and have proven very successful. Bifurcation analysis can uncover the sensitivity of steady states and limit cycles to variations in parameters. Borisuk and Tyson [7] used bifurcation analysis along with numerical simulation to study the control of mitosis in the Xenopus oocyte, in which a rich variety of physiological states could be discerned, including some that were previously unknown. Also, in a series of remarkable papers [32, 33], researchers found that a purely negative feedback loop could only have one steady state, but a positive feedback loop tends to have two stable states and the initial condition determines in which state the systems end up. Thomas [34, 35] conjectured that a necessary condition for stable periodicity is a negative feedback loop and multistationarity a positive feedback loop This was later proved [36, 37]. The stable periodicity of negative feedback loops can be interpreted as homeostasis, and multistationarity as differentiation [38].

Because kinetic parameters are hard to come by (see [39, 40] for efforts in this regard), and because for the most part there is no analytic solution to the kind of ODE's that arise from kinetics (for numerical solutions see [41, 42], and see [11, 7, 43] for those with bifurcation analysis), nonlinear ODEs have not seen widespread deployment in top-down modeling effort. There is one instance in kinematic modeling presented in subsection 2 of this section. In general current methods are very much inadequate and much hard work remains to be done. See [13] for some discussion on this matter.

5. Partial Differential Equations

In eukaryotic cells genes and proteins are not diffused everywhere but are compartmentalized in organelle and nucleus, so ODEs are not always appropriate. Introducing time delays into ODEs approximate diffusion and transport and can thus alleviate the problems a little. However, for multiple-cells modeling or embryogenesis where concentration gradients are vital, time delayed ODEs are still inappropriate, partial differential equations (PDEs) are needed instead.

Partial differential equations (PDEs) can be written in general as:

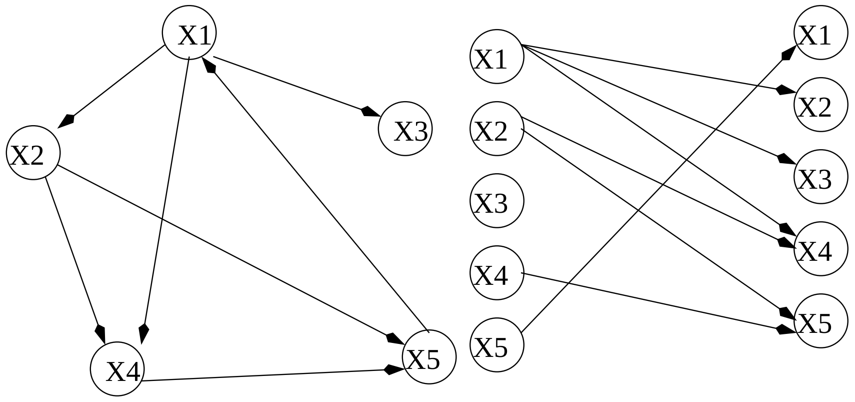
$$\frac{\partial x_i}{\partial t} = f_i(\vec{x}) + \delta_i \frac{\partial^2 x_i}{\partial l^2}$$

where x_i is the expression level of gene i, $\vec{x}$ a vector of all relevant concentrations, δ_i a parameter, l a spatial measurement, and f_i is a function of $\vec{x}$. Add boundary conditions to the above and it becomes a complete specification of PDEs. In general, there is no analytic solution to PDEs, and numerical simulation is tricky, more difficult than ODEs. In special cases, there are certain properties which can be obtained without an explicit solution, and analysis was done for example on the segmentation patterns of Drosophila embryos [44]. The result of that analysis could reproduce the segmentation patterns but only for chosen parameters and boundary conditions, contrary to biological observations that wide variation in individuals nevertheless produce the same patterns. In addition, scant experimental evidence has turned up to support a prediction of the analysis: two concentration variables suffice to explain the segmentation patterns. Those facts have cast doubts on the validity of the analysis.

PDEs are more theoretical than practical in biological modeling. It is reasonable to start with ODEs and go to PDEs later when the results are unsatisfactory with ODEs and it is known that localization plays an important role. The same is true of the chemical master equations (CME) which is stochastic and therefore even more difficult than PDEs.

D. Identification Methods for Genetic Networks

Much work in modeling genetic regulatory networks has gone toward identifying the networks, their structures, and their parameters. Parameter estimation is called sys-



tems identification in engineering and estimation in statistics, while structure learning can be classified as model selection. Boolean networks [45, 46, 47, 48] and variants such as probabilistic Boolean networks [49, 49, 50] represent the simplest dynamical models. Nonlinear differential or difference equations [51, 52] have also seen some interest. Dynamical Bayesian networks (DBN) are quite flexible and state-space models can be seen as one kind of DBN, as can Boolean networks or hidden Markov models, and a body of literature on learning DBN predates its application in biological modeling. So it is no surprise that much interest exists in learning a DBN model of genetic networks [53, 54, 55, 56].

In the subsequent subsections, some examples of learning genetic networks models are provided to illustrate existing approaching.

1. Dynamical Bayesian Networks (DBN)

Bayesian network (BN) is a familiar fixture in the machine learning community; it is a marriage of graphical model with probability theory. A recent development is the dynamical Bayesian network (DBN), which models time-series and can cope with cycles in the network. An example of BN is shown below (left).

Notice that it has cycles, which means it cannot be modeled by regular Bayesian networks. But the same influence model can be modeled by the graph at the right.

Biological networks are full of cycles because feedback is ubiquitous in biology. A rough sketch of the general learning procedure will be given before two examples of applying DBN to learn genetic regulatory networks are given. The two examples both use DBN but they diverge so much in details that they are instructive on how flexible dynamical Bayesian networks can be.

Every node in a Bayesian network, dynamical or not, is a random variable (RV) and edges represent influence. A RV is considered conditionally independent of all of its nondescentants given its parents. For expression profile study, the RVs, represented by lower case x, are gene expressions, and upper case X is a vector of these RVs. In this subsection we will be concerned with expression profiles. The value of a node or its random variable is called a state. If there are n time points and vector X_i represents all states at time i, then the conditional independence property can be written mathematically as:

$$P(X_1, \ldots, X_N) = P(X_1)P(X_2|X_1)\ldots P(X_n|X_{n-1}). \tag{2.1}$$

Assuming the network structure remains stable throughout all time points, we can write for each time point

$$P(X_i|X_{i-1}) = P(x_{i1}|\mathbf{p}_{i-1,1})\ldots P(x_{ip}|\mathbf{p}_{i-1,p}) \tag{2.2}$$

where lower case $\mathbf{p}_{i-1,j}$ is the parents of jth gene at time $i-1$, and x_{ip} the expression level of pth gene at time i.

After this, one needs to specify, for continuous cases, density function for the conditional distribution of states given their parents and the criterion for judging the goodness of a particular model. For inference on parameters, maximum posterior probability is sought with a prior distribution on parameters for Bayesian, or maximum likelihood for frequentists, and that serves as the inner loop for the learning structure, which in theory traverses all possible models to find the best one. In practice this is prohibitively expensive computationally, and heuristics like hill climbing are used to find the best structure. In biological applications biological restrictions are placed on candidate models to further cut down on the running time.

2. Kinematic Modeling

Nachman et al. [57] decided to get down to the biochemical level and give a kinetic model. They used the nonlinear Michaelis-Menten form of reaction speed

$$g(H, \beta) = \beta \frac{\gamma H}{1 + \gamma H}$$

where H is the concentration of regulator protein, β the maximum transcription rate of the gene, and γ the ratio of association and disassociation constants. This formula have some presuppositions, among which are: the system is at equilibrium, the association rate is much greater than the transcription rate, and the concentration of regulator protein is much less than that of the genes or the polymerase.

For two regulators and two binding sites, they defined the following quantities for fours cases of regulators occupying the binding sites:

$$\begin{array}{rlrl} S^{-.-} &= \dfrac{1}{Z} \text{ no binding} & S^{-.H_2} &= \gamma_2 H_2 / Z \text{ binding on 2nd site} \\ S^{H_1.-} &= \gamma_1 H_1 / Z \text{ binding on 1st site} & S^{H_1.H_2} &= \gamma_1 H_1 \gamma_2 H_2 / Z \text{ both binding} \end{array}$$

where $Z = (1 + \gamma_1 H_1)(1 + \gamma_2 H_2)$ is a normalizing constant, H is protein concentration level, and the regulator function can be defined as weighted sum over all possible binding states:

$$g(H_1, H_2 : \vec{\alpha}, \beta, \gamma_1, \gamma_2) = \beta(\alpha^{-.-} S^{-.-} + \alpha^{-.H_2} S^{-.H_2} + \alpha^{H_1.-} S^{H_1.-} + \alpha_{H_1.H_2} S^{H_1.H_2}) \tag{2.3}$$

where $\vec{\alpha}$ is the vector that has all the various cases of α and each α is an indicator variable that can take on only values 0 or 1 depending on the presence or absence, respectively, of the binding pattern. In [57] only binary values are assigned to α's, so for non-cooperative activators $\alpha^{-\cdot-}$ is 0 and all other α's are 1. This method can be easily extended to more than two regulators, so they will not be elaborated here.

Nachman et al. modeled the regulators at time $t+1$, H_i^{t+1}, from time t, H_i^t, with the persistence equation:

$$H_i^{t+1} = H_i^t + \epsilon_{hi}^{t+1}$$

where ϵ_{hi}^{t+1} is normally distributed noise with zero mean and variance σ_i. The conditional distribution for genes is different. For example, for genes regulated by two regulators the conditional distribution of transcription rate R_k^t given regulator activities of H_1 and H_2 is

$$R_k^t = g(H_1^t, H_2^t : \vec{\alpha}_k, \beta_k, \gamma_{k,1}, \gamma_{k,2})(1 + \epsilon_{rk}^t)$$

where ϵ_{rk}^t is a Gaussian noise variable with zero mean and variance σ_k and $g(\cdot)$ is as defined by equation (2.3).

A common problem with modeling at such a low level is that reaction rates are modeled but microarrays provide only expression levels, so Nachman et al. decided to model the mRNA changes as well, with $r_k^t - \delta_k e_k^t = \frac{d}{dt} e_k^t$, where r_k^t is the transcription rate, e_k^t is the expression level of gene k at time t, and δ_k is the decay constant of gene k. By solving this differential equation transcription rate r_k^t is obtained. It should be noted that estimating rates from expression levels is extremely unreliable in general.

The likelihood function can be written from the above equations, but to compensate for the tendency of maximum likelihood overfitting, a penalty term consisting of $\frac{N_{\text{param}}}{2} \log(T)$ where N_{param} is the number of parameters and T is the number of time points, is subtracted. This is called Bayesian information criterion (BIC) score. Another salient point is that Nachman et al. did not use hill climbing but something called structural EM by Friedman et al. [58] for learning structures.

3. Nonparametric Regression for Nonlinear Modeling

Kim et al. [54] wanted to model nonlinear dynamics of genetic regulation and decided to use nonparametric additive regression model with Gaussian noise as the conditional densities. The density function at time i given its previous states is

$$f_i(\mathbf{x}_i|\mathbf{x}_{i-1}) = g_i(x_{i1}|\mathbf{p}_{i-1,1})\dots g_p(x_{ip}|\mathbf{p}_{i-1,p})$$

(note that boldface $\mathbf{x}$ represents a vector of variables and x_{ip} their components) for p genes, and the full dynamical Bayesian network model can be described by the following density functions:

$$f(x_{11},\dots,x_{np}) = f_1(\mathbf{x}_1)\prod_{j=1}^{p}\left\{\prod_{i=2}^{n} g_j(x_{ij}|\mathbf{p}_{i-1,j})\right\},$$

where

$$g_j(x_{ij}|\mathbf{p}_{i-1,j}) = \frac{1}{\sqrt{2\pi\sigma_j^2}}e^{\left\{-\frac{(x_{ij}-\mu(\mathbf{p}_{i-1,j}))^2}{2\sigma^2}\right\}},$$

and $\mu(\mathbf{p}_{i-1,j})$ (a vector function) is the mean and a linear combination of basis functions of jth gene's parents' expression levels at time $i-1$. The basis functions in this case are B-splines.

The posterior probability of the network is proportional to

$$\pi(G)\int f(x_{11},\dots,x_{np})\pi(\theta_G|\lambda)d\,\theta_G,$$

where $\pi(G)$ is the prior probability of network G, and $\pi(\theta_G|\lambda)$ a prior distribution on parameters θ_G and λ a vector of hyperparameters. But this involves an integral which is hard to compute so Kim et al. decided to use the Laplace approximation for integrals to obtain the following criterion:

$$-2\log\pi(G) - r\log(\frac{2\pi}{n}) + \log|J_\lambda(\hat{\theta}_G)| - 2nl_\lambda(\hat{\theta}_G|\mathbf{X})$$

where r is the dimension of θ_G,

$$\begin{aligned} l_\lambda(\theta_G|\mathbf{X}) &= \frac{\log f(x_{11},\ldots,x_{np};\theta)}{n} + \frac{\log\pi(\theta_G|\lambda)}{n}, \\ J_\lambda(\theta_G) &= -\frac{\partial^2\{l_\lambda(\theta_G|\mathbf{X})\}}{\partial\theta_G\partial\theta_G^T}. \end{aligned}$$

$\hat{\theta}_G$ is the mode of $l_\lambda(\theta_G|\mathbf{X})$, and $J_\lambda(\theta_G)$ is a Hessian matrix. The prior on parameters will not be introduced here. Suffice it to say, it involves a lot of symbols in what is basically a Gaussian distribution. It will be noted that parameters are assumed to be independent.

4. Boolean Networks

Boolean networks consist of directed graphs where nodes take on binary values of 0 and 1, and Boolean functions that detail state transition given parents' states. In theory, Boolean networks are also Bayesian networks, but the difference is pronounced enough that the learning algorithms are completely different. A simple extension of Boolean network is probabilistic Boolean network (PBN), in which each node has a collection of possible Boolean functions and the choice of which function is used is probabilistic. But the example I chose to present below is deterministic, not only because it is from one of the most cited papers, but also that PBN learning builds on learning deterministic Boolean networks.

Boolean networks are learned using data from the steady-state expression levels of genes after some perturbations. The perturbation can be gene knockout, gene over-expression, or physical or biochemical stress such as radiation, temperature changes or drug treatment. Learning a Boolean network takes a lot of data. A network of size N if fully connected requires 2^N samples, an impossible requirement in practice, so people try to take advantage of the sparse nature of genetic networks to pare back those hefty requirements. Ideker et al. [45] has the following algorithm to learn a Boolean network and, if data is not enough to uniquely determine a network, new experiments to get more is suggested.

Suppose the following is a collection of data so far. There are four genes and 5 perturbation samples. The first row denote gene names and the rest are the data, one row per sample, starting with sample 0 to sample 4. The symbols _ and + denote the levels artificially set by the experimenter, of low and high, respectively, while 0 and 1 also represent low and high levels but they are from measurements.

$$\left[\begin{array}{ccccl} x_0 & x_1 & x_2 & x_3 & \\ 1 & 1 & 1 & 0 & \text{sample 0} \\ \_ & 1 & 0 & 1 & \text{sample 1} \\ 1 & \_ & 0 & 0 & \text{sample 2} \\ 1 & 1 & \_ & 1 & \text{sample 3} \\ 1 & 1 & 1 & + & \text{sample 4} \end{array}\right]$$

The algorithm consists of three steps for each gene:

1. For gene x_n, find all pairs of samples where x_n's values differ, except where x_n is artificially set. For instance, x_3 in the example has sample pairs $(0, 1)$, $(0, 3)$, $(1, 2)$, $(2, 3)$ with different x_3 values. Sample 4 is excluded since x_3 is artificially set. Then, from each pair, discover all genes whose expressions differ between the two samples except the original gene that generated the pair. Sample pair $(0, 1)$, denoted by S_{01} has (x_2, x_0) different, and $S_{12} = (x_0, x_1)$, $S_{23} = (x_1)$; but

no x_3 because different x_3 values generated these pairs..

2. Find a minium set, $S_{\min}$, of genes such that each S_{ij} collected from step 1 has at least one gene in $S_{\min}$. This is an NP-complete problem and Ideker et al. suggested a branch and bound algorithm, which leads to long running time for large-sized networks. One good thing about the branch and bound algorithm however is that it can discover all the minimum $S_{\min}$ networks, but since all of them must be kept around, and since this is done for every gene the number of candidate networks can grow quickly. $S_{\min} = (x_1, x_2)$ for the example.

3. For gene x_n, a truth table is constructed by taking expression levels of $S_{\min}$'s members directly from data samples. In the example we have used so far, x_3 has $(1,1)$, $(1,0)$, and $(0,1)$, from which we can construct a partial truth table.

This produces the most parsimonious network, and given a list of suggested perturbation experiments one could use information theory to find out which perturbation prunes the number of candidate networks the most. It goes like this: given a perturbation, there are totally S distinct states for L candidate networks, and for each state set s there are l_s networks, so we compute

$$H_p = -\sum_{s=1}^{S} \frac{l_s}{L} \log_2 \left(\frac{l_s}{L} \right)$$

for each perturbation and choose the one that generates maximum H_p.

CHAPTER III

CONSTRAINED SYSTEMS IDENTIFICATION

A. Introduction

Genetic regulatory networks, often abbreviated as genetic net-works, help us untangle the intricate interactions of multiple genes under different environmental conditions. Recent developments in microarray technologies allow scientists to simultaneously measure expressions of thousands, even tens of thousands of genes, over time. The time course of gene expression data can be used to reconstruct genetic networks. In the past decade, a wide variety of models have been developed to study genetic networks. They include Boolean networks [59, 60], differential equations [27, 61] and dynamic Bayesian networks [62, 63, 64, 65].

A special subclass of dynamic Bayesian networks (DBN) is linear dynamical systems (LDS) [66, 62, 67, 68], also known as linear state-space models. LDS assumes that observations depend on unobservable states that evolve under Markovian dynamics, i.e., future states are probabilistically determined only by the current state. The state-space approach can provide a general framework for the design of genetic networks in synthetic biology; how-ever, to date, most efforts in estimating an LDS have tried to estimate parameters without considering structural constraints. Wu et al. [69] proposed to use LDS to explore large time-course data. While they used hidden, unobserved variables as the states and modeled expression profiles as the output, they did not consider noise, and they estimated states from the output using maximum likelihood factor analysis. The number of states in estimated models was a variable estimated using the Bayesian information criterion (BIC), and the state transition matrix was estimated using least square methods. Wu et al. [69] considered the state transition matrix as the key parameter that embodies genetic interaction, and the stability or instability of this matrix was cited as supporting evidence for their method. One limitation of Wu's approach is that factors have no obvious biological counterpart.

Later, Yamaguchi and colleagues [70, 71] also tried to use similar linear state-space system to model genetic networks, where a state was defined as a module of interacting genes. The parameters in the model measured quantitative relationships between modules. The dimension of the states was determined by BIC and the parameters were estimated by the EM algorithm.

Rangel and her colleagues, on the other hand, in a series of papers [72, 73, 74], modeled individual gene interactions using linear state-space models. Their estimation method consisted of two parts. The inner part was the expectation-maximization (EM) algorithm with full connectivity assumed and therefore no structural constraints. To avoid over-modeling, they first estimated the dimension of hidden states using cross-validation. To avoid over-fitting, they augmented their objective function with terms that favor sparsity. In the outer loop, a bootstrapping method was used to estimate the confidence intervals for all the parameters. Presence or absence of connection between the genes in the network depends on whether the confidence inter-val includes zero or not, which added up to be the structure of the network. The final result of their method was a connectivity matrix, but no dynamical model resulted since none of the estimated models agreed with the inferred connectivity matrix. Our work seeks to estimate parameters that agree completely with a given connectivity matrix.

Another approach is to separate the task of parameter estimation and model selection and perform them separately, as Gennemark and Wedelin [10] did for S-system models of genetic networks, where parameter estimation was done under the constraint of currently estimated structure. This is useful also when connectivity is available from the literature [75, 76, 77] in which case only parameters need to be estimated. Rangel et al. [72] recognized imposing constraints on parameters as a possible extension of their work and suggested it in the discussion, and we will be following that lead precisely in this report.

The purpose of this report is to develop state-space representations of genetic networks with known structures. The method we propose is to combine linear dynamical modeling with structural constraints to produce biologically realistic models that can predict the dynamic behaviors of genetic networks. The motivation for imposing structural constraints is as follows: Any genetic network has a structure specifying interactions between the genes represented as connections. Not all genes are connected, and in fact, every evidence points to genetic networks being sparse. The network structures can often be determined from experiments by biologists, or they can be roughly inferred by model selection. In linear dynamical models, the structure of the genetic network is mainly reflected in the elements of the system matrices. If there are no connections between the genes or connections between the gene and the external stimuli, their corresponding elements in the system matrices should be equal to zero. Without such constraints the models cannot take the structure of the network into account. Therefore, to ensure that estimated models agree with a known structure, constraints must be imposed on parameters. A popular method for the estimation of parameters in state-space models is the expectation-maximization (EM) algorithm. However, conventional unconstrained EM algorithms cannot be applied to models with constraints, so modifications must be made. Incorporating network structure into the state-space model of genetic networks will lead to imposing constraints on the parameter space. Although constrained EM algorithms have been proposed in the engineering literature [78, 79], the proposed constrained EM algorithms require iterative numerical solutions to equations derived in the M-step of the EM algorithm. This iterative solution inside an already iterative method would make computation time intolerably long. While generalized EM algo-rithms like the one Wu et al. [80] used avoid iterative nu-merical solutions, they have slower convergence speed. Therefore, in this report, we present a new type of constrained EM algorithm that admits analytical and decoupled solution and thus preserves EM's speed while not resorting to numerical solutions or generalized EM. In the DBN community, this problem of structural constraint is called the known structure and partial observability for the learning of Bayesian networks [81]. Since structures of some genetic networks can be gleaned from the literature or discovered through ChIP-on-chip experiments [82], they should be taken into account whenever available [83]. Application to synthetic data and real world SOS data show that our method significantly outperforms conventional EM and that structural constraints are important in the reverse engineering of genetic networks.

B. Methods

1. Linear Dynamical Systems

We have adopted the linear state-space model as the underlying model for genetic networks, in particular the linear time-invariant (LTI) model. LTI is a linear state-space model where parameters do not change over time [84]. A linear state-space model of a dynamical system can be written as

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + w \\ y_t &= Cx_t + Du_t + v \end{aligned} \tag{3.1}$$

where x_t is the state vector, y_t the output vector, u_t the input vector, all at time t ; w and v are independent noise terms assumed to be white Gaussian with zero mean and covariance Q and R respectively; matrix A is called the state transition matrix, B the input matrix, C the output matrix, and D the feed-forward matrix. Matrices A, B, C, D and covariance matrices Q and R together make up the parameters of the dynamical system. Of all the matrices, A is the most important, as its eigenvalues determine the stability of LTI. The system is stable if all eigenvalues are inside the unit circle in the complex plane and unstable otherwise. The states represent the biological forces that regulate gene expression. They describe the behaviors of gene transcription but are hidden. The outputs denote the gene expression levels and are measured by microarrays or green fluorescent proteins (GFPs): The expression level is determined by the states of the regulated gene. The inputs can be any external stimuli that influence gene regulation, such as drugs, proteins, RNAs, or expression levels of connected genes.

The linear state-space model represented in equation (3.1) is quite general and can represent more than simple exponential growth and decay, for it can represent higher order dynamics, which we will look at next.

2. Higer Order Dynamics

If we stick with one gene for one state, then the system represented in equation (3.1) only will have first order dynamics associated with all the genes, which is exponential decay or growth, but since oscillation is widely observed in biology at least second order should be considered in models of genetic networks. We will give a simple derivation of how to add second order dynamics for the individual nodes of the networks using the principle of continuous to discrete conversion. This is similar to d'Alch-Buc's method [85]. Of course third or higher order dynamics can be similarly modeled, but care must be taken to avoid over-fitting. Suppose we have a second order linear differential equation describing the dynamics of a node:

$$\ddot{x} + \lambda_1 \dot{x} + \lambda_2 x = \sum_j w_j z_j,$$

where x is the state of the node we are interested in,z_j is the expression level of node j and w_j its corresponding weight, and λ_1 and $\lambda 2$ parameters. Let

$$x_1 = x,\ x_2 = \dot{x}.$$

Then we get

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} x_2 \\ \sum\limits_j w_j z_j - \lambda_1 x_2 - \lambda_2 x_1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ -\lambda_2 & -\lambda_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & \cdots \\ w_1 & \cdots \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \end{bmatrix}. \end{aligned} \tag{3.2}$$

If the steps are uniform, i.e.$\delta t = 1$, then we can represent the derivatives as

$$\frac{dx}{dt} \approx \frac{\Delta x}{\Delta t}, \text{ which becomes } \Delta x = x(k+1) - x(k),$$

where k is the time step and the equation (3.2) becomes

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} &= \begin{bmatrix} x_1(k) + x_2(k) \\ \sum\limits_j w_j z_j(k) - (\lambda_1 - 1) x_2(k) - \lambda_2 x_1(k) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 \\ -\lambda_2 & 1 - \lambda_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 & \cdots \\ w_1 & \cdots \end{bmatrix} \begin{bmatrix} z_1(k) \\ \vdots \end{bmatrix}. \end{aligned} \tag{3.3}$$

The ones and zeros in equation (3.3) are fixed except in $1 - \lambda_1$ where the whole term is variable. An interesting observation is that all interactions and inputs are in the second order term x_2.

We will apply this conversion to just one gene in the SOS network, lexA. But first we need to derive the constrained EM algorithm.

3. Expectation-Maximization

Expectation-maximization (EM) is a well known method in sys-tems identification [86, 87, 88, 89]. EM is a Maximum-Likelihood (ML) estimator of unobserved states and unknown parameters and it operates in an iterative fashion. Each iteration consists of two steps: the E-step and the M-step. In the E-step, states are estimated using Kalman smoother with previous estimates of parameters as model parameters. In the M-step, parameters are estimated using the estimated states obtained in the E-step, and parameters are calculated as to maximize the likelihood. We will focus on our modification to the M-step where network structure constraints are taken into account. We will follow the notations in Gibson and Ninness [88], while Kailath et al. [90] is a good source on Kalman filter. Rewriting equation (3.1) as

$$\begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} w \\ v \end{bmatrix},$$

while adopting the following definition for the sake of convenience:

$$z_t = \begin{bmatrix} x_t \\ u_t \end{bmatrix} \quad \xi_t = \begin{bmatrix} x_{t+1} \\ y_t \end{bmatrix}, \ \Gamma = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad \Pi = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}. \tag{3.4}$$

So equation (3.4) becomes

$$\xi_t = \Gamma z_t + \begin{bmatrix} w \\ v \end{bmatrix}, \ \begin{bmatrix} w \\ v \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Pi \right).$$

We also shall denote all the observations (or outputs) as$\mathbf{Y}$, all the inputs as $\mathbf{U}$, and all the states as $\mathbf{X}$.

4. E-step

E-step needs to figure out the conditional expectation

$$Q(\theta,\theta') = E_{\theta'}[\log P_\theta(\mathbf{X},\mathbf{Y}|\mathbf{U})|\mathbf{Y},\mathbf{U}]$$

where θ is a vector of model parameters, θ' is the current estimate of the parameters and all the outputs are represented as $\mathbf{Y}$, all the inputs as $\mathbf{U}$, and all of the states as $\mathbf{X}$. First, the likelihood function for one time series is

$$P_\theta(\mathbf{Y}_{\tau_n},\mathbf{X}_{\tau_n+1}|\mathbf{U}_{\tau_n}) = P_\theta(x_1)\prod_{t=1}^{\tau_n} P_\theta(x_{t+1},y_t|x_t,u_t), \tag{3.5}$$

where $\tau_n$is the number of time points of the time series, $\mathbf{Y}_{\tau_n}$ and $\mathbf{U}_{\tau_n}$ are all the observations and inputs for a particular time series, distribution $P_\theta(x_1)$ is $\mathcal{N}(\mu,\mathcal{P}_1)$ and distribution $P_\theta\left(\left[\begin{array}{c} x_{t+1} \\ y_t \end{array}\right]\middle| x_t, u_t\right)$ is $\mathcal{N}(\Gamma z_t,\Pi)$; the equation is obtained through noises being uncorrelated and Gaussian. Expand equation (3.5) and take logarithm to get

$$\begin{aligned} -2\log\mathrm{P}_\theta(\mathbf{Y}_{\tau_n},\mathbf{X}_{\tau_n+1}|\mathbf{U}_{\tau_n}) &= \log|\mathcal{P}_1| + (x_1-\mu)^T\mathcal{P}_1^{-1}(x_1-\mu) \\ &+\tau_n\log|\Pi| + \sum_{t=1}^{\tau_n}(\xi_t-\Gamma z_t)^T\Pi^{-1}(\xi_t-\Gamma z_t) \quad . \end{aligned} \tag{3.6}$$

Define the following notations for N time series.

$$\begin{aligned}
&\Upsilon = \sum_{n=1}^{N} \tau_n,\ \Phi = \frac{1}{\Upsilon}\sum_{n=1}^{N}\sum_{t=1}^{\tau_n} E_{\theta'}\{\xi_t^n\,(\xi_t^n)^T\,|\mathbf{Y}_{\tau_n}^n, \mathbf{U}_{\tau_n}^n\},\\
&\Psi = \frac{1}{\Upsilon}\sum_{n=1}^{N}\sum_{t=1}^{\tau_n} E_{\theta'}\{\xi_t^n\,(z_t^n)^T\,|\mathbf{Y}_{\tau_n}^n, \mathbf{U}_{\tau_n}^n\},\\
&\Sigma = \frac{1}{\Upsilon}\sum_{n=1}^{N}\sum_{t=1}^{\tau_n} E_{\theta'}\{z_t^n\,(z_t^n)^T\,|\mathbf{Y}_{\tau_n}^n, \mathbf{U}_{\tau_n}^n\},
\end{aligned} \tag{3.7}$$

where τ_n is the number of time-points in time series n, assuming there are N time series and the superscript denotes the nth time series. EM needs the expectation of the log-likelihood function, so we take expectation of equation (3.6) to obtain

$$\begin{aligned}
-2Q(\theta,\theta') = \log|\mathcal{P}_1| + \mathrm{trace}\{\mathcal{P}_1^{-1}E_{\theta'}\{(x_1-\mu)^T(x_1-\mu)\} + \Upsilon\log|\Pi|\\
+\Upsilon\mathrm{trace}\left\{\Pi^{-1}\left[\Phi - \Psi\Gamma^T - \Gamma\Psi^T + \Gamma\Sigma\Gamma^T\right]\right\}\quad .
\end{aligned} \tag{3.8}$$

Equation (3.8) are made up of terms defined in equation (3.7), which in turn can be found from these expectations:

$$\begin{aligned}
&\mathrm{E}_{\theta'}\{y_t x_t^T|\mathbf{Y}_{\tau_n}, \mathbf{U}_{\tau_n}\} = y_t\hat{x}_{t|\tau_n}^T\\
&\mathrm{E}_{\theta'}\{x_t x_t^T|\mathbf{Y}_{\tau_n}, \mathbf{U}_{\tau_n}\} = \hat{x}_{t|\tau_n}\hat{x}_{t|\tau_n}^T + P_{t|\tau_n}\\
&\mathrm{E}_{\theta'}\{x_t x_{t-1}^T|\mathbf{Y}_{\tau_n}, \mathbf{U}_{\tau_n}\} = \hat{x}_{t|\tau_n}\hat{x}_{t-1|\tau_n}^T + M_{t|\tau_n},
\end{aligned}$$

$$\begin{aligned}
&\text{where } \hat{x}_{t|\tau_n} = \mathrm{E}\left[x_t|\mathbf{Y}_{\tau_n}, \mathbf{U}_{\tau_n}\right],\ P_{t|\tau_n} = \mathrm{var}\left[x_t|\mathbf{Y}_{\tau_n}, \mathbf{U}_{\tau_n}\right],\\
&M_{t|\tau_n} = \mathrm{cov}\left[x_t, x_{t-1}|\mathbf{Y}_{\tau_n}, \mathbf{U}_{\tau_n}\right],
\end{aligned}$$

and τ_n is the number of time-points in the nth time series, $\mathbf{Y}_{\tau_n}$ and $\mathbf{U}_{\tau_n}$ are the observations and inputs for the time series. They can be otained from Kalman smoother

[91]

$$
\begin{aligned}
&J_t = P_{t|t}A^T P_{t+1|t}^{-1} \\
&\hat{x}_{t|\tau_n} = \hat{x}_{t|t} + J_t[\hat{x}_{t+1|\tau_n} - A\hat{x}_{t|t} - Bu_t - R^{-1}y_t] \\
&P_{t|\tau_n} = P_{t|t} + J_t[P_{t+1|\tau_n} - P_{t+1|t}]J_t^T \\
&M_{t|\tau_n} = P_{t|t}J_{t-1}^T + J_t[M_{t+1|\tau_n} - AP_{t|t}]J_{t-1}^T
\end{aligned}
$$

where $\hat{x}_{t|t}, P_{t|t}, P_{t|t-1}$ are calculated from Kalman filter

$$
\begin{aligned}
&P_{t|t-1} = AP_{t-1|t-1}A^T + Q \\
&G_t = P_{t|t-1}C^T(CP_{t|t-1}C^T + R)^{-1} \\
&P_{t|t} = P_{t|t-1} - G_tCP_{t|t-1} \\
&\hat{x}_{t|t-1} = A\hat{x}_{t-1|t-1} + Bu_{t-1} \\
&\hat{x}_{t|t} = \hat{x}_{t|t-1} + G_t(y_t - C\hat{x}_{t|t-1} - Du_t) \\
&t = 1, ..., \tau_n, \\
&M_{\tau_n|\tau_n} = (I - G_{\tau_n}C)AP_{\tau_n-1|\tau_n-1}.
\end{aligned}
$$

The Kalman filter, which is essentially recursive least-square and is an optimal linear estimator, progresses forward in time, and the Kalman smoother, which mathematically can be proved by conditioning normal distributions, goes backward in time, hence they are also called forward-backward algorithm.

5. M-step

Reuse the notation defined in equation (3.7), for all N time series needed for the M-step. Here the superscript denotes the nth time series.

The conventional EM would have the M-step as

$$\mu = \hat{x}_{1|\tau_n},\ \mathcal{P}_1 = P_{1|\tau_n},\ \Gamma_{\text{new}} = \Psi\Sigma^{-1},\ \Pi_{\text{new}} = \Phi - \Psi\Sigma^{-1}\Psi^T, \tag{3.9}$$

where μ is the estimated mean of x_1 and $\mathcal{P}_1$ the estimated variance. We note here that the last two equations in (3.9) are de-coupled and everything on the right hand side can be computed from the E-step results; thus the M-step in conventional EM has an analytic solution and is very fast. This feature of con-ventional EM is one reason for its popularity, but it may be lost if constraints are imposed on parameters because structural constraints force some parameters to be zeros while leaving others free to change. In that case, equation (3.9) is no longer valid due to structural constraints, and numerical solution may be required for maximization. Having an iterative solver within an already iterative method will significantly increase computation time. Generalized EM, another solution that can permit constraints on parameters however, it is known to have slower convergence. Since parameter estimation could become the inner loop of a bigger model selection algorithm, we want to strive for decoupled and analytic solutions. Fortunately, with a mild assumption on the type of noise, that it has diagonal variance, we are able to obtain an analytic solution.

6. Structural Systems Identification

Given a network whose structure is known, for example if a pair of genes is represented by two states and they have no interaction, then the corresponding entry in matrix A should be zero. Similarly, if an input has no influence on a gene that is represented as a state, then the corresponding entry in B should be zero. The same goes for entries in C that describe how measurements depend on the states. Matrix D is usually all zeros because genes do not impact other genes' expression level instantaneously.

Take for example the SOS network in Figure 1. The sole input is the gene recA, so input vector u is a scalar, and matrix B is a vector whose entries are all zeros except for the first element. Among the rest of the genes, only lexA interacts with other genes so matrix A only has the first column and the diagonal entries as nonzero

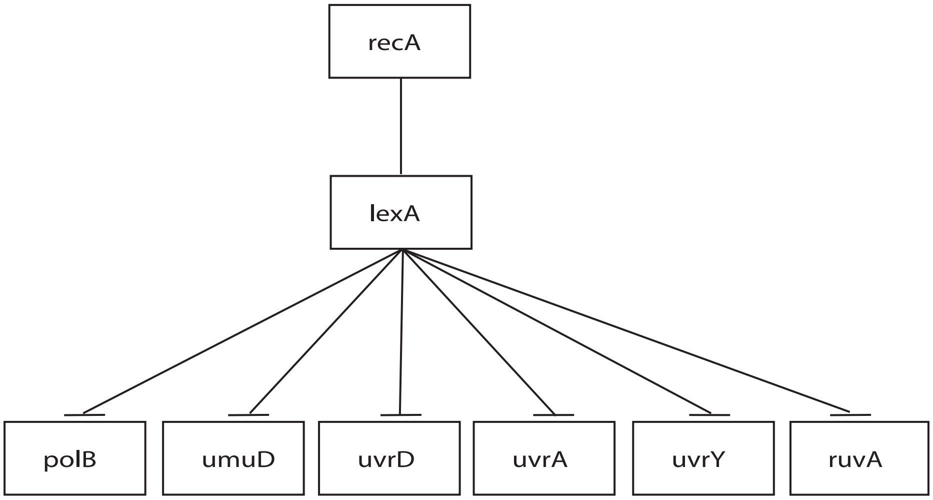


Fig. 1.: This is a diagram of eight essential genes of the SOS DNA repair networks [92].

for a first order system. The matrix diagonal entries are nonzero because, in general, genes impact their own expression levels. Since we measure all the genes' expression levels directly, we set C to be an identity matrix. D is a zero matrix as explained above. So putting all this together, we have parameters A, B, C, and D initially determined as follows:

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 & 0 & 0 \\ a_{31} & 0 & a_{33} & 0 & 0 & 0 & 0 \\ a_{41} & 0 & 0 & a_{44} & 0 & 0 & 0 \\ a_{51} & 0 & 0 & 0 & a_{55} & 0 & 0 \\ a_{61} & 0 & 0 & 0 & 0 & a_{66} & 0 \\ a_{71} & 0 & 0 & 0 & 0 & 0 & a_{77} \end{bmatrix} B = \begin{bmatrix} b_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, D = 0.$$

But in initial estimations, we discovered that first order dynamics does not adequately describe the time series, so we increased the order of the lexA gene to two and that proved successful. Accordingly, the parameters A, B, C, and D are changed to

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & 0 & a_{33} & 0 & 0 & 0 & 0 & 0 \\ a_{41} & 0 & 0 & a_{44} & 0 & 0 & 0 & 0 \\ a_{51} & 0 & 0 & 0 & a_{55} & 0 & 0 & 0 \\ a_{61} & 0 & 0 & 0 & 0 & a_{66} & 0 & 0 \\ a_{71} & 0 & 0 & 0 & 0 & 0 & a_{77} & 0 \\ a_{81} & 0 & 0 & 0 & 0 & 0 & 0 & a_{88} \end{bmatrix} B = \begin{bmatrix} 0 \\ b_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, D = 0.$$

Here we expanded gene lexA to have second order dynamics, resulting in two scalar states, x_1 and x_2, where x_2 is the derivative of x_1 discretized, while x_1 represents lexA's expression level. Matrix B is changed because all interactions are on the second order term, and C is changed to reflect the fact that we do not have measurement for x_2 (column 2 is all zeroes). We also assume Π to be diagonal since we have no reason to believe that noise in each state or measurement is correlated. This also is a standard assumption unless there is specific evidence that contravenes the assumption. Making Π diagonal also makes the analytic form of M-step possible.

Incorporating structural constraints results in an M-step that is more complicated than the M-step in the conventional EM algo-rithms:

$$[\Gamma_{\text{new}}\Sigma - \Psi] \circ M = 0 \tag{3.10}$$

$$\Pi_{\text{new}} = \left\{\Phi - \Psi\Gamma_{\text{new}}^T - \Gamma_{\text{new}}\Psi^T + \Gamma_{\text{new}}\Sigma\Gamma_{\text{new}}^T\right\} \circ I \tag{3.11}$$

where I is an identity matrix of appropriate size and M is a constraint matrix of Γ so that if an entry of Γ is constrained, the corresponding entry of M is 0, and 1 otherwise. The notation $\circ$ represents element-wise product, also known as the Hadamard product. Equations (3.10) and (3.11) are quite general, so the formulas admit nonzero constrained values.

Equation (3.11) looks more complicated, but the calculation is actually explicit as long as we have Γ_{new} from equation (3.10). Equation (3.10) can be solved row-wise by the following procedure:

Explicit solution for Γ in $[\Gamma\Sigma \circ M = 0]$ can be obtained by:

1. for each row of Γ, Γ_j, suppose r_j= indices of con-strained elements of Γ_j ;

2. delete all elements of Γ_j and Ψ_j , the ith row of Ψ , whose indices are in r_j;

3. delete all rows and columns of Σ whose indices are in r_j;

4. solve $[\Psi_j]_{mod} - \Sigma_{mod}[\Gamma_j]_{mod}^T$ where the notation $[]_{mod}$ denotes respective vectors and matrices after deletion in step 2 and 3.

Therefore the procedure above and equation (3.11) plus the first two equations in equation (3.9) constitute the modified M-step.

7. Data Source

First, the synthetic data was generated by a system that had four states and four outputs and the parameters were as follows:

$$\text{A} = \begin{bmatrix} 0.8 & 0 & 0.8 & 0 \\ 0 & 0.8 & 0 & 0.8 \\ 0 & 0 & 0.8 & 0 \\ 0.8 & 0 & 0 & 0.8 \end{bmatrix} \quad B = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D = 0.$$

We generated 200 time points for cross-validation. Since C is fixed in estimation as an identity matrix, no equivalent system exists by similarity transformation, and this system is identifiable.

Second, to validate our method with real-world data, we chose the SOS DNA repair network of the *Escherichia coli* with 8 essential genes. The SOS network is a highly conserved system and is a well studied network [93, 94]. It consists of about 30 genes, the master regulator being the lexA gene. Gene lexA inhibits all the rest of the SOS network's gene under normal condition, and when DNA damage is sensed, the normally suppressed genes become active. A diagram of SOS network with 8 essential genes is shown in Figure 1.

The experimental data for the SOS system can be downloaded from Uri Alon's homepage. Ronen et al. [92] used green fluo-rescent proteins (GFP) to track 8

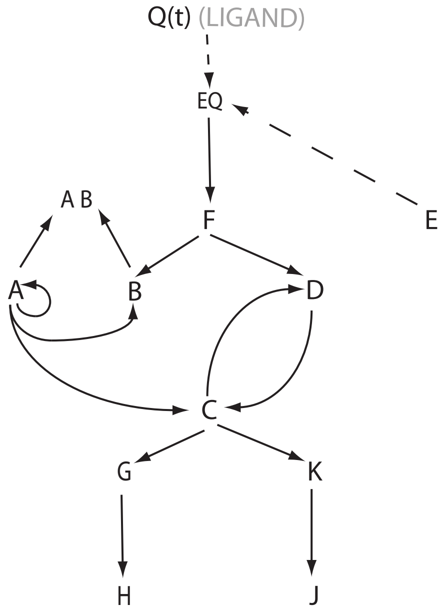


Fig. 2.: This diagram of a simulated system from Zak et al. [83] shows a biologically inspired system driven by ligand binding. This figure illustrates the relative expression levels when the ligand is high.

genes of the SOS network as they react to different irradiation levels, 5Jm^{-1} and 20 Jm^{-1}; each level has two samples and each sample has 50 evenly spaced time points. They monitored eight genes: uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA, and polB. They performed extensive data preprocessing on the raw data using hybrid Gaussian median filter and polynomial fit for smoothing. They also assumed that the rate of accumulation of GFP was proportional to transcript production. We shall make the same assumption. To test our method on more complex models, we decided to use a biologically inspired artificial system by Zak et al. [83]. They used stochastic simulation to simulate gene expressions and protein interactions. The data had 550 time points, but very few time-course data currently available are that long. So we sampled one time point out of every five consecutive time points to obtain a time series of 110 time points. A complete diagram of the artificial system is shown in Figure 2, and the parameters are constrained as follows:

$$A = \begin{bmatrix} a_{11} & 0 & 0 & a_{14} & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & 0 & a_{33} & 0 & a_{35} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{43} & a_{44} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{53} & 0 & a_{55} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{65} & a_{66} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{76} & a_{77} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{85} & 0 & 0 & a_{88} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{98} & a_{99} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & 0 \\ 0 & b_{22} \\ 0 & 0 \\ 0 & 0 \\ 0 & b_{52} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D = 0.$$

C. Results

To examine the consistency of the constrained EM approach and to test its biological applicability, we applied our new method to two sets of synthetic data and the

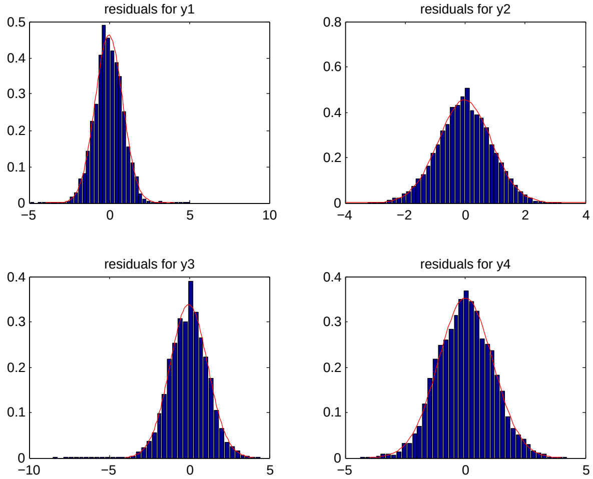


Fig. 3.: These are the histograms of standardized residuals of outputs y_1, y_2, y_3, y_4, for the synthetic data. They resemble standard normal distribution.

SOS DNA repair network data. We first examined the distributions of standardized residuals (or errors) of the Kalman filter for the synthetic data and found that they largely resemble Gaussian distributions. For the synthetic data sets, we compared the predictive power and estimation precision of our constrained EM and the unconstrained EM through prediction errors and confidence intervals. For the SOS data, two replications were not enough for bootstrapping so no confidence interval could be derived.

1. Errors Distribution

To examine the consistency of the constrained EM and to test their biological applicability, we have applied our new method to synthetic data and real SOS DNA repair network data. For the synthetic data, we first examine the distribution of the standardized residuals (or errors) in the Kalman filter results. This is a common way to perform diagnostics on model assumption and the estimation itself. The

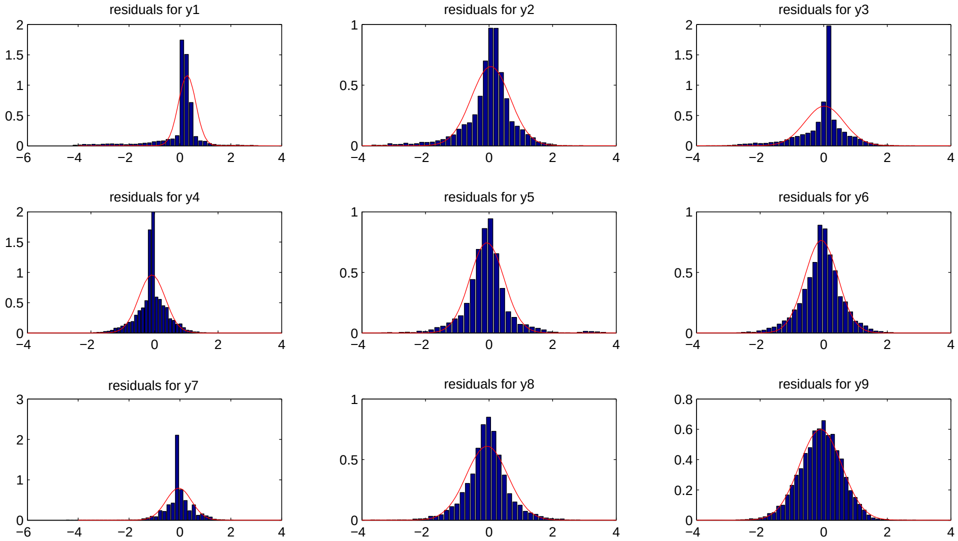


Fig. 4.: Histograms of standardized residuals for Zak and colleagues' data are superimposed with estimated Gaussian distribution. The histograms are largely Gaussian with some expceptions. There are two possible causes for non-Gaussian residuals. One is that Zak's data has long periods of zero as steady states which are then reflected in zero residuals. Another is model mismatch, that is, noise was not Gaussian.

panels of Figure 3 show the histograms of the residuals of the Kalman filter for our synthetic data with a standard normal plot superimposed. From Figure 3 we can see that the four residuals' histograms approximate standard normal distributions well. This adds confidence to the correctness of our algorithm. We also examined the standardized residuals of an estimated model from Zak and colleagues' data and the histograms are plotted in the panels of Figure 4. The histograms are largely Gaussian with some notable exception. While numerically this could be accounted by the fact that long sequences of zeros exist in Zak's data as steady states, which is not likely in the real world data and may be due to unrealistic stochastic noise assumption made in the simulation, there could be some model mismatch. However, from prediction error comparisons we know that constrained EM still could predict much better than unconstrained EM, therefore, even when Gaussian assumption of model noises is violated, constrained EM can still have good predictive power.

2. Prediction Errors

To compare the predictive power of identified models from conventional EM and models from our constrained EM, we used cross-validation. For our synthetic data, we generated a sample of 200 time points, of which the first 100 were used for pa-rameter estimation, and the rest for prediction. Of Zak's data, we selected 110 time points, out of which the first 80 time points were used for estimation and the remaining 30 for validation. The error in prediction is defined to be the difference between the measured gene expression levels and the predicted gene expression levels by the estimated model. Using constrained EM, the error in prediction for four gene expression data in the artificial network from $t = 101$ to $t = 200$ were calculated (Figure 5). In Figure 5, for comparison, we plotted the error in prediction by conventional EM. From the plots, we can see that conventional EM starts off with small errors, and sometimes

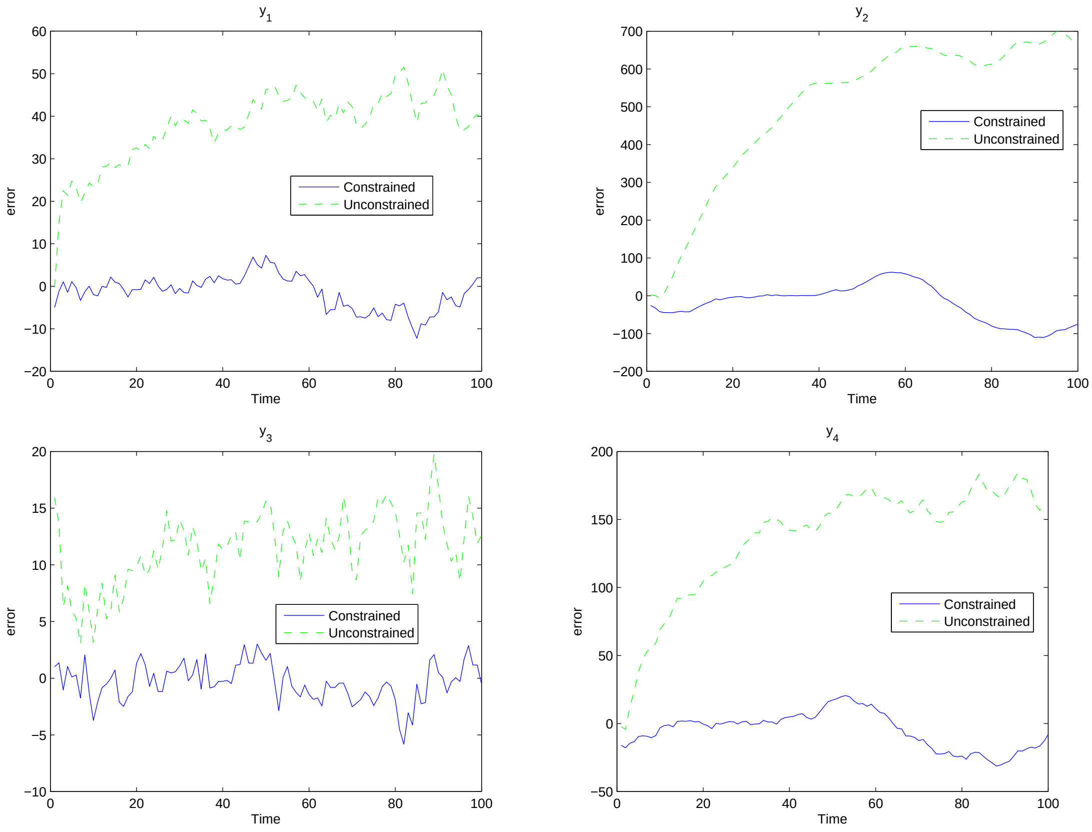


Fig. 5.: These are the errors in the predicted outputs of y_1 and y_2, using conventional EM (dashed) and constrained EM (solid), for the synthetic data. The errors are the differences between the predictions of the estimated model and the observed values. We can see that conventional EM produces models that have large prediction errors and thus poor predictive power.

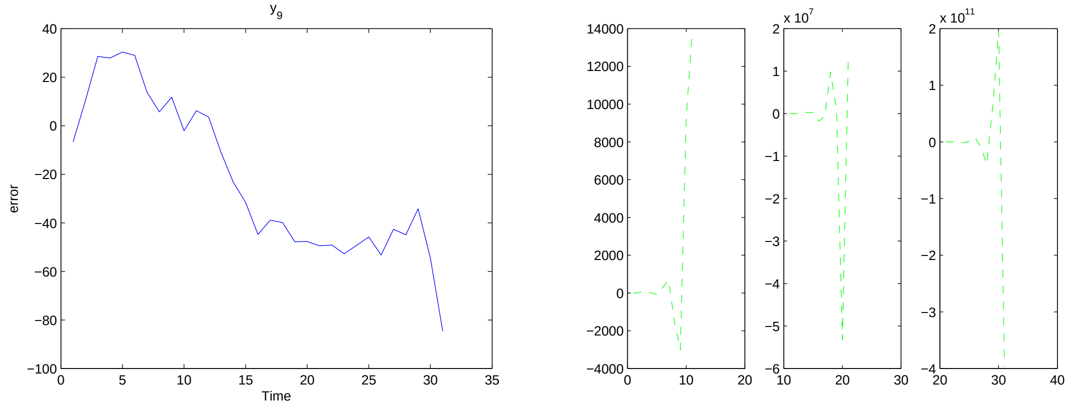


Fig. 6.: The plot on the top is the prediction errors of gene J for the model estimated by the constrained EM, and the three panels on the bottom are the prediction errors of the same output for the model estimated by unconstrained EM. The reason for plotting the three separate panels on the bottom is that the latter errors become too large and completely obscure the earlier errors.

it produces smaller errors initially than our method. But very quickly it strays into wrong directions with larger and larger errors, which makes models estimated from conventional EM having little predictive power. This is even more striking with Zak's data. The constrained EM algorithm yields models with prediction error at worst around 100%, with the output of the worst error plotted in the top panel of Figure 6, while unconstrained EM generates models whose prediction error grows without bound as seen in the bottom panels of Figure 6. In fact, we were forced to cut the plot into three panels so that later values would not obscure earlier ones. This demonstrates that conventional EM, which does not take into account the structure, tends to over-fit.

To further test our method we considered the variation in the estimated values. Since many parameter values could fit data equally well, confidence intervals are usually preferred over a single estimation. Using bootstrapping, we were able to estimate

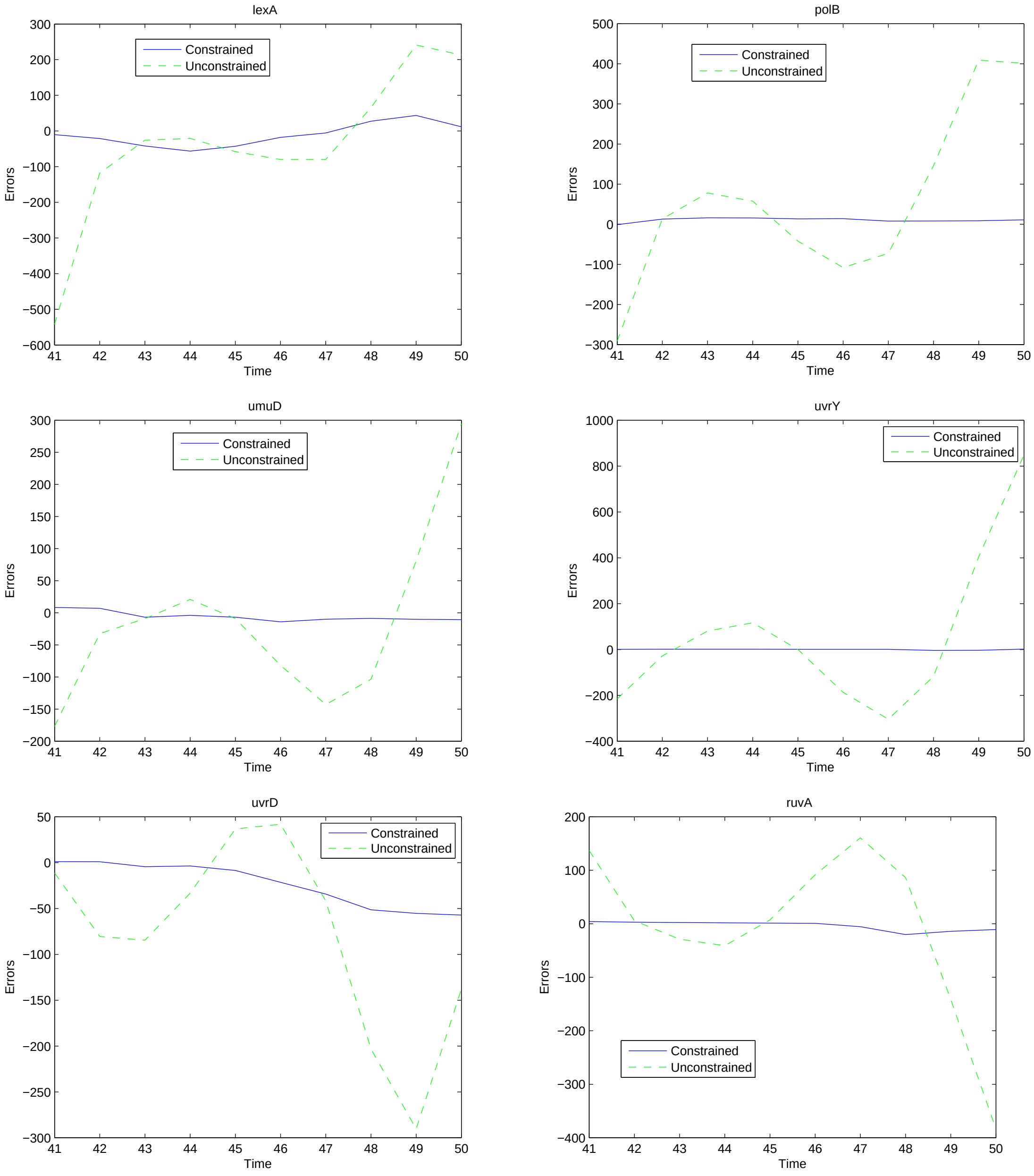


Fig. 7.: The plots are the errors in the predicted outputs of gene lexA, polB, umuD, uvrY, uvrD, and ruvA of the SOS DNA repair network, for conventional (dashed) and constrained EM (solid). The differences between predicted values and measured values are large for the model estimated by the conventional EM.

Table I.: This table compares 95% confidence intervals for our constrained EM and the conventional EM, both using our simulated data. For our system, both constrained and unconstrained methods could estimate eigenvalues with some fidelity, keeping in mind that the true eigenvalues are all 0.8. However, our constrained methods have very tight bounds around the true value while the unconstrained EM has wider intervals.

Confidence Intervals of the Eigenvalues of Our Simulated System					
Lower Bound	-0.0073	0.5636	0.5612	0.2598	Unconstrained EM
Upper Bound	0.9717	0.9934	0.9892	0.9697	
Lower Bound	0.7856	0.7781	0.7843	0.7820	Constrained EM
Upper Bound	0.9073	0.8169	0.8236	0.8312	

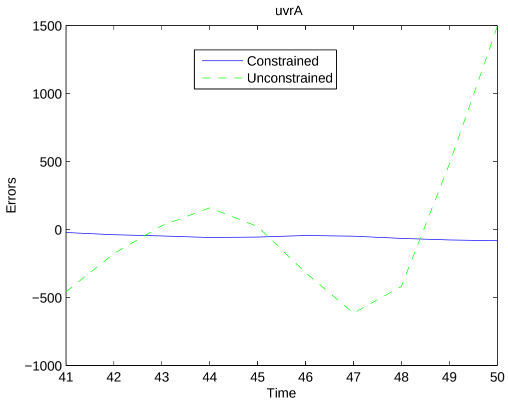


Fig. 8.: The errors in the predicted outputs of gene uvrA of the SOS DNA repair network, for conventional (dashed) and constrained EM (solid).

Table II.: This table compares 95% confidence intervals for our constrained EM and the conventional EM, using Zak's data. For Zak's model, unconstrained EM has one very wide interval which is mostly outside the unit circle and therefore implies that the estimated system is unstable. But as we fail to see any unstable behavior in Zak's data while observing the wildly incorrect predictions of the unconstrained model, we suspect these eigenvalues were not correctly estimated. Our constrained EM yielded much tighter bounds and from them we think the underlying system is mostly likely to be stable, at worst marginally unstable.

Confidence Intervals of the Eigenvalues of Zak's Model Using Unconstrained EM									
Lower Bound	-14.4193	-1.7669	-0.4093	0.0339	0.1385	0.2048	0.2943	0.2110	0.1722
Upper Bound	13.3590	3.2895	1.466	1.2113	1.0416	1.0077	0.9934	0.9661	0.9681
Confidence Intervals of the Eigenvalues of Zak's Model Using Constrained EM									
Lower Bound	0.8914	0.9907	0.9796	0.9454	0.9818	0.5402	0.8477	0.8241	0.52
Upper Bound	0.9037	1.0075	0.9918	0.9695	0.9912	0.9971	0.9972	0.8890	0.8890

95% confidence intervals for all free parameters and the eigenvalues of the estimated system (same as the eigenvalues of matrix A in equation (1)). The idea is that tighter intervals are better than wide intervals and that the estimation methods need to get eigenvalues roughly right, since they are invariant to similarity transformation and they determine important dynamical properties. We found that for almost all free parameters, our constrained EM produced much tighter bounds than those produced by unconstrained EM (see Table III for selected parameters), and for all eigenvalues, our method uniformly produced better bounds (presented in Table I and II). As we can see from Table I and II, estimated eigenvalues for our simulated model all include the true eigenvalues 0.8, but our method has much tighter bounds. For Zak's model, unconstrained EM resulted in one eigenvalue having wide interval while con-strained EM all have tight bounds. That wide interval, much of it outside the unit circle, could be a sign of a misestimated eigenvalue, since it could account for the unbounded prediction error we observed. A misestimated eigenvalue is a serious concern because the trajectory of an LTI system largely depends on its eigenvalues.

To examine the biological applicability of our method and to evaluate its performance with real world data, we applied our constrained EM to the SOS DNA repair network data. Since there are only 50 time points available, we used the last 10 time points for validation purpose. The error in prediction of the seven-gene expression data in the SOS DNA repair network by the constrained and the conventional EM are plotted in Figure 7 and 8. Two features of the plots can be observed. First, the errors oscillate within a certain range rather than blow up in one direction. This suggests that with fewer time points conventional EM performs better than with more data, a classic sign of over-fitting. Second, the constrained EM's estimated model has much smaller deviation from measurement than conventional EM's estimated model, once again demonstrating constrained EM's merits. The figures show that the constrained

Table III.: This table presents a comparison of 95% confidence intervals of diagonal entries of matrix A in our model. For our simulated system, the confidence intervals of our constrained EM are tight around the true value 0.8, while unconstrained EM shows much higher variability. While we do not have the true value for Zak et al.'s model, our constrained EM yield much smaller intervals than unconstrained EM, and, therefore, better estimates.

Confidence Intervals of Diagonal Entries of Matrix A of Our Simulated System					
Lower Bound	-1.3862	-1.4586	-1.9220	-1.3095	Unconstrained EM
Upper Bound	2.8322	3.3041	3.3422	2.6551	
Lower Bound	0.7843	0.7856	0.7820	0.7781	Constrained EM
Upper Bound	0.8236	0.9073	0.8312	0.8169	

Confidence Intervals of Diagonal of Matrix A of Zak's Model Using Unconstrained EM									
Lower Bound	-25.59	-26.71	-22.2107	-35.2	-29.1	-22.0	-25.38	-23.4	-20.3
Upper Bound	25.51	25.7647	27.6785	23.1	30.4192	24.0750	22.65	29.1	30.1
Confidence Intervals of Diagonal of Matrix A of Zak et al.'s Model Using Constrained EM									
Lower Bound	0.521	0.8914	0.8608	0.9778	0.799	0.9796	0.991	0.9818	0.95
Upper Bound	0.596	0.9037	0.8864	1.003	0.883	0.9918	1.0	0.9912	0.97

EM better approximates the true model of the SOS DNA repair network and generate models with superior predictive power.

D. Discussion

It is increasingly recognized that dynamics of genetic networks can impact phenotypes, and the study of dynamics can provide new insights into diseases and potential treatments for the diseases [95]. But before we can analyze genetic networks and propose possible treatments, we need a quantitative model that can predict, with reasonable accuracy, the dynamical behaviors of the genetic network. In this report, we proposed a new method that can learn such a model, a linear state-space system, and tested on both synthetic and real ex-perimental data.

Researchers so far have used the linear state-space model primarily in two ways. Either they are used in black-box dynamical modeling or in inferring a genetic network's structure. Wu et al.'s [69] is representative of the first approach, where the internals of the model is not important but only the dynamical behaviors is, where the number of states is a parameter depending on the data, and where the states have no biological interpretation. This black-box approach is perfectly valid when we have little information regarding the mechanistic details. However, some-times we have structural information for some genetic networks, either through existing knowledge in the literature or ChIP-on-chip experiments, in which case parameter estimation should take the known structure into account to get a better model. A better model is in the sense that the estimated model in the end does not contradict known biological facts represented by the structure and possess better predictive power. Another use of linear state-space modeling is to infer a genetic network's structure. Using the linear model for network inference is especially appealing because the structure and the parameters have a simple relationship: there is a straightforward mapping between parameters and edges in the network. A naive approach would be to estimate a model with all parameters free to be estimated and to consider those parameters whose values are below a threshold as really being zeros and thereby signifying no interaction. We have seen that conventional EM tends to over-fit and produce a model that has limited predictive power. One approach to alleviate over-fitting is to enforce sparsity on the parameters [85]. Another approach is to separate parameter estimation and structural inference, to incorporate structural constraints into parameter estimation, as in Gennemark and Wedlin [10]. Our method can be seen as the parameter estimation part of the overall system identification, which also infers structure. In order to incorporate structural constraints into an identifica-tion of genetic networks in this report, we presented a framework where certain parameters are fixed while others remain variable. Imposing constraints on parameters lead to a set of nonlinear equations to be solved in the M-step. In order to have fast convergence, we intentionally avoided generalized EM (which is slower than our method) or using iterative solutions to the set of nonlinear equations in the M-step (which can also be very slow). Instead, with only mild assumptions about noise, we obtained a closed-form, decoupled, explicit solution to the equations arising from the maximization of likelihood in the M-step. To evaluate the performance of our new method, we applied it to two synthetic data sets and a real world SOS DNA repair network data set. From the results, we can see positive features of incorporating structural constraints in general and constrained EM in particular. First, by incorporating known connectivity between genes, we have better biological realism along with a biological interpretation for the identified model. Second, a state-space model with structural constraints exploits the sparse nature of genetic networks to reduce the number of parameters that need to be estimated. We know that reverse engineering of genetic networks is a grossly underdetermined task, and that we have too few data for a fully connected network. However, the problem can be alleviated if we can effectively incorporate more information by imposing structural constraints. Increasingly, connectivity information is becoming available for more genetic networks [75, 76, 77], and it becomes necessary for us to use this information. Third, by using analytic, explicit solution to the maximization of likelihood, we have fast computation, so that our method can be easily incorporated into larger structural inference algorithms. The last and the most important feature of our work is that by incorporating structural constraints into an identified model we identify models that are not prone to over-fitting, that have better predicative power than unconstrained models, and, therefore, are closer to the true model. Cross-validation using both the synthetic data and the SOS data demonstrated this point. With better predictive power, the identified model can then be used for the analysis of dynamical properties or for the design of control strategies for the genetic network under study. There is considerable work that remains. On the modeling front, our method does not take into account the signs of parameters, which represent whether regulation is activation or inhibition. This is crucial information that should be incorporated as another set of constraints. A problem of a more theoretical nature is that we have as yet no principled way to determine the order of each gene, as too high an order can result in over-fitting and too low an order leaves too much unmodeled dynamics. Finally, as LDS is only a linear approximation of gene regulation systems, care must be exercised in extrapolating results presented here to real world expression data. Much more work on diagnostics and model validation remains as well.

CHAPTER IV

DYNAMICAL PATHWAY ANALYSIS

A. Backgroup

Cell functions are complex temporal processes and should be studied as complex dynamical processes rather than only in their individual steady states. It is increasingly recognized that it is the dynamics and the internal structures of the biological systems that give rise to the functioning of cells [96]. Currently, uncovering co-expressed genes and discovering differentially expressed genes are the primary methods for discovering the role of genes in disease pathogenesis [97], but these methods offer only static views and steady-state explanations and thus fail to account for the transient behaviours that influence phenotypes. Genetic regulatory networks seek to model complex interactions and dynamics of gene regulations. Genetic networks should behave differently in sick cells vs. healthy cells because genes that cause diseases behave fundamentally differently, and that difference should be reflected in their dynamical properties. Dynamical properties of genetic networks such as their response time have been studied mostly in the context of network motifs [98, 99], but now I propose that they be investigated for their difference in normal vs. abnormal cells.

In this chapter I studied four dynamical properties: stability, relative stability, controllability, and transient behaviours (overshoot, settling time, and rise time). Stability governs how a system responds to internal noise and external perturbation and determines whether the system returns to steady states and whether the effect of noise and perturbation diminishes over time. Biologically, an unstable cellular system is very brittle and the slightest disturbance can drive the system beyond tolerance and possibly result in cell death. Prill et al. [100] used stability as a criterion to discern network motifs and their organizing principles, and synthetic biologists are beginning to pay close attention to the stability of their artificial networks [101]. Furthermore, the stability of the system under pure gain feedback control can be analyzed by the root-locus method and the result can be interpreted as a measure of relative stability. In control theory, the root-locus method is a design tool but it is also used as an analytic tool, to see how large a gain can drive the system unstable with feedback loops: the larger margins of stabilizing gains, the better. Related to feedback control, controllability is another pivotal concept in control theory. It and its dual property, observability, were originally conceived as solutions to existence and uniqueness problems of optimal control [102], and the controllability of a dynamical system roughly refers to the ability to move the states of the system around the state space with reasonable efforts. Although controllability is a binary question, there is a measure of the degree of controllability, the idea being that the more controllable a system is the less effort is needed to move the system. Less theoretical than stability and controllability are transient behaviours like settling time and overshoots, which have also received attention from systems biologists [98, 99, 93]. These four dynamical properties are determined by the parameters of the dynamical system and the unknown parameters of biological systems need to be estimated.

Parameter estimation must be done under a particular modelling framework. Several modelling frameworks have been proposed: Boolean networks [59, 103, 104, 60], differential equations [27], S-system [105, 106], and dynamical Bayesian networks [62, 64]. A special case of dynamical Bayesian networks is the state-space model, which has been used to model genetic regulatory networks [69, 72, 73, 107, 67]. A state-space model has states, inputs, and outputs, where hidden states contain complete information of the system driven by the inputs, and the outputs are the measurements made by scientists. In the state-space models of genetic networks, states

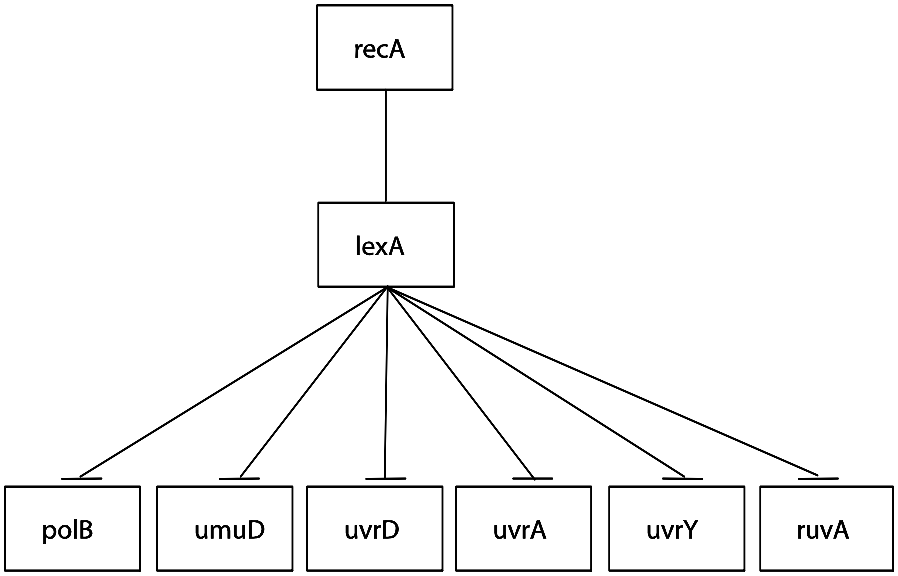


Fig. 9.: The diagram of SOS DNA repair network.

are the regulatory elements, and the inputs and the outputs can be environmental stimuli or expression levels. Because genetic networks have many unknown quantities, state-space models can serve as a good modelling framework.

In this chapter, the parameters in state-space models were estimated from the time course of gene expressions using Kalman filter and the constrained expectation-maximization (EM) algorithm (a modified EM algorithm that incorporates prior knowledge about the structure of genetic networks). The regular EM algorithm is commonly used to estimate parameters in the presence of hidden quantities, and they comprise two steps, E-step (expectation) and M-step (maximization), where the E-step estimates the hidden states, and the M-step the parameters [91]. I applied EM algorithm to three sets of time course data and estimated three genetic networks for analysis. The first network I used is the SOS DNA repair system. The SOS network is a highly conserved system [93, 94] and consists of about 30 genes, the master regulator being gene lexA. The lexA gene inhibits the rest of the SOS

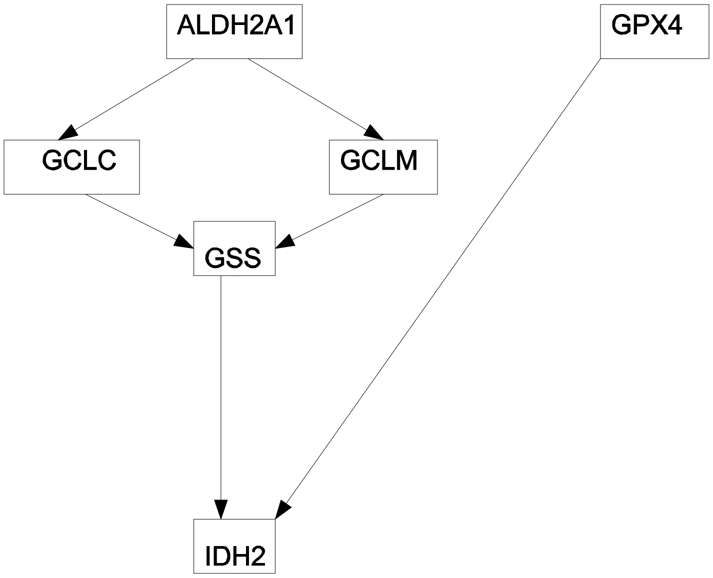


Fig. 10.: The diagram of GSH redox cycle.

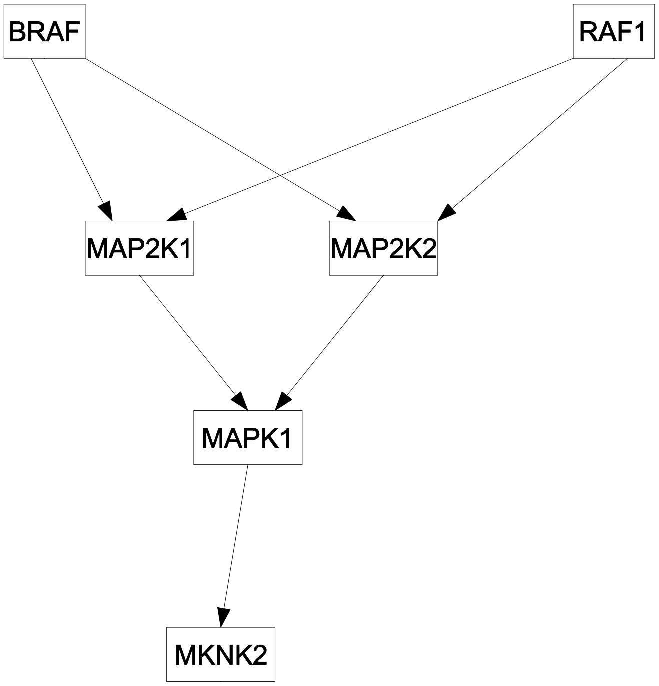


Fig. 11.: The diagram of MAPK network.

network's genes under normal conditions, but when DNA damage is sensed, protein LexA is cleaved and the genes normally suppressed are activated. A diagram of the SOS network with 8 essential genes is shown in Figure 9. Shown in Figure 10 is the second system I modelled, the glutathione (GSH) redox cycle with one gene from the urea cycle that interacts with the redox cycle [108]. The data are from Sciuto et al. [109] who investigated the differential gene expressions in mice lung cells exposed to either carbonyl chloride (phosgene) or normal air. They found elements of the GSH redox cycle differentially expressed, which is not surprising given that the redox cycle is heavily involved in protecting organisms from reactive oxygen species, that it is heavily present in the lung, and that phosgene causes massive lung damages. The third system I investigated is the mitogen-activated protein kinase (MAPK) network in cell lines disturbed by either the wild type HIV type I Vpr or the mutant type R73A or the mutant type R80A. HIV-1 Vpr is an important protein in promoting the pathogenesis of AIDS by facilitating apoptosis and cell cycle stall at G2. Yoshizuka et al. [110] studied the effects of Vpr on MAPK-network-related genes in stalling cell cycle, so they obtained cell lines that can express wild type or mutant Vpr under a tetracycline-inducible promoter. They found that many genes related to the MAPK network differentially expressed when subjected to different types of Vpr. The MAPK network used for this chapter is shown in Figure 11. All those data sets compare the organism's reactions to different environmental perturbations, and from estimated genetic networks I hope to discover the differential dynamical properties of genetic networks under stress.

I applied our framework to three real-world time series datasets above and found differential stability, transient responses, and controllability of genetic networks in normal vs. abnormal cells.

B. Method

1. Data Sources

To test the method on real-world data, I obtained three data sets: *E. coli* under radiation, mice lung cells exposed to the normal air and a toxin, and mammalian cell lines under the influence of various types of Vpr. They were chosen because they all have time course data of organisms reacting to different perturbations and therefore could embody differential dynamical properties.

Ronen et al. [92] irradiated E. coli and used green fluorescent protein (GFP) to obtain the rate of transcription of various genes in the SOS network. Ronen et al. tracked 8 genes of the SOS network as they reacted to different irradiation levels, 5Jm^{-2} and 20Jm^{-2}. Each level had two samples and each sample had 50 time points. They monitored eight genes: uvrD, lexA, umuD, recA, uvrA, uvrY, ruvA, and polB. They performed extensive data pre-processing on the raw data using hybrid Gaussian median filter and polynomial fit for smoothing. They also assumed that the rate of accumulation of GFP was proportional to transcript production, so I shall make the same assumption.

Sciuto et al. [109] measured the effects of carbonyl chloride (phosgene) on mice lung. They exposed the mice to either normal air or phosgene for 20 minutes at a concentration of 32-42mg/m^3 and sacrificed some of the mice at each time point. Each time point had 3 samples for air or phosgene and two replications. All experimental data were collected using Affymetrix Mouse 430A oligonucleotide arrays. The raw data were transformed by adding a constant first, and then they performed a log transformation.

Yoshizuka et al. [110] observed the effect of viral protein R (Vpr) on cell cycles. They transfected plasmids that expressed wild type Vpr and mutated Vpr (R73A and

R80A) into mammalian cells. The microarrays (Hs Operon V2) containing 22,434 oligonucleotide (60- to 70-mer) spots on a glass slide were used to generate the data. There were three replications for each time point.

The analysis in this chapter was done exclusively on the three data sets above.

2. Transfer Functions and Dynamical Properties

A transfer function is a Laplace transform of a linear ordinary differential equation of constant coefficients with zero initial conditions. A single transfer function represents a single-input-single-output (SISO) system and one can obtain a series of transfer functions from a state-space representation of a dynamical system and vice versa [111]. The zeroes are roots of the numerator. The characteristic equation of the transfer function is the denominator equal to zero, and it determines a lot of the dynamical properties of the system. In particular, the roots of the characteristic equation are the poles of the system, which determine the stability of the system and have great influence over other dynamical properties.

a. Stability analysis

For discrete linear time-invariant systems, the system is stable (its steady states do not diverge) if and only if all of the eigenvalues of the state transition matrix or all of the poles of all the transfer functions have magnitude less than 1 [102]. For continuous systems the requirement is that all eigenvalues or poles have negative real part. The simplicity of determining stability belies its importance, for it is one of the most important, best analyzed, and best known dynamical property. Feedback control's first task is to ensure stability and robust control spends a great deal of efforts to ensure stability for uncertain models [112, 113].

b. Root-locus plots

The root-locus method graphically illustrates how the poles of the closed-loop system change as the gain of a pure gain controller is varied. Later it is generalized to show how the roots change as any parameter of the characteristic equation varies. The parameters must be in the form of $1 + KG(s) = 0$ where K is the gain (or the parameter), $G(s)$ is a transfer function, and s is a complex variable. The gain is required to be non-negative but this is not a problem because I could just make $-G(s)$ the new nominal system. I only need two criteria to determine the trajectory

$$|KG(s)| = 1$$

$$\angle KG(s) = 180^\circ + k360^\circ$$

where k is some integer. The root-locus plot lies in the complex plane. The path of roots starts at the open-loop poles and ends at the open-loop zeros, and if part of the path lies on the real axis, then it lies to the left of an odd number of poles [111].

c. Controllability

Controllability is a concept central in systems theory. It is about the ability of a system to move from any initial state to any final state with final control in finite time. The controllability matrix is defined as $H = [B\,AB\,A^2B\ \cdots]$ for a linear time invariant system (LTI) of $\dot{x} = Ax + Bu$ where u is $m \times 1$ an vector, x an $n \times 1$ vector, matrix, A an $n \times n$, and B an $n \times m$ matrix. If the controllability matrix has full rank, then LTI is controllable; otherwise it is uncontrollable. Another way of saying that a matrix is not full rank is that it is singular, and due to numerical inaccuracy of digital computers and model uncertainty, condition number is used to measure how close to singularity a matrix is. The condition number of a matrix is defined to be $\|H\| \cdot \|H^{-1}\|$ where $\|\cdot\|$ is any matrix norm. I used 2-norm in this chapter. The condition number of the controllability matrix can be seen as a measure of the degree of controllability. The larger the condition number is, the greater the inputs are needed to reach a target state, even though reaching nearby states requires no great efforts.

d. Unit-step signal and step-response plots

A unit-step signal is a constant signal of strength one. The step response is the output of a dynamical system in response to a unit-step input. The step-response plot graphically gives much information about the dynamical properties of a system. The most important property the step response manifests is stability. A stable systems plot will converge to a steady state while an unstable system will diverge or oscillate. Step-response plots also show settling time, rise time, and percent overshoot. Settling time measures how fast the system achieves the steady state and rise time how quickly the system responds to perturbation. Rise time is defined to be the time for the output to go from 10% to 90% of the steady state. Settling time is defined to be the time for the output to reach and stay within a 2% neighborhood of the steady-state value. Percent overshoot or undershoot is the percentage of the maximum or minimum minus the steady state and divided by the steady state. Rise time is generally associated with the speed of the dynamics, that is, how fast the system responds to inputs, while overshoot and settling time measure how close the transient responses stay within the vicinity of the steady states. They are also inversely related in nature, that is, both rise time and settling time cannot be kept small: decrease in one necessitates increase in the other if nothing else changes. The root-locus technique is one way to use feedbacks to design a closed-loop system with better rise time, better settling time, and better overshoot.

e. Parameter estimation

The issue of paramter estimation for linear state-space dynamic model of genetic regulatory networks with structural constraints has been dealt with in the previous chapter, so it will not be treated here again.

C. Results

1. Models of Genetic Networks and Their Application to Real Data Sets

I modeled genetic networks as dynamical systems, more specifically as linear state-space systems. A linear state-space model of dynamical systems can be written as

$$x(t+1) = Ax(t) + Bu(t) + w$$
$$y(t) = Cx(t) + Du(t) + v$$

where $x(t)$ is the state vector, $y(t)$ the output vector, and $u(t)$ the input vector, all at time t; w and w are independent noise terms assumed to be white Gaussian with zero means and covariance Q and R respectively. Matrix A is called the state transition matrix, B the input matrix, C the output matrix, and D the feed-forward matrix. Matrices A, B, C, D and covariance matrices Q and R together make up the parameters of the dynamical system.

The states represent the biological forces that regulate gene regulation; they describe the behaviours of gene transcription but are hidden. The outputs denote the gene expression levels and are measured, and it is assumed that the expression level of a gene is determined by the state of the regulated gene. The inputs can be any external stimuli that influence gene regulation: substances like drugs, proteins, RNAs, or expression levels of other genes.

2. Estimated System

For the SOS system, x_2 is the discretized first derivative of x_1, whereas x_1 is the expression level of gene lexA, x_3 gene polB, x_4 gene umuD, x_5 gene uvrD, x_6 gene uvrA, x_7 gene uvrY, and x_8 gene ruvA. The outputs are the measured expression levels of the seven genes listed above, and the input is gene recA. In Figure 12 and Figure 13, I included the estimated outputs and the measured outputs superimposed into one plot, as well as estimation errors in a separate panel for each gene. From the plots I can see that the estimated trajectory largely follows measured values. The estimated system parameters are listed below for the low level of radiation:

$$\begin{aligned}
x_1(t+1) &= x_1(t) + x_2(t) \\
x_2(t+1) &= -0.17x_1(t) + 0.59x_2(t) + 0.084u(t) \\
x_3(t+1) &= 0.009x(t)_1 + 0.81x_3(t) \\
x_4(t+1) &= 0.037x_1(t) + 0.74x_4(t) \\
x_5(t+1) &= -0.007x_1(t) + 0.964x_5(t) \\
x_6(t+1) &= -0.037x_1(t) + 0.965x_6(t) \\
x_7(t+1) &= 0.008x_1(t) + 0.697x_7(t) \\
x_8(t+1) &= 0.009x_1(t) + 0.621x_8(t).
\end{aligned}$$

$$\begin{aligned}
y_1(t) &= x_1(t) \\
y_2(t) &= x_3(t) \\
y_3(t) &= x_4(t) \\
y_4(t) &= x_5(t) \\
y_5(t) &= x_6(t) \\
y_6(t) &= x_7(t) \\
y_7(t) &= x_8(t).
\end{aligned}$$

Fig. 12.: Comparison of estimated and observed expression levels of SOS system under low radiation level. For each gene of the SOS system, I have superimposed the estimated expression levels on measured expression levels and plotted the error in estimation in a separate panel. I have done this for the low radiation level data set in Figure 12. The estimations genetrally show good behaviors. Sub-figures a, b, c, d, e, f, g are for the low radiation level data set, and plotted genes lexA, polB, umuD, uvrD, uvrA, uvrY, and ruvA, respectively. Each gene has two plots; the bottom panel shows estimated expression level superimposed on measured expression level, while the top panel is the estimation error.

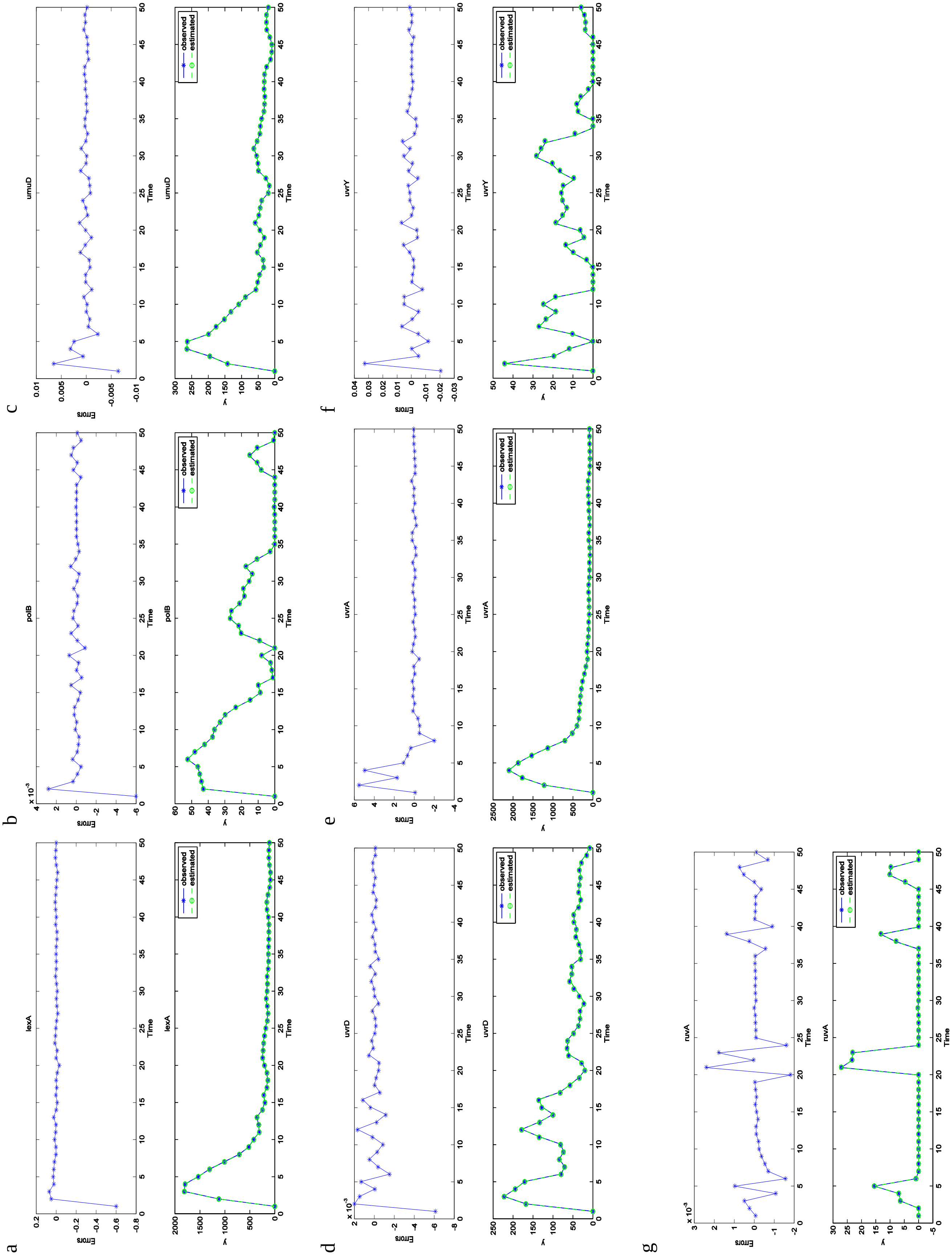
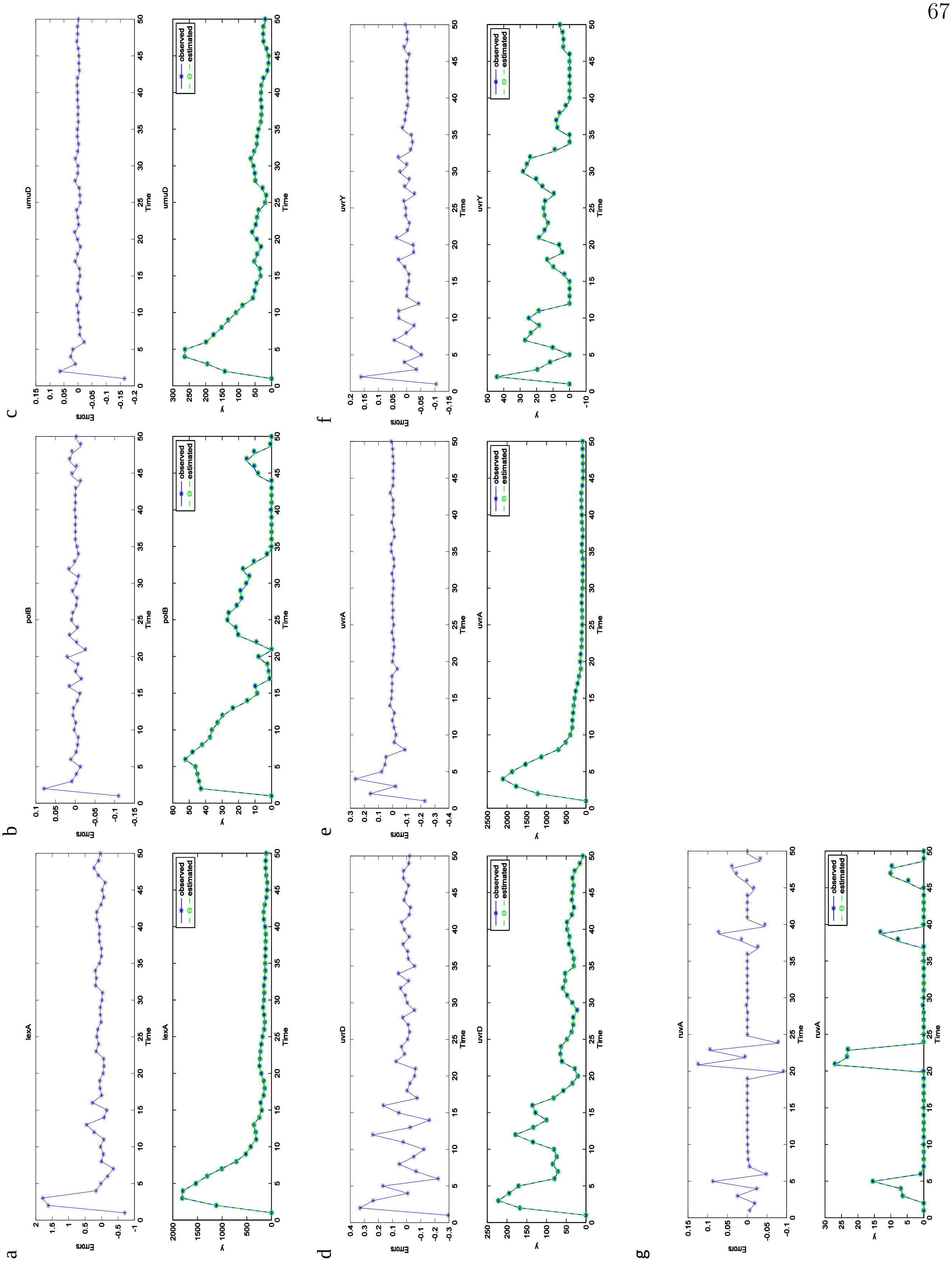

a
lexA
Errors
y
Time
observed
estimated
b
polB
x 10^-3
c
umuD
d
uvrD
e
uvrA
f
uvrY
g
ruvA

Fig. 13.: Comparison of estimated and observed expression levels of SOS system under high radiation level. I have superimposed the estimated expression levels on measured expression levels and plotted the error in estimation in a separate panel for the high radiation level data set in this figure. a, b, c, d, e, f, g are for the high radiation level data set plotted genes lexA, polB, umuD, uvrD, uvrA, uvrY, and ruvA, respectively. Each gene has two plots; the bottom panel shows estimated expression level superimposed on measured expression level, while the top panel is the estimation error.


a
lexA
Errors
Time
y
observed
estimated
b
polB
c
umuD
d
uvrD
e
uvrA
f
uvrY
g
ruvA

For the high level of radiation, the estimated system is

$$\begin{aligned}
x_1(t+1) &= x_1(t) + x_2(t) \\
x_2(t+1) &= -0.242x_1(t) + 0.329x_2(t) - 0.014u(t) \\
x_3(t+1) &= 0.008x_1(t) + 0.832x_3(t) \\
x_4(t+1) &= 0.051x_1(t) + 0.653x_4(t) \\
x_5(t+1) &= 0.01x_1(t) + 0.889x_5(t) \\
x_6(t+1) &= -0.366x_1(t) + 1.22x_6(t) \\
x_7(t+1) &= 0.002x_1(t) + 0.906x_7(t) \\
x_8(t+1) &= 0.001x_1(t) + 0.629x_8(t).
\end{aligned}
\qquad
\begin{aligned}
y_1(t) &= x_1(t) \\
y_2(t) &= x_3(t) \\
y_3(t) &= x_4(t) \\
y_4(t) &= x_5(t) \\
y_5(t) &= x_6(t) \\
y_6(t) &= x_7(t) \\
y_7(t) &= x_8(t).
\end{aligned}$$

For the GSH redox cycle there are two inputs, gene ALD2A1 as u_1 and GPX4 as u_2. All the states were modelled with second order dynamics so the last four states x_5, x_6, x_7, and x_8 are the discretized first derivatives of x_1, x_2, x_3, and x_4, respectively. Here, gene GCLC is x_1, gene GCLM x_2, gene GSS x_3, and gene IDH2 x_4. The estimated system for exposure to normal air is

$$\begin{aligned}
x_1(t+1) &= x_1(t) + x_5(t) \\
x_2(t+1) &= x_2(t) + x_6(t) \\
x_3(t+1) &= x_3(t) + x_7(t) \\
x_4(t+1) &= x_4(t) + x_8(t) \\
x_5(t+1) &= -0.37x_1(t) - 0.39x_5(t) + 0.814u_1(t) \\
x_6(t+1) &= -0.429x_2(t) - 0.006x_6(t) + 0.632u_2(t) \\
x_7(t+1) &= 0.095x_1(t) - 0.015x_2(t) - 0.217x_3(t) - 0.128x_7(t) \\
x_8(t+1) &= 0.753x_3(t) - 0.409x_4(t) - 0.867x_8(t) + 0.017u_2(t).
\end{aligned}
\qquad
\begin{aligned}
y_1(t) &= x_1(t) \\
y_2(t) &= x_2(t) \\
y_3(t) &= x_3(t) \\
y_4(t) &= x_4(t).
\end{aligned}$$

For exposure to phosgene, the estimated model is

$$\begin{aligned}
x_1(t+1) &= x_1(t) + x_5(t) \\
x_2(t+1) &= x_2(t) + x_6(t) \\
x_3(t+1) &= x_3(t) + x_7(t) && y_1(t) = x_1(t) \\
x_4(t+1) &= x_4(t) + x_8(t) && y_2(t) = x_2(t) \\
x_5(t+1) &= -0.141x_1(t) + 1.95x_5(t) + 0.77u_1(t) && y_3(t) = x_3(t) \\
x_6(t+1) &= -0.076x_2(t) - 1.09x_6(t) - 0.13u_2(t) && y_4(t) = x_4(t). \\
x_7(t+1) &= 0.05x_1(t) - 0.161x_2(t) - 0.09x_3(t) - 0.705x_7(t) \\
x_8(t+1) &= 0.336x_3(t) - 0.179x_4(t) - 1.126x_8(t) + 0.103u_2(t).
\end{aligned}$$

As for the MAPK system, the inputs are gene BRAF as u_1 and gene RAF1 as u_2. The states x_1, x_2, x_3 and x_4 are genes MAP2K1, MAP2K2, MAPK1, and MKNK2, respectively; the other four states are the discretized first derivatives as in the system for the GSH redox cycle. The estimated system for the wild type Vpr is

$$\begin{aligned}
x_1(t+1) &= x_1(t) + x_5(t) \\
x_2(t+1) &= x_2(t) + x_6(t) \\
x_3(t+1) &= x_3(t) + x_7(t) && y_1(t) = x_1(t) \\
x_4(t+1) &= x_4(t) + x_8(t) && y_2(t) = x_2(t) \\
x_5(t+1) &= -1.48x_1(t) - 1.32x_5(t) + 0.14u_1(t) + 0.2u_2(t) && y_3(t) = x_3(t) \\
x_6(t+1) &= -0.098x_2(t) - 0.52x_6(t) - 0.079u_1 - 0.314u_2(t) && y_4(t) = x_4(t), \\
x_7(t+1) &= 0.498x_1(t) + 0.052x_2(t) - 0.215x_3 - 0.618x_7(t) \\
x_8(t+1) &= 0.123x_3(t) - 0.169x_4(t) - 0.602x_8(t),
\end{aligned}$$

and for the R73A mutant

$$\begin{aligned}
x_1(t+1) &= x_1(t) + x_5(t) & & \\
x_2(t+1) &= x_2(t) + x_6(t) & & \\
x_3(t+1) &= x_3(t) + x_7(t) & y_1(t) &= x_1(t) \\
x_4(t+1) &= x_4(t) + x_8(t) & y_2(t) &= x_2(t) \\
x_5(t+1) &= -1.098x_1(t) - 1.087x_5(t) + 0.068u_1(t) - 0.23u_2(t) & y_3(t) &= x_3(t) \\
x_6(t+1) &= -0.76x_2(t) - x_6(t) - 0.06u_1(t) - 0.03u_2(t) & y_4(t) &= x_4(t), \\
x_7(t+1) &= 0.073x_1(t) + 0.5x_2(t) - 0.355x_3(t) - 0.647x_7(t) & & \\
x_8(t+1) &= 0.79x_3(t) - 0.876x_4(t) - 1.179x_8(t), & &
\end{aligned}$$

and for R80A mutant

$$\begin{aligned}
x_1(t+1) &= x_1(t) + x_5(t) & & \\
x_2(t+1) &= x_2(t) + x_6(t) & & \\
x_3(t+1) &= x_3(t) + x_7(t) & y_1(t) &= x_1(t) \\
x_4(t+1) &= x_4(t) + x_8(t) & y_2(t) &= x_2(t) \\
x_5(t+1) &= -0.582x_1(t) - 0.821x_5(t) + 0.082u_1(t) - 0.085u_2(t) & y_3(t) &= x_3(t) \\
x_6(t+1) &= -0.28x_2(t) - 0.836x_6(t) - 0.149u_1(t) - 0.009u_2(t) & y_4(t) &= x_4(t). \\
x_7(t+1) &= -0.019x_1(t) + 0.273x_2(t) - 0.056x_3(t) - 0.249x_7(t) & & \\
x_8(t+1) &= 0.112x_3(t) - 0.467x_4(t) - 1.248x_8(t). & &
\end{aligned}$$

Although the number of parameters is small compared with the number of states, which agrees with the knowledge that genetic networks are sparse [85], it is still hard to see at a glance whether they differ in any fundamental way. For that, I must apply systematic analysis to the estimated systems.

Table IV.: Differential stability of the SOS network

Low Dosage	High Dosage
0.8117	0.8321
0.7367	0.6530
0.9637	0.8893
0.9652	1.2216 (**unstable)**
0.6969	0.9062
0.6219	0.6291
0.7952 + 0.3630i	0.6647 + 0.3597i
0.7952 - 0.3630i	0.6647 - 0.3597i

3. Differential Stability of Systems under Different Perturbations

Stability is a very important property of a biological system, for an unstable system puts great stress on neighbouring systems and may even lead to cell death. A system is stable if it will converge to steady states after disturbance; it is unstable otherwise. The stability of a discrete linear system can be determined by the eigenvalues of its state transition matrix A: if all the eigenvalues are within the unit circle in the complex plane, then the discrete system is stable. The eigenvalues of the three analyzed networks are listed in Table IV, V,and VI, and their implications discussed below.

I analyzed the SOS DNA network under low and high dosage of radiation and discovered that the network was stable for low dosage and unstable for high dosage. I found that the eigenvalues of SOS network under low dosage of radiation to have the eigenvalues' norm all less than one, and therefore the network was stable. On the other hand, the SOS network was unstable under high dosage of radiation, as the norm of one of its eigenvalues was greater than one.

Table V.: Differential stability of GSH redox cycle

Normal Air	Phosgene
-0.6141	2.0830 (**unstable**)
0.1177	-1.0383 (**unstable**)
0.4803 + 0.3199i	-0.7561
0.4803 - 0.3199i	1.0512 (**unstable**)
0.7467	0.9120
0.7542	0.8696
0.4972 + 0.4196i	1.0470 + 0.2711i
0.4972 - 0.4196i	1.0470 - 0.2711i

I also analyzed the redox cycle in mice lung cells that were exposed to either carbonyl chloride (phosgene), an industrial toxin, or normal air; and I found that GSH redox system in normal lung cells was stable – all eigenvalues were within the unit circle, and that the network exposed to toxin was unstable – some eigenvalues were outside the unit circle. Whether the unstable detoxification system contributed to the death of mice exposed to phosgene is not yet known, but Sciuto et al. [109] speculated that the poison might have overwhelmed the detoxification system.

I also analyzed the activity data from the MAPK network in mammalian cells that expressed either wild type Vpr, mutant R73A Vpr, or mutant R80A Vpr; and I found that both the wild type and R73A produced stable behaviours, and R80A caused the network to become unstable. A stable MAPK network helps the virus most, for Yoshizuka et al. [110] found the HIV virus uses MAPK network to cause cell cycle G2 arrest, and over-expression of MAP2K2 reversed the arrest.

Table VI.: Differential stability of the MAPK network

Wild type	R73A	R80A
0.8527	0.7448	-1.0169 (**unstable**)
0.8862	-0.0437 + 0.0925i	-0.4078
-0.1615 + 0.3646i	-0.0437 - 0.0925i	-0.2023
-0.1615 - 0.3646i	-0.6472	0.5867
-0.4884	-0.3913	0.9534
-0.4601	0.4680	0.7685
0.9324	0.4877	-0.6676
-0.4477	-0.4916	0.8315

4. Differential Relative Stability Analyzed via Root Locus

The relative stability of genetic networks is also important; it is a measure of robustness. I studied relative stability by examining the stability margins of pure gain feedback loops through root-locus plots. Given a dynamical system, one forms a feedback loop from the output to the input through only a pure gain controller. Depending on whether the control signal is negated as it is fed into the input, the feedback can be positive (not negated) or negative (negated). The original system is called the open-loop system, and its zeros and poles are the open-loop zeros and poles; the zeros and poles of the overall system are called the closed-loop zeros and poles. A dynamical system's zeroes are the roots of the numerator of the transfer function (for an explanation of the transfer function, see Methods), and the poles are the roots of the denominator. The stability of closed-loop systems depend on the closed-loop poles. The root-locus method generates a plot that traces the closed-loop poles as the gain of the controller is varied, and the portion of gains that make the closed-loop stable is called the stability margin. In the root locus plot, the open-loop zero is represented by a circle (◯), the open-loop pole by a cross (×), and if there is a zero-pole cancellation I will see a circle and a cross on top of each other (⊗). The root-locus method can only study systems with single input and single output (SISO), but the dynamical properties of SISO systems is a reflection of the overall system's dynamical properties, so that the performance of the SISO system will manifest itself in the overall system's performance.

In the SOS DNA repair network, the recA to uvrA SISO system showed differential root-locus plots, depending on radiation levels. Their respective root-locus plots, for both negative and positive feedbacks, are shown in Figure 14. Under low level of radiation, I found that the SISO system was comfortable with positive feedback, which had larger margin of stabilizing gains, whereas negative feedback allowed far narrower choices. Under high level of radiation, the opposite was true: positive feedback had no stabilizing gain whereas negative feedback had a large margin. The need for positive feedback loop in low radiation level is an interesting discovery from our root-locus analysis, because it runs counter to the common perception that negative feedback loop promotes stability and positive feedback loop leads to instability. Perhaps under low radiation level, the SOS network is not sufficiently stimulated and positive feedback fully activates the network which then leads to overall stability.

In the GSH redox network, I discovered that the ALDH2A1 to IDH2 SISO system showed a simpler but more striking difference under different environmental conditions. When exposed to normal air, the SISO system was stable and the root-locus plot in Figure 15 shows that sizeable gain values do not destabilize the closed-loop system, which represents a nice scenario, because the subsystem can sustain a lot of stress. But, as I can see in Figure 15 and Figure 15, the same SISO system, when exposed to toxin, not only had an unstable open-loop system, but the closed-loop sys-

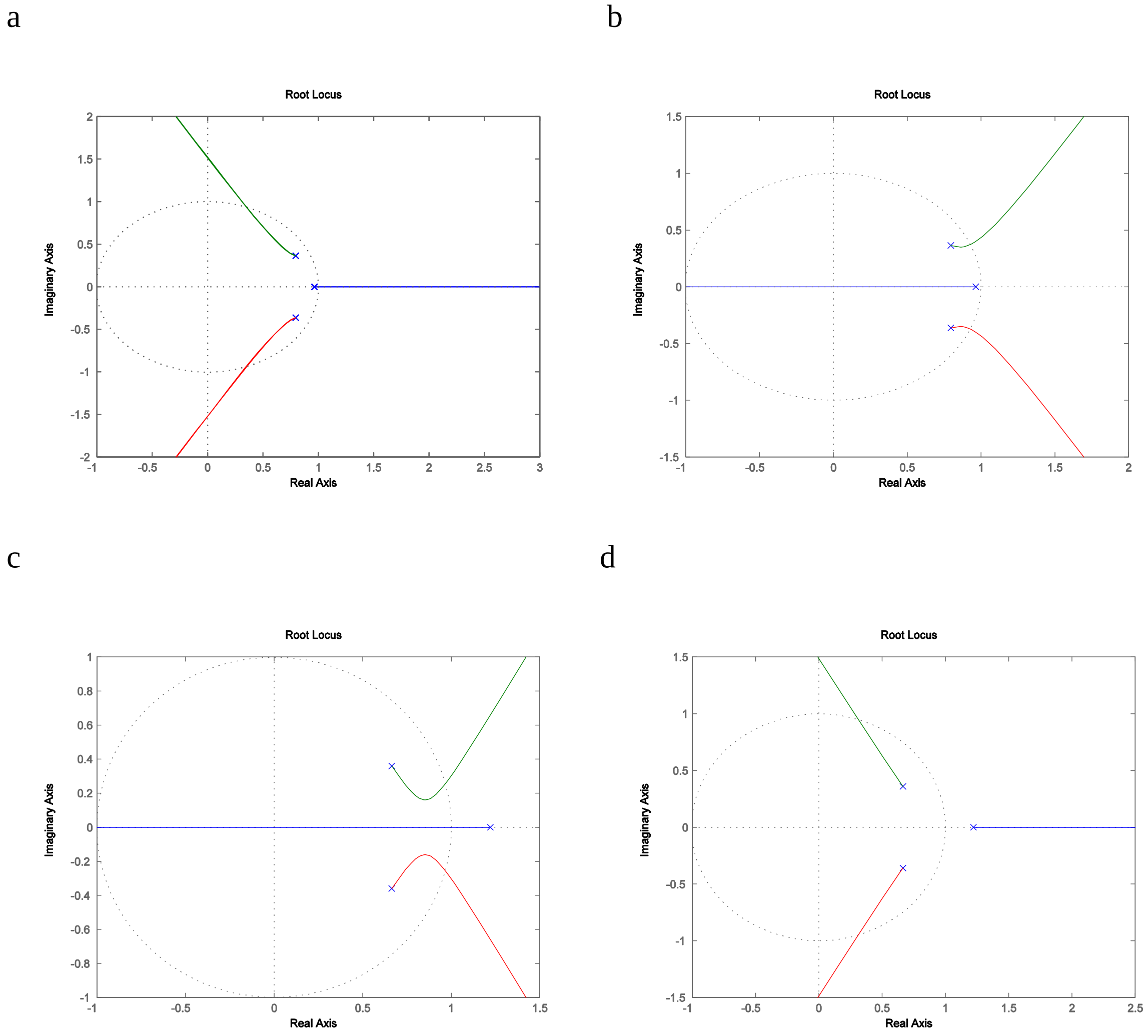


Fig. 14.: Root locus plots of recA to uvrA SISO system. These plots trace the poles of the closed-loop system as the gainis varied from zero to infinity. The trajectories start at the open-loop poles which are represented by the cross, and could end at the open-loop zeros which are represented by an open circle, or they could go on infinitely in some direction. The different colours represent distinct trajectories of different closed-loop poles. a. This is the root locus plot of recA to uvrA system under low level of radiation with negative feedback, where the locus on the real axis goes out of the unit circle quickly and therefore shows small stability margins. (The dotted circle is the unit circle.) b. This is the root locus plot of recA to uvrA system under low level of radiation with positive feedback, with some stability margins. c. This is the root locus plot of recA to uvrA system under high level of radiation with negative feedback, where a good portion of all three loci stays within the unit circle and therefore exhibits large stability margins. d. This is the root locus plot of recA to uvrA system under high level of radiation with positive feedback, which has no stability margin.

tem also remained unstable no matter what value of the gain was, positive or negative. This means that not only the ALDH2A1 to IDH2 SISO system was very unstable, but that a higher order controller must be used to produce a stable closed-loop system, a sign of very serious damage.

I also found that MAPK network in mammalian cell lines subject to different versions of Vpr of HIV type I virus had similarly different root locus plots, which are shown in Figure 16. The RAF1 to MKNK2 SISO system was stable under both the wild type and the R73A mutant Vpr perturbation, and both showed comfortable margin of gain values for which the closed-loop system was stable. The SISO system under R80A mutant protein exhibited a stable closed-loop system with only a small margin of gain with positive feedback and none with negative feedback. If that small margin does not include a gain that can produce a closed system with satisfactory performance, then a higher order controller is called for.

5. Differential Degree of Controllability

Since one goal of systems biology is to aid the development of therapeutic treatments, which in the context of genetic networks is to bring the network from undesirable states to healthy states by manipulating inputs, the relative ease of moving around in the state space is an important issue. The ability to move a system from one point in the state space to another in finite time with only finite inputs is called controllability, which is a pivotal concept in linear time systems theory [102]. Controllability can be tested by the rank of controllability matrix; if the controllability matrix is of full rank, then the system is controllable, otherwise uncontrollable. Beyond the binary test (controllable or not) there are also degrees of controllability. The condition number of the controllability matrix can be considered as a measure of the degree of controllability, the bigger the number the less the controllability. A system with

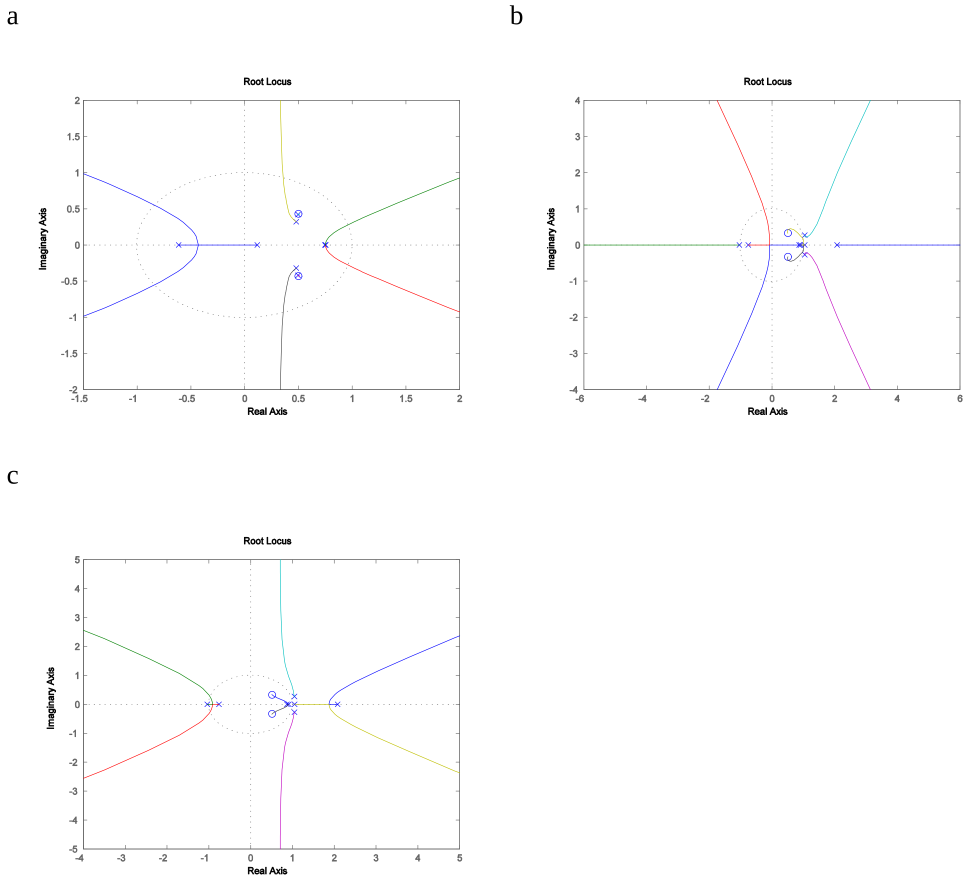


Fig. 15.: Root locus plots of recA to uvrA SISO system. a. Root locus plot of ALDH2A1 to IDH2 system exposed to normal air with negative feedback is shown, with large stability margins. b. Root locus plot of ALDH2A1 to IDH2 system exposed to poisonous air with negative feedback, where the locus on the positive real axis is entirely outside of the unit circle and therefore it has no stability margin. c. Root locus plot of ALDH2A1 to IDH2 system exposed to poisonous air with positive feedback, showing no stability margin because of the locus at the right.

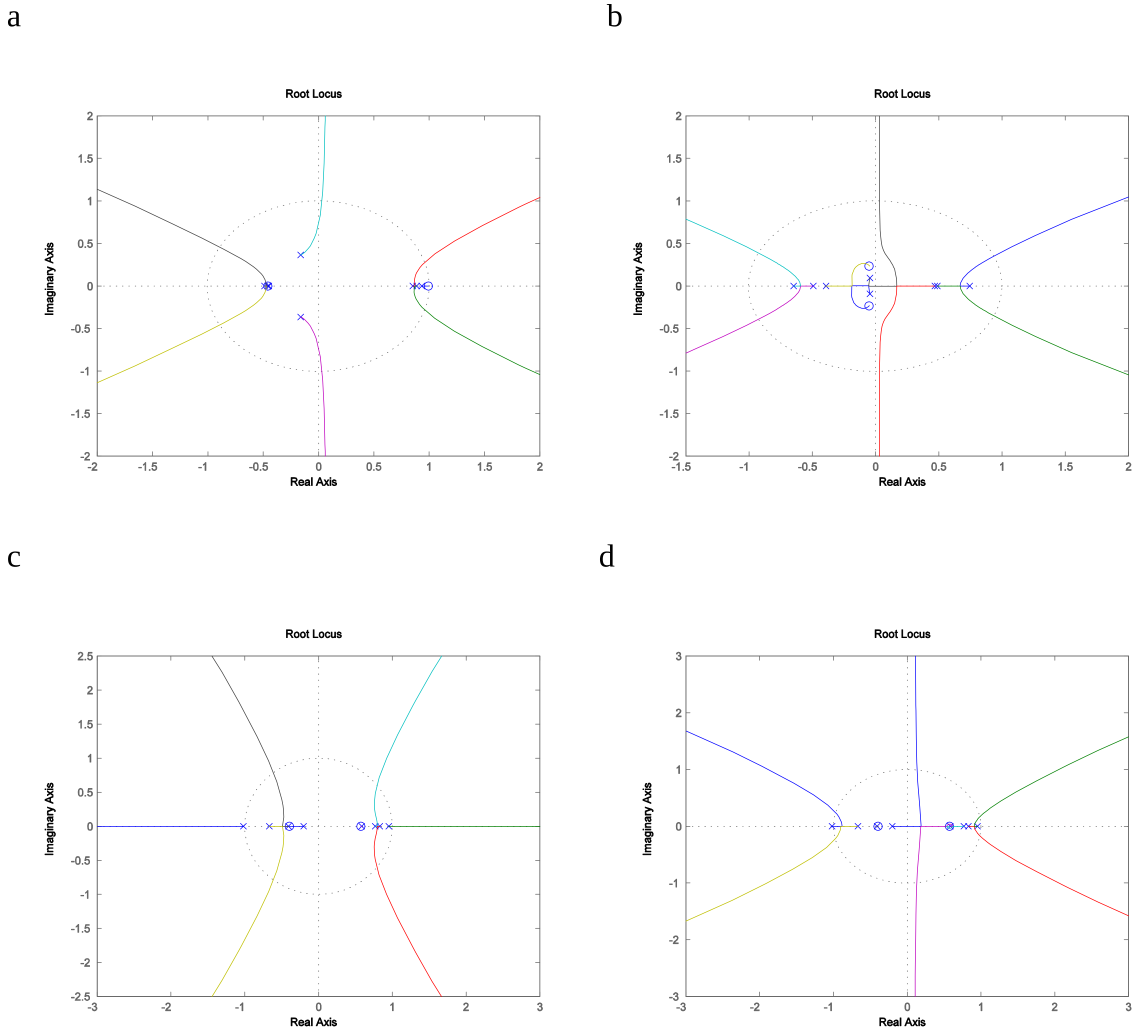


Fig. 16.: Root locus plots of RAF1 to MKNK2 SISO system. a. Root locus plot of RAF1 to MKNK2 system perturbed by wild type Vpr with negative feedback where a large portion of the locus can be seen within the unit circle. b. Root locus plot of RAF1 to MKNK2 system perturbed by R73A mutant Vpr with negative feedback showing very good stability margins. c. Root locus plot of RAF1 to MKNK2 system perturbed by R80A mutant Vpr with negative feedback, where there is basically no stability margin due to the two loci on the real axis. d. Root locus plot of RAF1 to MKNK2 system perturbed by R80A mutant Vpr with positive feedback and small stability margins.

less controllability may require much greater inputs to achieve the desired final state, which could be a problem as the inputs for biological systems are drugs, radiation therapy, things in limited supply and subject to cost factors. As I will see, different systems could have radically different degrees of controllability.

Although I found all the three genetic regulatory networks controllable under all circumstances, their condition numbers differed, for one significantly. I discovered that the SOS DNA repair system under high dose of radiation had a condition number of 2.8×10^9 for its controllability matrix, and that under low dosage the condition number was 2.6×10^9. The similarly large condition numbers suggest the SOS system under study is difficult to control; whether this is due to radiation is not known. On the other hand, in mice lung exposed to normal air I saw that the redox system had a condition number of 567 for its controllability matrix, and that those exposed to toxin had 70267. The different condition numbers peg the redox system as much more difficult to control after exposure to poison, perhaps due to damages or the fact that the network was being overwhelmed by the effects of the toxin. The third network, the MAPK network in mammalian cell lines, was found to have a condition number of 62.15 when exposed to the wild type Vpr, 88.5 for those exposed to the R80A mutant, and 285.4 for the R73A mutant. It is obvious that R73A mutant results in a stodgier MAPK network than other variants, but overall the MAPK system retains good controllability, making it a good target for treatment.

6. Differential Transient Responses

To study cell functions as temporal processes means I must take stock of transient behaviors in addition to steady states. One way to characterize transient behaviors is through the transient response of the dynamical system to inputs like step input and impulse input, but because the step responses and impulses responses give same information for linear systems, I will concentrate on the step input responses. A step input is a constant input, a unit step, a constant unity. The rise time is a measure of the speed of the dynamics, and the settling time and the overshoot gauge how close to the steady state the transient behaviors are. Of course, systems that exhibit differential stability will have different transient responses, but because differential stability is addressed earlier, I will disregard any difference in transient responses due to stability difference.

The transient responses are by their nature studied as input-output pairs, also called a single-input-single-output (SISO) system. Although I will look at individual SISO system extracted from multiple-input-multiple-output (MIMO) systems, the transient responses are still the intrinsic properties of the original system, and differential transient responses suggest fundamentally different dynamical behaviors of the original system in response to external perturbations.

The SOS DNA repair network has only one SISO system, besides those due to differential stability, that exhibited differential rise time and settling time, the recA to uvrD system. The SISO system, when exposed to high radiation dosage, was almost twice as fast as the system exposed to low dosage of radiation, to reach their respective steady states. This suggests that the SOS system needs uvrD to respond faster to, and therefore has faster dynamics under, higher levels of radiation. With no overshoot in both cases and a smaller settling time for a higher dosage, the recA to uvrD system under high radiation level stayed closer to the steady states. The rise time and settling time are listed in Table VII.

The MAPK network in mammalian cells exhibited differential transient responses to three types of Vpr of HIV type I virus. The BRAF to MAP2K2 SISO system displayed slower dynamics and were more distant from the steady state under the wild type than both mutants, and among the mutants, R73A had faster dynamics

Table VII.: Different transient responses of the SOS network

	recA to uvrD
Low Radiation Dosage	RiseTime:58.1993
	SettlingTime:108.1459
High Radiation Dosage	RiseTime:17.9888
	SettlingTime:35.8853

and better ability to stay close to the steady state. On the other hand, the BRAF to MAPK1 systems transient behaviors in response to the wild type Vpr were dominated by a overshoot, and with its long settling time the systems transient responses were far from the steady state. The system perturbed by the wild type protein also had faster dynamics due to its smaller rise time, and the R73A mutant produced a system that had relatively fast dynamics and transient response closer to the steady state. The R80A mutant resulted in a system with slow dynamics and transient responses distant from the steady state with its relative large rise time and settling time and no overshoot. The respective rise time, settling time, and overshoots are in Table VIII.

Although overshoot is generally considered undesirable in engineering (whether fast dynamics or staying close to the steady states are good or bad depends on the circumstances and cannot be determined a priori;) a large overshoot can be a fast way of signalling, or it can be an unbearable disturbance to cells. But being aware of the difference in transient responses is the first step toward devising treatment strategies that shape biological systems dynamics to our liking.

Table VIII.: Different transient responses of the MAPK network

	BRAF to MAP2K2	BRAF to MAPK1
Wild type	RiseTime:31.5 SettlingTime:56.5 Overshoot:0	RiseTime:0.26 SettlingTime:76.1 Overshoot:440.4
R73A Mutant	RiseTime:2.6 SettlingTime:5.8 Overshoot:0	RiseTime:8.2 SettlingTime:17.0 Overshoot:0
R80A Mutant	RiseTime:11.7 SettlingTime:21.7 Overshoot:0	RiseTime:48.5 SettlingTime:90.2 Overshoot:0

D. Discussion

Discovering differentially expressed genes and clustering co-expressed genes into functional groups have given researchers hints about the role of genes in pathogenesis. However, with increasing recognition that cell functions are temporal processes and that the dynamics of gene expression levels and gene interactions play a vital role in determining the health of the organism [96, 114], there is a need to distinguish peculiar dynamical behaviors that result in sickness from those that do not. Dynamical properties succinctly characterize dynamical behaviors, and differential dynamical properties of gene networks can be seen as a natural extension of differentially expressed genes.

In this chapter I analyzed the dynamical properties of genetic networks, such as their stability, their closed-loop stability embodied in the root-locus plot, their step responses, and their controllability. First, I estimated the state-space models of three genetic networks: the SOS DNA repair network, the GSH redox cycle system, and the MAPK network; then I performed analysis on the estimated models. From the preliminary results, I found that significant differences in dynamical properties exist in all three networks.

All three genetic networks exhibited differential stability. Stability is a fundamental dynamical property in any dynamical system. A dynamical system is unstable if it diverges or oscillates after being subjected to perturbations. An unstable system is sensitive to perturbation or noise, and it will have erratic behaviors, possibly causing irreparable cell damage, leading to impairment of cell functions and maybe even cell death. A stable genetic network on the other hand confers a degree of robustness against noise on the overall organism. Recently Hornstein and Shomron [115] proposed that miRNAs play a stabilizing role for a number of genetic networks and the stability was necessary for the proper functioning of organisms. It would be interesting to see whether restoring stability to an organisms genetic networks restores the organisms health.

Besides stability, I also studied relative stability. Root-locus plots track the stability of the closed-loop system under the influence of a pure gain controller for single-input-single-output (SISO) systems, and they can be seen as a measure of the relative stability of the SISO system. As biological systems are often under control of other, bigger systems, wide margins of stabilizing gains give more leeway to, and can sustain some stress from, the controlling systems, and therefore they are more relatively stable than those with narrow margins. The redox cycle system in mice lungs is the clearest example. Exposed to normal air, the ALDH2A1 to IDH2 system was itself stable and the closed-loop system was stable for all possible gains, which makes this SISO system robust in normal tissues. But when exposed to toxin, not only was it unstable in itself, but no gain value could make the closed-loop system stable, which makes the system brittle. Systems that change from high relative stability to low relative stability can be considered for association with diseases, because they impact the outer loop systems and could make the overall system unstable. However, relative stability is not the only thing root-locus plots can show. In the recA to uvrA SISO system of the SOS network, positive feedback enabled a lot of stabilizing gains for the SISO system exposed to low level of radiation, as opposed to the same system exposed to high level of radiation, which needed negative feedback for large margins of stabilizing gains. This may portend drastic changes in the outer loops, as changing from promotion to inhibition is not easy for biochemical reactions, and it could be a major sign that this system is associated with unhealthy conditions.

The last dynamical property I looked at was controllability. Therapeutic treatments can be seen as pushing gene expression levels from unhealthy states to healthy states, and controllability is a theoretical guarantee that there are possible inputs that can achieve healthy states. Although I found all systems to be controllable, I did find different degrees of controllability. The condition number of the controllability matrix was taken as a measure of degree of controllability and the redox cycle system in mice lung exhibited over 100 times difference in its condition number, suggesting a much higher possibility that unacceptably large inputs are required to move the system into desired states.

Of course much work remains. So far in this chapter I have only analyzed a small number of dynamical properties while many more remain. Robustness is an important property that some consider an organizing principle of complex biological system [116, 117], yet I have not investigated it. There is also the issue of the robustness of estimation. Due to inherent noise in measurements, there are inevitable uncertainties in any parameter estimation. In general, increasing the sample size will increase the reliability of the results for dynamic properties. Another way to deal with this is to obtain confidence intervals for estimated parameter values. However, confidence intervals on individual parameters do not directly translate into confidence intervals for dynamical properties, especially because I have imposed constraints on the parameter space. This should be a topic for further study.

On the issue of scalability, it is known that the number of floating point operations roughly grows to the cubic power of the number of states [91], assuming that the number of states is larger than either the number of inputs or that of outputs. I have implemented our method in Matlab and for the systems studied in this chapter computation time is around ten minutes on a 1.6GHz Core Duo laptop, so I expect our program to have no difficulty with a network with dozens of genes. For large systems, I should investigate hierarchical system identification method [118].

E. Conclusion

Dynamical properties are considered to be pivotal in determining cellular functions such as apoptosis, cell division, proliferation, etc. [119], and it follows that differential dynamical properties can serve as important indicators for discovering the role of specific biological processes in causing the malfunction of cells. Only by comparing fundamentally different dynamical behaviours between normal and abnormal cells can I begin to untangle the complex interactions and roles of genes in pathogenesis. This will not only add to our understanding of diseases but could also be a step toward effective treatments.

CHAPTER V

OPTIMAL LINEAR QUADRATIC CONTROL OF GENETIC NETWORKS

A. Introduction

Study of optimal control of genetic regulatory and signal transduction networks serves two purposes:

1. Unravel the underlying design principles of cells in an evolutionary context [93, 120, 121],
2. Design the treatment strategies for curing diseases [122, 123, 124, 125, 126].

Evolution is long thought as optimizing a fitness measure that balance competeing pressures [120], and if the fitness measure is dynamic behavior, then evolution can be seen as solving a dynamic optimization problem. Although that is a new and promising research problem, this chapter of my dissertation is concerned with the second purpose, optimal control of genetic regulatory and signal transduction networks for treatment strategy.

Expression profiles of genes or proteins are closely related to the phenotype of cells and are the fingerprints of cellular physiology [127, 128]. Germline or somatic mutations lead to subsequent transcriptional and translational changes which in turn will change the dynamic behaviors of genetic networks, and finally affect the phenotype of cells and cause diseases. Therapeutic interventions such as radiation, drug, gene therapy, and small RNA interference try to alter gene expressions from an undesirable or abnormal state to a desirable or normal state. Gene regulation is a complex biological system that is highly organized into genetic networks and must accomplish complex tasks with high accuracy. To change individual genes' expression and to prevent and control the undesirable dynamic behavior of genetic networks are fundamental to therapeutic treatment using genetics [129]. Our ultimate goal is to be able to design medical treatments directly intervening in biological networks and altering their dynamic behaviors.

In the past several years, researchers in optimal control of genetic regulatory and signal transduction networks have focused on using variants of Boolean networks as their models [130, 131, 132, 122, 124, 133]. Boolean network models assume that each gene has discrete states, often limited to two states, ON or OFF. If the gene is expressed then it takes the value ON, otherwise, it takes the value OFF. Optimal control of a Boolean network approach is formulated as minimizing a performance index over a period of time by external control signals. A disadvantage of the Boolean network is its inability to include many of the details of cellular processes [134]. A more detailed approach is the rate-equation approach, in which ordinary differential equations are used to model chemical kinetics of reactions [135, 136]. The rate-equation approach allows a more accurate physical representation of biological networks, and it permits a large body of analytical techniques of dynamic systems theory to be applied to the analysis of the dynamics of biological networks [137]. In fact, a majority of works on signal transduction networks uses the rate-equation approach. However, the rate-equation approach is computationally intensive and requires kinetic parameters which are often difficult to obtain outside of the few well-studied networks.

A linear dynamic system is simpler than rate equations but still retains considerably more details of biological networks than Boolean networks. Therefore, we formulate the control of genetic regulatory and signal transduction networks as a optimal linear quadratic control problem in modern control theory, which is very popular in engineering [138]. Optimal linear quadratic control is based on state-space models. We already note the manifold advantages of the state-space models in the introduction, so we will belabor the point here. Instead, we just reiterate here that the development of RNA interference for repression of gene expression [139, 140] and synthetic biology in designing artificial regulatory circuits [141, 134, 142, 143] will open a new avenue for targeted gene therapy [144] and that the state-space models are particularly suitable for modeling these therapeutic strategies.

Gene expression patterns embody the fundamental state of cells and can be used to characterize genetic manipulations, drug actions, and cellular responses to various environmental stimuli. Changes in expression levels of genes and proteins often lead to transition of cells from being normal to being abnormal and therefore differentially expressed genes may serve as drug targets [145]. It is hypothesized that the disease is caused by germline or somatic mutations such as point mutations or insertions or deletions or chromosomal translocations that result in the subsequent transcriptional and translational alterations [146, 147]. The purpose of drug actions is to compensate for the resulting molecular changes.

One way to do so is to transform cells from an undesirable state to a desirable one by altering gene or protein expressions [148]. This implies that a drug development problem can be formulated as a control problem of a complex biological system. A dynamic control system involves three types of variables: input, output, and state variables [149, 150]. In the drug development problem, the variables that determine the states of genetic regulatory system can be taken as the state variables. Therapeutic interventions such as radiation, chemo-therapy, siRNA and gene therapy provide tools to influence the states and can therefore be taken as inputs. The expression levels of the gene and proteins which can be measured are the outputs.

The control problem of the biological system (or specifically, gene regulatory system) is to choose the input values to be fed to the biological system so that the performance of the system (e.g. desired states of gene expressions, the steady-state and transient response of genetic networks, etc.) are optimum (or satisfies prespecified condition) with respect to a performance criterion [151]. In this dissertation, I choose a quadratic function as the performance index and a linear state-space model to study the dynamic behavior of the genetic regulatory and signal transduction networks. This chapter is organized as follows. In section B, we formulate the control problem of genetic regulatory and signal transduction networks as an optimal tracking control problem and provide the necessary background for finite time-horizon and infinite time-horizon optimal quadratic tracking control problems. In section C, the proposed optimal tracking control of genetic networks is applied to three real data sets. In section D, we discuss and summarize the results and address further research issues.

B. Methods

1. Formulation of Optimal Control of Genetic Regulatory and Signal Transduction Networks

This section closely follows Lewis and Syrmos' book. [149]

The task of gene regulation and signal transduction is to control production of mRNA and protein. In order to reduce the complexity of the problems, we formulate the control of genetic regulatory and signal transduction networks as an optimal control problem of linear deterministic systems with quadratic performance index and bounded controls [138]. In general, a controlled system consists of three components: the states of the system, the outputs and external control signals also called inputs. The outputs of the genetic regulatory and signal transduction networks here are the expressions of the genes in the networks. The environmental stimulus, drugs, small RNA interference can be taken as external inputs. The goal of the controller is to find an admissible control function that minimizes the performance index starting with given initial states.

Consider the following continuous-time linear dynamic system for modeling genetic regulatory and signal transduction networks:

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) \\ x(t_0) &= x_0,\end{aligned} \tag{5.1}$$

where $x(t)$ is an n-dimensional state vector that describes the behavior of gene regulation, but is hidden; $u(t)$ is a r-dimensional input vector that influence gene regulation, things like environmental forces, drugs, proteins, RNAs, or expression levels of connected genes or nutrition; $y(t)$ is an m-dimensional output vector, for example, gene expressions and intermediate phenotypes; and $A(t)$,$B(t)$, and $C(t)$ are $n \times n$, $n \times r$, and $m \times n$ matrices, respectively. Matrix $A(t)$ is called a state transition matrix whose elements denote the regulatory strength of one gene on another gene, $B(t)$, input to state matrix whose elements quantify the effects of the inputs on the states of the system, and $C(t)$, state to output matrix whose elements describe the relations between the states and measured gene expression levels. This is a time-varying system because the parameters $A(t)$, $B(t)$, and $C(t)$ are functions of time. Performance index is defined as

$$J = \Phi[x(t_f), t_f] + \int_{t_0}^{t} L(x(t), u(t), t)\mathrm{d}\, t$$

which measures deviation of states from the desired conditions or the errors between the current abnormal states and desired normal states of the systems. Here, function $\Phi(\cdot)$ penalizes the final state while $L(\cdot)$ penalizes the interior states. We will discuss specific functional form of $\Phi(\cdot)$ and $L(\cdot)$ in the specific problems in the next two sections. The optimal control problem is to find the optimal control $u(t)$ and the corresponding state trajectory $x(t)$, $0 \leq t \leq t_f$ satisfying equations (5.1) that minimizes the performance index J, which could lead to the transition of the system from undesirable states to desirable states. In general, the performance index J can be any nonlinear function of states and inputs, but in this dissertation, I will focus on the quadratic performance index.

2. The Continuous-Time Linear-Quadratic Regulator

We will discuss the regulator first because it is the simpler problem than optimal tracking and may serve as a gentle introduction to tracking. The goal of a regulator is to maintain the system at a desired equilibrium state. To achieve this goal, the quadratic performance index is

$$J = \frac{1}{2}x^T(t_f)Fx(t_f) + \frac{1}{2}\int_{t_0}^{t_f} [x^T(t)Q(t)x(t) + u^T(t)R(t)u(t)]\mathrm{d}\,t \tag{5.2}$$

where F is a real symmetric positive semidefinite matrix, $Q(t)$ is a real symmetric positive semidefinite matrix, and $R(t)$ is a real symmetric positive definite matrix. Our goal is to select control input $u(t)$ for minimizing the performance index J.

To design a regulator, we first construct the system Hamiltonian:

$$H(x(t), u(t), \lambda(t)) = \frac{1}{2}x^T(t)Q(t)x(t) + \frac{1}{2}u^T(t)R(t)u(t) + \lambda^T(t)[A(t)x(t) + B(t)u(t)]$$

where $\lambda(t)$ is an $n \times 1$ adjoint vector, also known as the Lagrange multiplier. The reason to include $\lambda(t)$ is to make sure that the systems equations in equation (5.2) are obeyed for the optimal solution. $Q(t)$ and $R(t)$ are parameters chosen by designers to balance the importance of minimizing states and minimizing inputs. The minimum principle requires that

$$\frac{\partial H}{\partial u} = R(t)u(t) + B^T(t)\lambda(t) = 0 \tag{5.3}$$

$$\frac{d\lambda(t)}{dt} = -\frac{\partial H}{\partial x} = -Q(t)x(t) - A^T(t)\lambda(t) \tag{5.4}$$

with terminal condition $\lambda(t_f) = Fx(t_f)$. Solving equation (5.3), the optimal control is given by

$$u(t) = -R^{-1}(t)B^T(t)\lambda(t). \tag{5.5}$$

We assume that the n-dimensional state and adjoint vectors can be related by the following linear transformation:

$$\lambda(t) = P(t)x(t). \tag{5.6}$$

Lewis and Syrmos call $P(t)$ the solution to the Riccati differential equation that will be defined below, and according to Bryson and Ho [152], it can also be derived if one assumes possession of a set of linearly independent solutions for $x(t)$ and $\lambda(t)$. In any event, the functional form assumed above lends itself to the sweep method for solving ordinary differential equations. Substituting equation (5.6) into equation (5.5) yields

$$u(t) = -R^{-1}(t)B^T(t)P(t)x(t). \tag{5.7}$$

It follows from equations (5.1) and (5.7) that

$$\dot{x}(t) = [A(t) - B(t)R^{-1}(t)B^T(t)P(t)]x(t). \tag{5.8}$$

Substituting equation (5.7) into equation (5.4), we obtain

$$\dot{P}(t)x(t) + P(t)\dot{x}(t) = -Q(t)x(t) - A^T(t)P(t)x(t). \tag{5.9}$$

Substituting equation (5.8) into equation (5.9), we have

$$\dot{P}(t)x(t) + P(t)[A(t) - B(t)R^{-1}(t)B^T(t)P(t)]x(t) = -[Q(t) + A^T(t)P(t)]x(t),$$

which implies that

$$[\dot{P}(t) + P(t)A(t) + A^T(t)P(t) - P(t)B(t)R^{-1}(t)B^T(t)P(t) + Q(t)]x(t) = 0. \quad (5.10)$$

Since $x(t) \neq 0$, then we have the following matrix Riccati differential equation:

$$-\dot{P}(t) = P(t)A(t) + A^T(t)P(t) - P(t)B(t)R^{-1}(t)B^T(t)P(t) + Q(t). \quad (5.11)$$

Combining the terminal condition $\lambda(t_f) = FX(t_f)$ and equation (5.6), we obtain

$$P(t_f) = F.$$

This is the boundary condition for the Riccati differential equation, and we "sweep" backward from the terminal point to the starting time, similar to treating every time-point as a terminal point, so solution $P(t)$ is like carrying the terminal condition backward in time. In summary, these are the steps to obtaining the optimal control sequences:

1. Solve matrix Riccati differential equation,$-\dot{P}(t) = P(t)A(t) + A^T(t)P(t) - P(t)B(t)R^{-1}(t)B^T(t)P(t) + Q(t)$ with boundary condition $P(t_f) + F$;

2. Solve the differential equation $\dot{x}(t) = [A(t) - B(t)R^{-1}(t)B^T(t)P(t)]x(t)$, $x(t_0) = x_0$;

3. the input $u(t)$ is given by $u(t) = -R^{-1}(t)B^T(t)P(t)x(t)$; and

4. the optimal performance index is given by $J = \frac{1}{2}x^T(t_0)P(t_0)x(t_0)$.

3. Optimal Linear Quadratic Tracking (LQT)

The aim of optimal linear quadratic tracking is to force the system to track a desired trajectory. Assume that $y_r(t)$ is the desired output. Let

$$e(t) = y_r(t) - y(t).$$

The performance index is defined as

$$J = \frac{1}{2}e^T(t_f)Fe(t_f) + \frac{1}{2}\int_{t_0}^{t_f} [e^T(t)Q(t)e(t) + u^T(t)R(t)u(t)]dt. \tag{5.12}$$

Optimal linear quadratic tracking is to find a control signal that minimizes the performance index J. To achieve this, we first construct the system Hamiltonian:

$$H(x(t), u(t), \lambda(t)) = \frac{1}{2}e^T(t)Q(t)e(t) + \frac{1}{2}u^T(t)R(t)u(t) + \lambda^T(t)[A(t)x(t) + B(t)u(t)].$$

The minimum principle requires

$$\frac{\partial H}{\partial u} = R(t)u(t) + B^T(t)\lambda(t) = 0 \tag{5.13}$$

$$\frac{\mathrm{d}\,\lambda(t)}{\mathrm{d}\,t} = -\frac{\partial H}{\partial x} = C^T(t)Q(t)[y_r(t) - C(t)x(t)] - A^T(t)\lambda(t), \tag{5.14}$$

with the boundary condition

$$\lambda(t_f) = C^T(t_f)F[C(t_f)x(t_f) - y_r(t_f)]. \tag{5.15}$$

Solving equation (5.13), we have the input signal

$$u(t) = -R^{-1}(t)B^T(t)P(t)x(t).$$

Note that the state and adjoint vectors can be related by the following linear transformation:

$$\lambda(t) = P(t)x(t) - g(t), \tag{5.16}$$

where $g(t)$ is an additional to account for the reference trajectory in the performance index and the terminal condition, which makes $\lambda(t)$ an affine function of state instead of a linear function as in LQR. Substituting equation (5.16) into equation (5.14), we obtain

$$\begin{aligned}&\frac{\mathrm{d}\,P(t)}{\mathrm{d}\,t}x(t) + P(t)\frac{\mathrm{d}\,x(t)}{\mathrm{d}\,t} - \frac{\mathrm{d}\,g(t)}{\mathrm{d}\,t}\\ &= C^T(t)Q(t)y_r(t) - C^T(t)Q(t)C(t)x(t) - A^T(t)[P(t)x(t) - g(t)]\end{aligned} \tag{5.17}$$

However,

$$\begin{aligned}\frac{\mathrm{d}\,x(t)}{\mathrm{d}\,t} &= A(t)x(t) + B(t)u(t)\\ &= [A(t) - B(t)R^{-1}(t)B^T(t)P(t)]x(t) + B(t)R^{-1}(t)B^T(t)g(t).\end{aligned} \tag{5.18}$$

Substituting equation (5.18) into equation (5.17) yields

$$\begin{aligned}&\left[\frac{\mathrm{d}\,P(t)}{\mathrm{d}\,t} + P(t)A(t) + A^T(t)P(t) - P(t)B(t)R^{-1}(t)B^T(t)P(t) + C^T(t)Q(t)C(t)\right]\\ &+P(t)B(t)R^{-1}(t)B^T(t)g(t) - \frac{\mathrm{d}\,g(t)}{\mathrm{d}\,t}\\ &= -A^T(t)P(t)x(t) + A^T(t)g(t) + C^T(t)Q(t)Y_r(t),\end{aligned} \tag{5.19}$$

which implies that

$$-\frac{\mathrm{d}\,P(t)}{\mathrm{d}\,t} = P(t)A(t) + A^T(t)P(t) - P(t)B(t)R^{-1}(t)B^T(t)P(t) + C^T(t)Q(t)C(t) \tag{5.20}$$

$$-\frac{\mathrm{d}\,g(t)}{\mathrm{d}\,t} = [A(t) - B(t)R^{-1}(t)B^T(t)P(t)]^T g(t) + C^T(t)Q(t)Y_r(t), \tag{5.21}$$

with boundary conditions

$$P(t_f) = C^T(t_f)FC(t_f)$$

and

$$g(t_f) = C^T(t_f)FY_r(t_f).$$

In summary, the steps to obtaining the optimal inputs are:

1. Solve differential equations

$$\begin{aligned}
-\frac{\mathrm{d}\,P(t)}{\mathrm{d}\,t} &= P(t)A(t) + A^T(t)P(t) - P(t)B(t)R^{-1}(t)B^T(t)P(t) + C^T(t)Q(t)C(t) \\
-\frac{\mathrm{d}\,g(t)}{\mathrm{d}\,t} &= [A(t) - B(t)R^{-1}(t)B^T(t)P(t)]^T g(t) + C^T(t)Q(t)Y_r(t) \\
P(t_f) &= C^T(t_f)FC(t_f) \\
g(t_f) &= C^T(t_f)FY_r(t_f),
\end{aligned}$$

   where $P(t)$ carries backward the terminal condition and $g(t)$ is the additional term to account for the reference trajectory in the performance index and the terminal condition;

2. Solve differential equation

$$\frac{\mathrm{d}\,x(t)}{\mathrm{d}\,t} = [A(t) - B(t)R^{-1}(t)B^T(t)P(t)]x(t) + B(t)R^{-1}(t)B^T(t)g(t)$$

   with initial condition

$$x(t_0) = x_0$$

   ; and

3. Calculate the optimal input siganl by $u(t) = -R^{-1}(t)B^T(t)[P(t)x(t) - g(t)]$.

If we assume that all the parameters matrices, A, B, C, D, Q, and R are constant, then when $t_f \to \infty$, we have

$$PA + A^TP - PBR^{-1}B^TP + C^TQC = 0$$

$$g = [PBR^{-1}B^T - A^T]^{-1}C^TQy_r$$

$$u(t) = -R^{-1}B^TPx(t) + R^{-1}B^Tg$$

$$\frac{\mathrm{d}\,x}{\mathrm{d}\,t} = [A - BR^{-1}B^TP]x(t) + BR^{-1}B^Tg$$

$$x(t_0) = x_0.$$

C. Experiments and Results

In this section, three real world examples are given to illustrate how to use optimal linear quadratic tracking for optimal control of genetic regulatory and signal transduction networks. All data are gene expressions although the methods can also be applied to other genetic and molecular biology problems.

1. Scleroderma Fibroblasts in Response to Perturbation of Environmental Stimuli

Diseases are believed to arise from dysregulation of biological systems (pathways) perturbed by environmental triggers. It is systems dynamics that play an essential role in giving rise to cellular function/dysfunction such as growth, differentiation, division and apoptosis. As a proof of principle, we examine TGFβ signal transduction pathways in human fibroblasts from the autoimmune fibrosing disease, scleroderma (SSc) in response to perturbation by silica. SSc is a typical complex disease in which fibrosis occurs in multiple organs. The major source of fibrosis in SSc is from production of collagens from fibroblasts. Fibroblasts obtained from SSc patients appear to be genetically engineered to produce more collagens and cytokines [153]. The biological system of fibroblasts reacting to silica exposure must involve complex regulations and coordination of molecules to maintain their desired status. I will design a linear quadratic tracking of TGFβ pathway to reduce the concentration of COL1A2, COL3A1. I will study a subnetwork of TGFβ pathway as shown in Figure 17, where the expression levels of the genes in the normal and SSc fibroblasts are shown with-

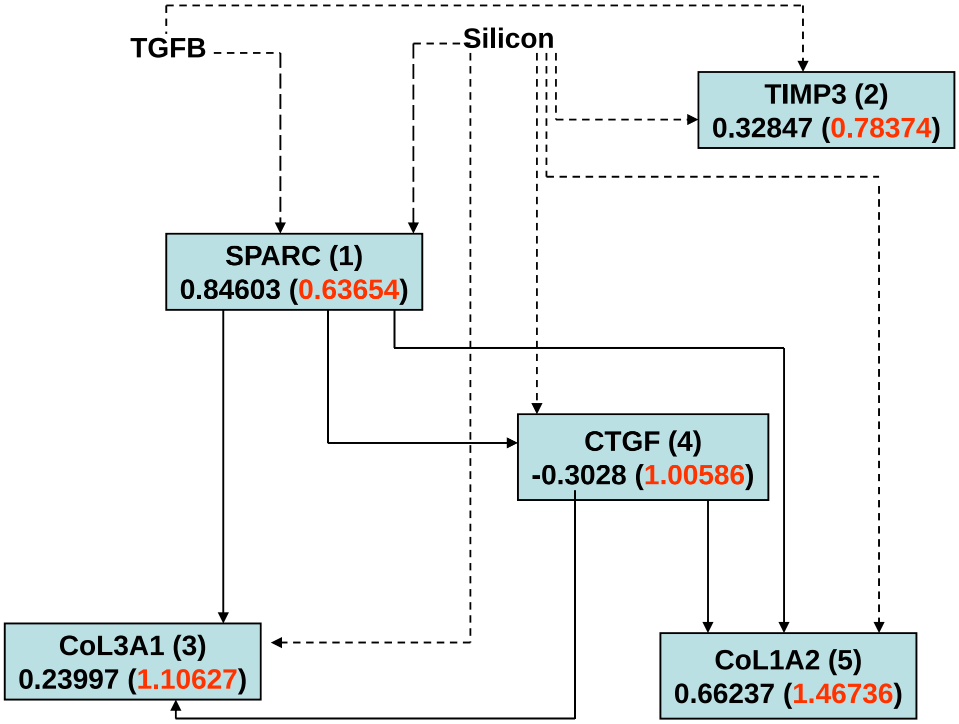


Fig. 17.: Scheme of TGFβ pathway

out and within parentheses, respectively. Let x_1, x_2, x_3, x_4, and x_5 be the expression levels of genes SPARC, TIMP3, COL3A1, CTGF and COL1A2, respectively. Let u_1 and u_2 be the expression of the TGFBR1 and silicon. The differential equations that were estimated from real time PCR experiments in SSc using my constrained EM algorithm and converted to a continuous system are given below,

$$\begin{aligned}
\frac{\mathrm{d}\,x_1}{\mathrm{d}\,t} &= -0.4518x_1 + 0.3123u_1 + 0.6091u_2 \\
\frac{\mathrm{d}\,x_2}{\mathrm{d}\,t} &= -0.2437x_2 + 0.0872u_1 + 0.1259u_2 \\
\frac{\mathrm{d}\,x_3}{\mathrm{d}\,t} &= -0.7101x_1 + 0.1013x_3 + 0.4181x_4 + 0.0937u_1 + 0.4218u_2 \\
\frac{\mathrm{d}\,x_4}{\mathrm{d}\,t} &= -0.0823x_1 + 0.0059x_4 + 0.0111u_1 + 0.0286u_2 \\
\frac{\mathrm{d}\,x_5}{\mathrm{d}\,t} &= -0.8163x_1 + 0.2368x_4 + 0.3835x_5 + 0.1033u_1 + 0.4180u_2
\end{aligned} \tag{5.22}$$

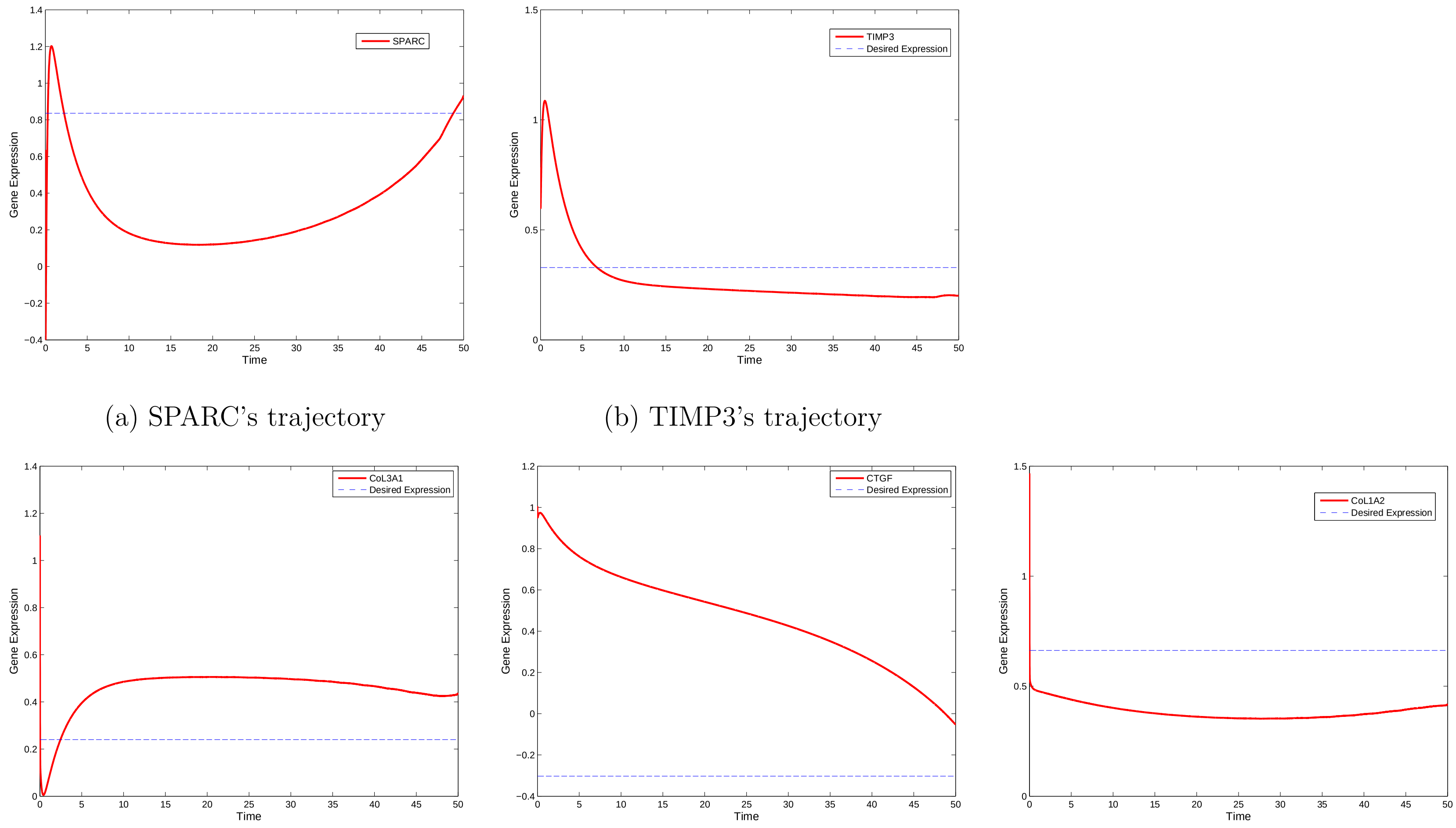
and all the gene expression levels are directly observable. The mean gene expressions in SSc fibroblasts are taken as the initial states of the differential equation (5.22) with values

$$x_0 = [0.6365,\ 0.7837,\ 1.1063,\ 1.0059,\ 1.4674]^{\mathrm{T}}.$$

The required states of the system is the mean expression levels of the genes in the TGFβ pathway in the normal fibroblasts and are given by

$$y_r = [0.8460\ 0.3285\ 0.2400\ \text{-}\ 0.3028\ 0.6624]^{\mathrm{T}}.$$

Both finite and infinite horizon linear-quadratic tracking controller (LQT) are used to determine the optimal expression level of the gene TGFβ and concentration of the silicon for reducing the expressions of the genes COL1A2 and COL3A1 to the desired normal levels. We know that in general, the infinite time-horizon LQT problem does not have a solution in the strict sense due to the possible divergence of the performance index. However, for applications, if the reference signal is generated by an asymptot-



(a) SPARC's trajectory

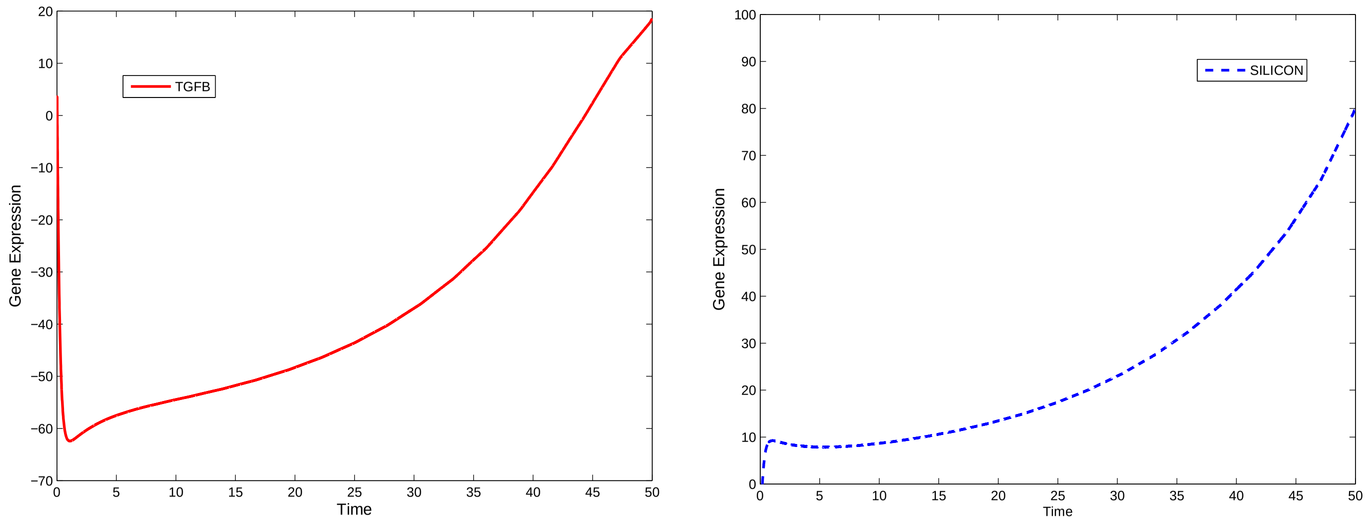
(b) TIMP3's trajectory

(c) COL3A1's trajectory

(d) CTGF's trajectory

(e) COL1A2's trajectory

Fig. 18.: Various trajectories of TGFβ pathway in finite time-horizon LQT. Although SPARC and CTGF show some large deviation from the reference values, they do end up being close at the end. Other genes show better behaviors by staying close for much of the time period when they are under control.



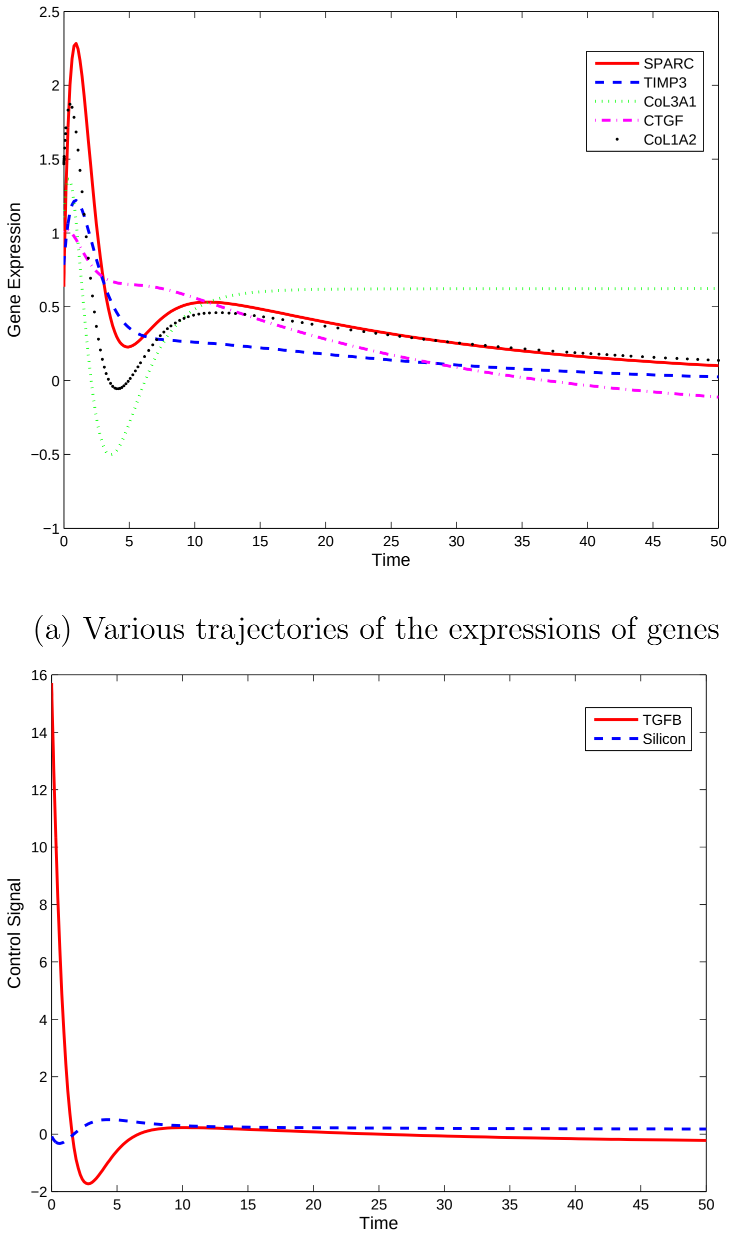
(a) TGFβ

(b) Silicon

Fig. 19.: Inputs for TFGβ pathway in finite time-horizon LQT.

ically stable system, in broad sense, the problem is well defined and approximately has an optimal solution [154]. Figures 18(a),18(b),18(c),18(d), and 18(e) show the expression levels of genes SPARC, TIMP3, COL3A1, CTGF and COL1A2 as a function of the time, which are the trajectories of the states in optimal LQT. Figures 19(a) and 19(b) show the trajectories of the control inputs, TGFβ expression and Silicon dosage. The final expression levels of the genes SPARC, TIMP3, COL3A1, CTGF and COL1A2 at the time point 50 are 0.9262, 0.2002, 0.4332, -0.0529, and 0.4144, respectively. The major cause of fibrosis in SSc is over-production of COL3A1 and COL1A2 from fibroblasts. Although genes COL3A1 and COL1A2 do not reach their desired values 0.2400 and 0.6624 (the mean expression levels of the genes COL3A1 and COL1A2), they are much reduced from their over-expressed values 1.1063 and 1.4674 in SSc.

Next we consider the infinite horizon LQT controller for TGFβ pathway. Figures 20(a) and 20(b) plot the trajectories of five genes SPARC, TIMP3, COL3A1, CTGF and COL1A2 and control signals. LQT controller aims to reduce the expression levels of the genes COL3A1 and COL1A2. To achieve this goal, at the beginning we see oscillation of the expressions of the five genes in the network. As control process proceeds, the expressions of all five genes are steadily reduced. The expression of gene COL3A1 converges to value 0.6225, which is higher than the reference value 0.2400, but still much lower than the initial, undesired high value 1.1063. After it oscillated for a while, the expression of the gene COL1A2 decreases to the value 0.1370, which is much lower than the initial high value of 1.4674. Changes of three other genes SPARC, TIMP3, CTGF follow similar patterns as gene COL1A2. It is noted that optimal LQT can also improve the dynamic properties of the TGFβ pathway. Table IX shows the eigenvalues of the original system and the new system of infinite time-horizon LQT. Transition matrix A for the SSc has three positive eigenvalues, but real



(a) Various trajectories of the expressions of genes

(b) Inputs

Fig. 20.: Trajectories and inputs of TGFβ pathway in infinite time-horizon LQT. All the genes show improvements over their over-expressed values, although no one is able to reach reference values and stay there. The inputs go toward zeros as time goes on, which is good because we do not want non-zero inputs for infinite horizon control problems.

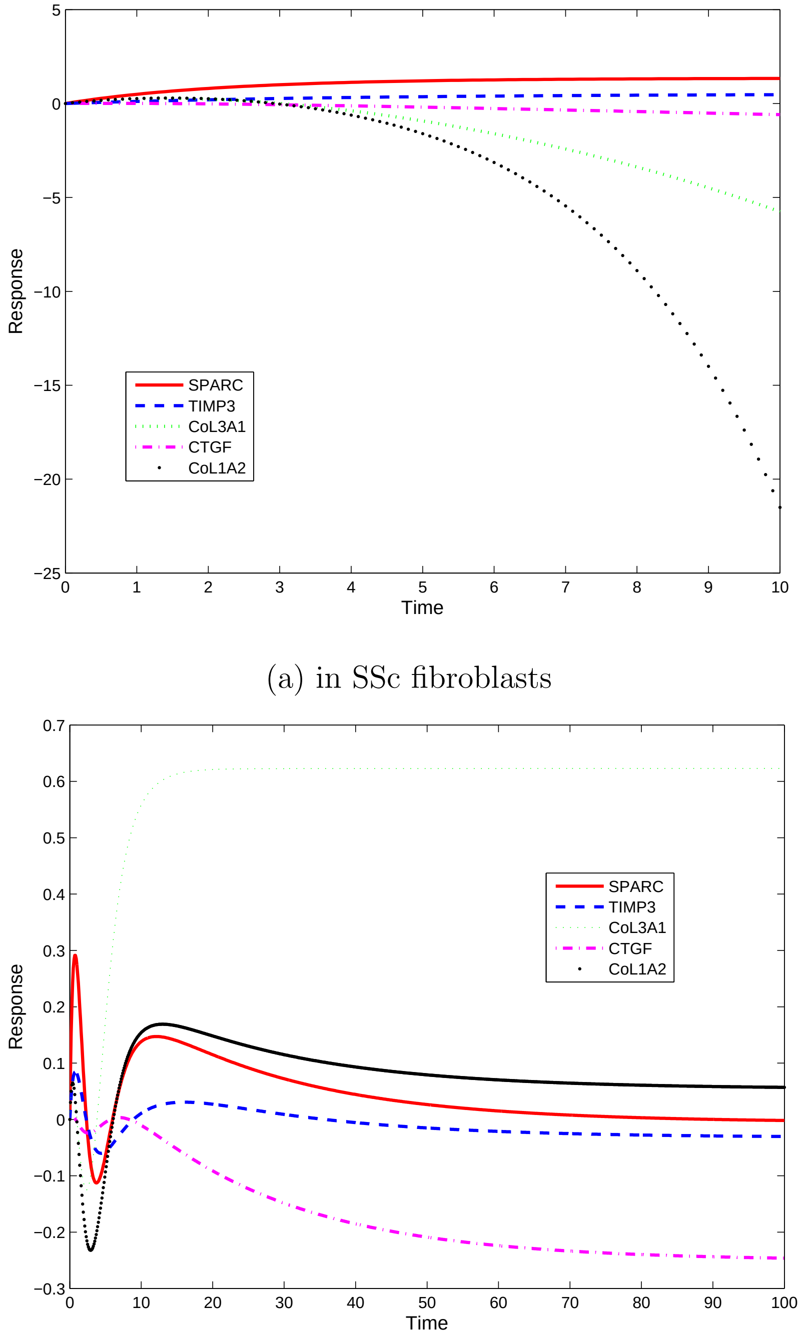
Table IX.: Eigenvalues of matrix A of the state-space model for the TGFβ pathway for the original SSc fibroblasts and in infinite time-horizon LQT.

SSc fibroblasts	Infinite Horizon LQT
-0.4518	-0.7746 + 0.4977i
0.3835	-0.7746 - 0.4977i
-0.2437	-0.3835
0.1013	-0.2437
0.0059	-0.0447

parts of all eigenvalues of the same matrix for in the infinite time-horizon LQT are negative. This demonstrates the stability of TGFβ pathway in infinite time-horizon LQT is much improved. Figures 21(a) and 21(b) show the step response curves of the TGFβ pathway to perturbation of silicon in the SSc fibroblast and to synthesized control signal g in the infinite horizon LQT. We can see from Figures 21(a) and 21(b) that the expressions of all five genes after perturbation of external signal in LQT quickly reach the steady states, but the expressions of COL1A2 and COL3A1 in the SSc fibroblasts after perturbation of silicon were unstable and will never reach the steady-state values, thus LQT improves the stability of the overall system.

2. Glutathione Redox Cycle in Mice Lung Cells Exposed to Carbonyl Chloride

The Glutathione (GSH) redox cycle in mice lung cells exposed to phosgene (another name for Carbonyl Chloride) involves six genes ALD2A1,GPX4, GCLC, GCLM, GSS and IDH2 as shown in Figure 22. A state-space model is used to model the response of the GSH redox cycle in mice lung cells to the perturbation of phosgene. Two genes ALD2A1 and GPX4 are modeled as inputs u_1 and u_2, respectively. All four other



(a) in SSc fibroblasts

(b) pathway in infinite time-horizon LQT

Fig. 21.: Step response of TGFβ pathway to perturbation of silicon. The pathway shows improvements in stability for the overall system after the pathway is placed under LQT controller. The top panel is the pathway without any controller, showing unstable step responses, while the bottom panel is the step responses of the overall system, showing stable step responses.

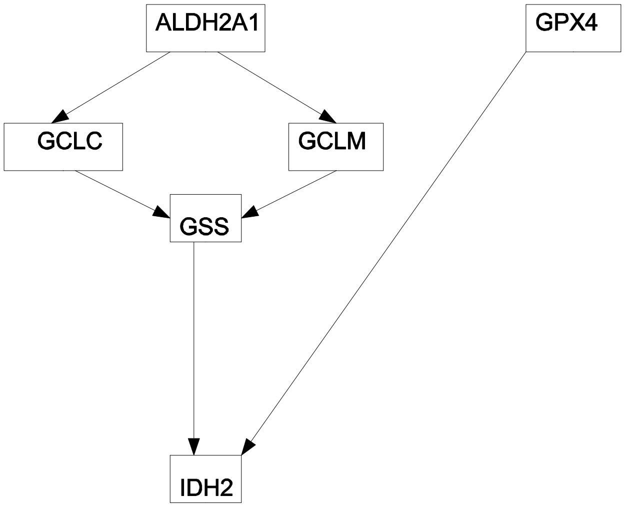
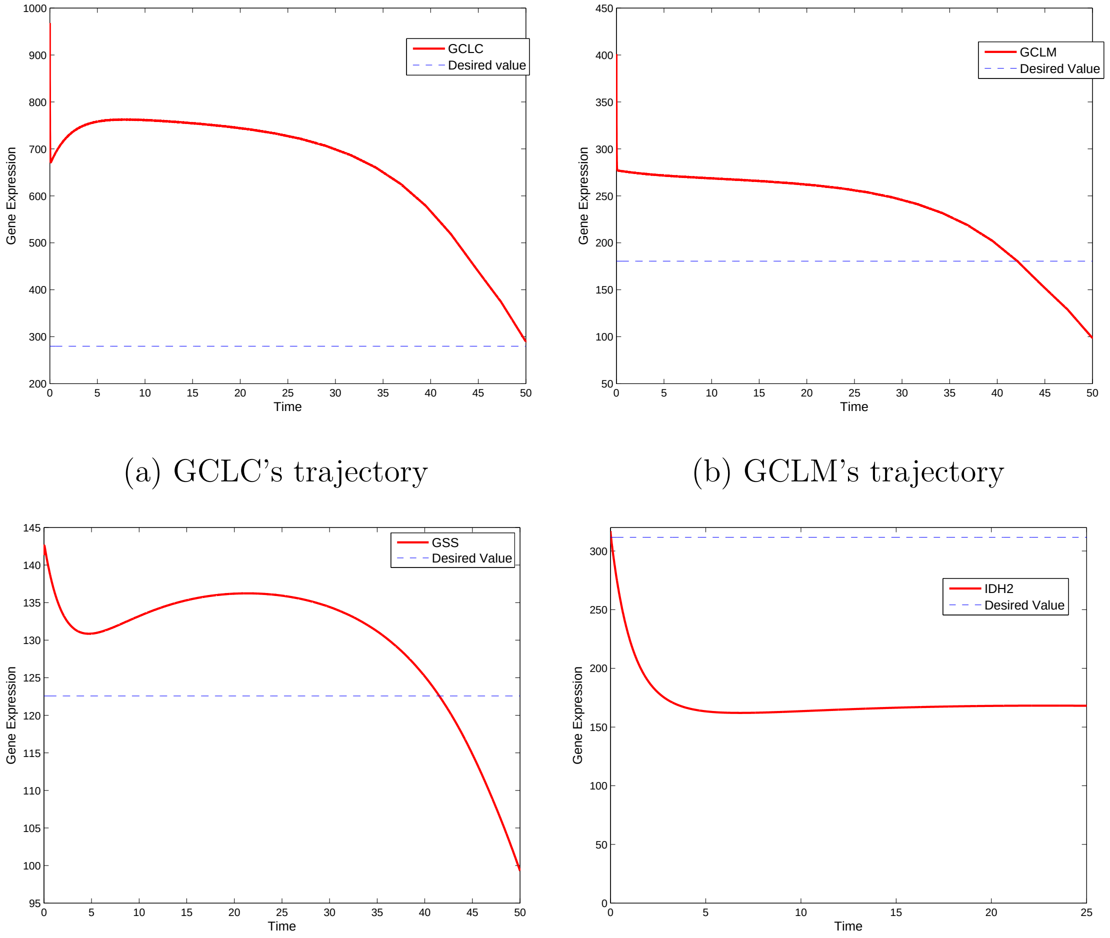


Fig. 22.: The scheme of GSH redox cycle.

genes are taken as the states of the system and are modeled by state-space equations. Here, gene GCLC is denoted by x_1 , gene GCLM x_2 , gene GSS x_3 , and gene IDH2 x_4. The observed expressions of the genes GCLC, GCLM, GSS and IDH2 are denoted by y_1, y_2, y_3 and y_4. I used EM algorithms to estimate the parameters of the model. The continuous model describing GSH redox cycle in mice lung cells exposed to phosgene, as estimated by the constrained EM algorithm, is given by

$$\begin{aligned}
&\frac{\mathrm{d}\,x_1}{\mathrm{d}\,t} = -0.4385x_1 + 4.5356u_1\\
&\frac{\mathrm{d}\,x_2}{\mathrm{d}\,t} = -0.5082x_2 + 1.8522u_1\\
&\frac{\mathrm{d}\,x_3}{\mathrm{d}\,t} = 0.0672x_1 - 0.1358x_2 - 0.092x_3 - 0.0259u_1\\
&\frac{\mathrm{d}\,x_4}{\mathrm{d}\,t} = -0.0575x_1 + 0.1178x_2 + 1.3707x_3 - 1.0518x_4 + 0.0294u_1 + 0.1437u_2\\
&y_1 = x_1, y_2 = x_2, y_3 = x_3, y_4 = x_4.
\end{aligned} \tag{5.23}$$

The initial values are the mean expressions of genes GCLC, GCLM, GSS and IDH2 (1615.7, 341.1, 124.7, and 287.9) when the GSH redox cycle in lung cells exposed to phosgene. The desired nominal values are their expressions (279.763, 180.5685, 122.5685, and 311.65) when lung cells are exposed to air. The aim of LQT is to make the expressions of the genes GCLC, GCLM, GSS and IDH2 in the GSH redox cycle in mice lung cells exposed to phosgene track as closely as possible their corresponding values in mice lung cells exposed to air. The control signals are the expressions of genes ALD2A1 and GPX4. The optimal expressions of gene GPX4 in the unconstrained LQT are always a negative function of the time, so the expressions of the gene GPX4 is represed to zero in the below analysis. Figures 23(a), 23(b), 23(c),and 23(d) show optimal expressions of the genes GCLC, GCLM, GSS and IDH2 in finite horizon LQT. Observe that the expressions of GCLC and GCLM almost monotonically decrease and the expression of the gene GCLC all run higher than the desired expression, but at the end of the time period, they are very close to the desired nominal values. The expression of the gene GCLM crossed the desired nominal line before the end of the period and reachs the value close to 100. The trend of the expression of the gene GSS is that it decreases although there is a short period of time to increase its expression. The expression of the gene IDH2 quickly reach its stable value, but all run much lower than the desired nominal value due to small impact of the control signal ALDHA1 on the expression of the gene IDH2. Figure 24 plots the curve of the control signal. The optimal way to change the expressions of the four genes from undesired values to desired values calls for extremely high expression of the gene ALDHA1. This demonstrates the difficulty of optimal LQT for repairing damages of GSH redox cycle in mice lung cells exposed to poisonous phosgene and may be related to the relative high value of the condition number of the controllability matrix of the system noted in the previous chapter. Figure 25(a) plots the trajectories of the expressions of genes GCLC, GCLM, GSS, and IDH2 in infinite horizon LQT. At the beginning the expressions of all four genes rapidly decreased. Then, they quickly reach the values



(a) GCLC's trajectory

(b) GCLM's trajectory

(c) GSS's trajectory

(d) IDH2's trajectory

Fig. 23.: Various trajectories of GSH redox cycle in finite time-horizon LQT.

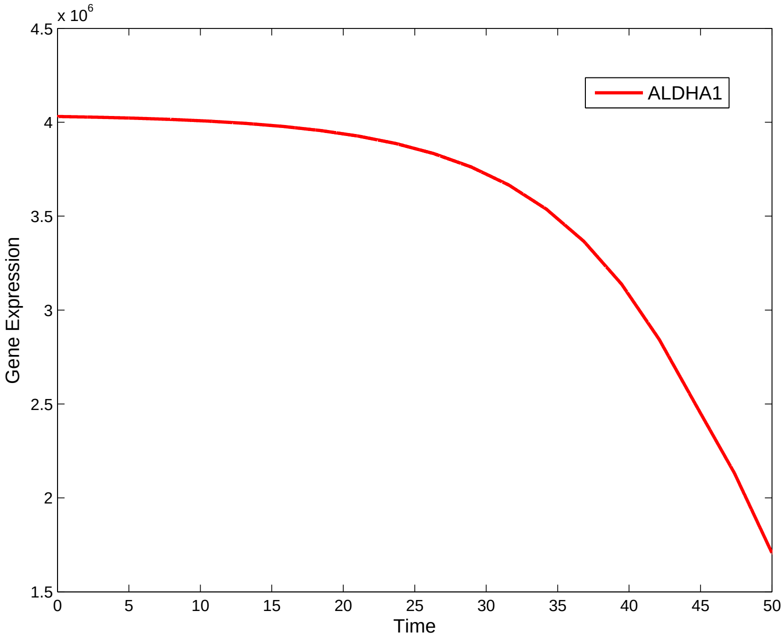
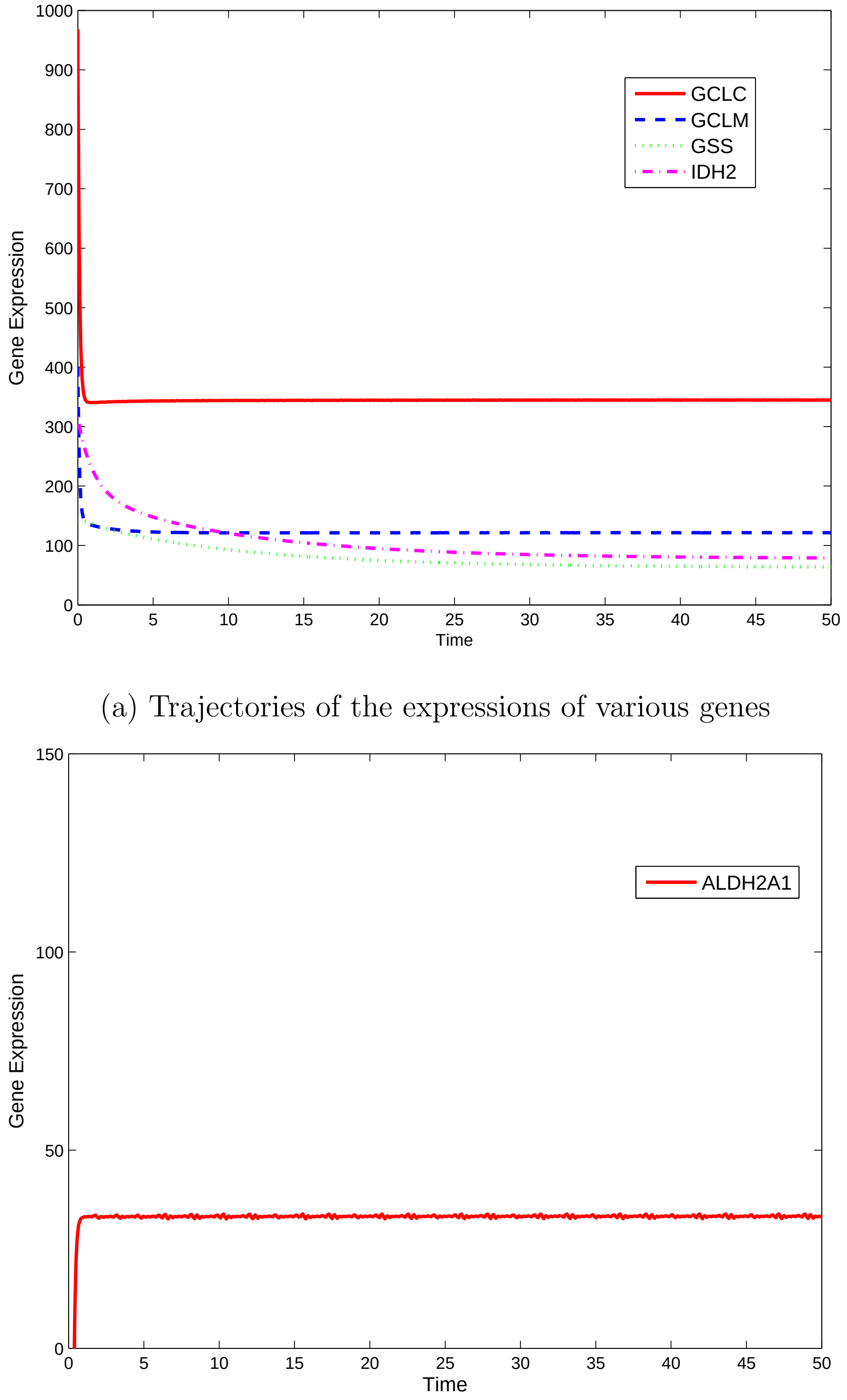


Fig. 24.: Input (gene ALDHA1) in finite time-horizon LQT.

344.6551, 121.4521, 63.8734, and 79.0345 and then stay there. Figure 25(b) shows the control input (expression of the gene ALDH2A1) in infinite horizon LQT. It starts with the rapid increase of the gene ALDH2A1 to force the reduction of genes GCLC and GCLM' expression levels. When the expression of gene ALDH2A1 reached value 33.2740 it stays there to maintain the expressions of the four controlled genes around their stable values.

3. LQT for MAPK Pathway in Cell Lines Disturbed by Wild Type and Mutant Type R80A HIV Type I Vpr Protein

HIV-1 Vpr protein is an important protein in promoting the pathogenesis of AIDS by facilitating apoptosis and cell cycle stall at G2. HIV-1 Vpr protein influences the expressions of the genes in the MAPK pathway. Yoshizuka and colleagues [110] observed that different genotypes of the gene HIV-1 Vpr caused differential expressions of the



(a) Trajectories of the expressions of various genes

(b) Inputs (expression of gene ALDH2A1)

Fig. 25.: Trajectories and inputs of GSH redox cycle in infinite time-horizon LQT.

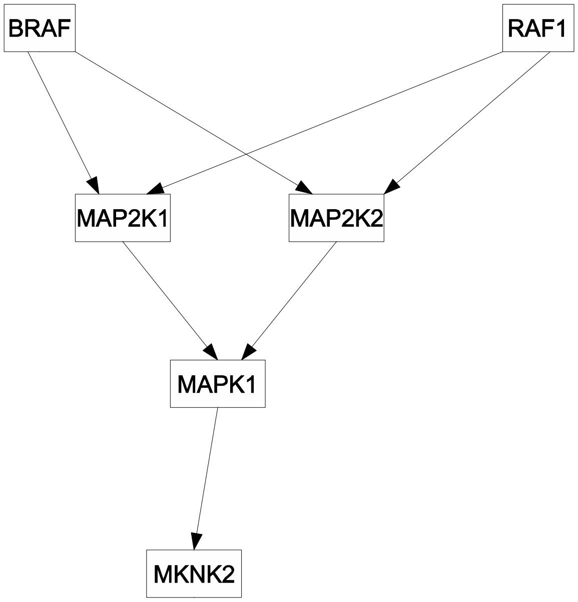


Fig. 26.: The scheme of MAPK pathway.

genes in the Mitogen-activated Protein Kinase (MAPK) pathway. MAPK pathway is shown in Figure 26. We study whether we can change the expressions of the genes in the MAPK pathway in cell lines expressing HIV-1 Vpr mutant types to their values in cell lines expressing wild type HIV-1 Vpr by LQT. The gene BRAF and RAF1 are taken as inputs and denoted by u_1 and u_2, respectively. Let x_1, x_2, x_3, x_4 and y_1, y_2, y_3, y_4 denote the expression levels of genes MAP2K1, MAP2K2, MAPK1, MKNK2 and their observed expressions, respectively. Let $x = [x_1, x_2, x_3, x_4]^T$, $u = [u_1, u_2]^T$ and $y = [y_1, y_2, y_3, y_4]$. I used EM algorithm to fit the data from the experiments performed by Yoshizuka et al. [110]. The resulting linear state-space model for MAPK pathway in the cell lines expressing HIV-1 Vpr mutant type is given by

$$\begin{aligned}
\frac{\mathrm{d}\, x_1}{\mathrm{d}\, t} &= -1.4159x_1 - 0.1756u_1 - 0.1279u_2 \\
\frac{\mathrm{d}\, x_2}{\mathrm{d}\, t} &= -0.6505x_2 - 0.3127u_1 + 0.0876u_2 \\
\frac{\mathrm{d}\, x_3}{\mathrm{d}\, t} &= -0.4009x_1 + 0.0534x_2 - 0.2494x_3 - 0.0168u_1 - 0.0194u_2 \\
\frac{\mathrm{d}\, x_4}{\mathrm{d}\, t} &= -0.0006x_1 + 0.0001x_2 - 0.0025x_3 - 0.0974x_4 \\
y_1 &= x_1, y_2 = x_2, y_3 = x_3, y_4 = x_4.
\end{aligned} \tag{5.24}$$

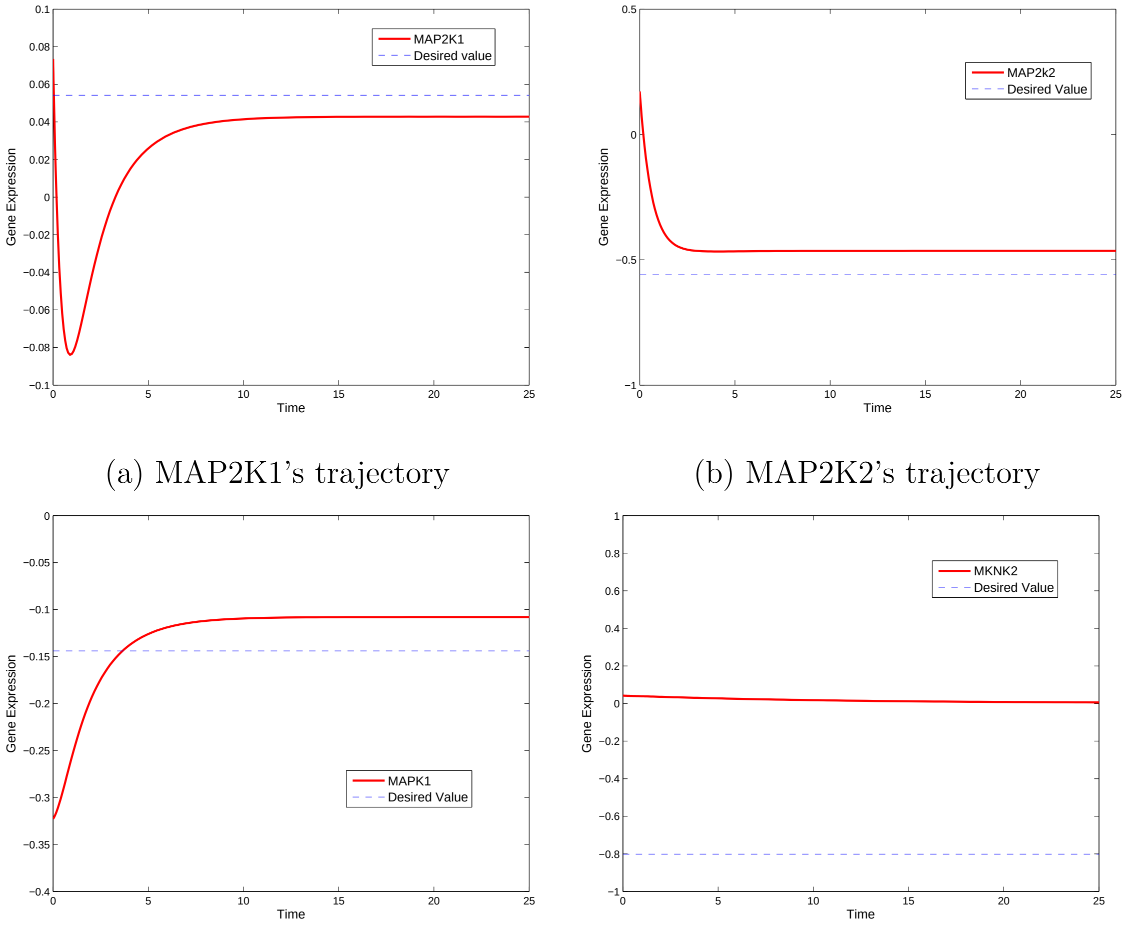
The initial conditions of the state variables are the mean expression values of genes MAP2K1, MAP2K2, MAPK1, MKNK2 in the cell lines expressing HIV-1 Vpr mutant types and are given by

$$x_0 = [0.0735, 0.1719, -0.3227, 0.0419]^T.$$

The desired nominal values are the mean expression of the genes MAP2K1, MAP2K2, MAPK1, MKNK2 in the in cell lines expressing HIV-1 Vpr wild types and are as follows:

$$y_r = [0.0542, -0.5601, -0.1440, -0.8013]^T.$$

The goal of LQT is to change the expressions of genes MAP2K1, MAP2K2, MAPK1, MKNK2. Both finite and infinite horizon LQT analyses were performed. Figures 27(a), 27(b), 27(c), and 27(d) plot the trajectories of genes MAP2K1, MAP2K2, MAPK1, and MKNK2 in finite time-horizon LQT. At the beginning the expression of the gene MAP2K1 rapidly decreases from the initial value 0.0735 to -0.08 then increases and converges to value 0.0427, which is less than the desired value 0.0542. Starting to decrease from the initial value 0.1719, the expression of gene MAP2K2 rapidly converges to the value -0.4648, which is higher than the desired value -0.5601 and then stayed there during the remaining time of the planning period. The expres-



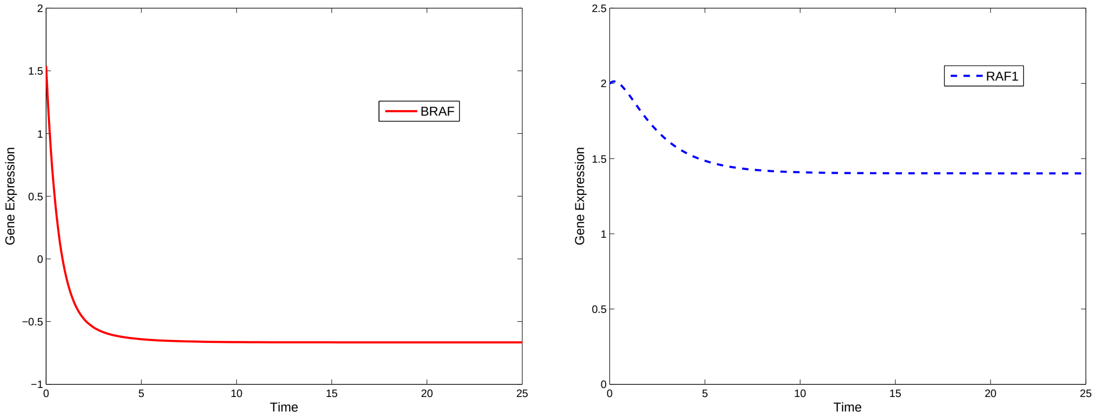
(a) MAP2K1's trajectory

(b) MAP2K2's trajectory

(c) MAPK1's trajectory

(d) MKNK2's trajectory

Fig. 27.: Trajectories of genes in MAPK pathway in finite time-horizon LQT.



(a) BRAF's trajectory

(b) RAF1's trajectory

Fig. 28.: Trajectories of two inputs of the MAPK pathway in finite time-horizon LQT.

sion of gene MAPK1 monotonically increases from -0.3227 to -0.1079, which is larger than the desired value -0.1440. The expression of gene MKNK2 converged to 0.0057, which is much larger than the desired value, -0.80131. From Figure 27(d) we can observe that the expression of gene MKNK2 is almost constant during the planning period. Gene MKNK2 is farther removed than other genes in the MAPK pathway, and the linear state-space model (5.24) shows that regulatory effect of gene MAPK1 on MKNK2 is very small. Therefore, changing the expressions of the genes BRAF and RAF1 does not have much impact on MNKK2 and hence it is difficult to regulate the expression of gene MKNK2 by control signals BRAF and RAF1. Figures 28(a) and 28(b) plot the control signals (expressions of the genes BRAF and RAF1) in the finite time-horizon LQT. Figures 29 and 30 show the trajectories of the expressions of MAP2K1, MAP2K2, MAPK1, and MKNK2 and control signals in infinite time-horizon LQT of MAPK pathway. The expressions of MAP2K1, MAP2K2, MAPK1 and MKNK2 quickly converge to values 0.0543, -0.4329, -0.1127, and 0.0029. The expressions of genes MAP2K1, MAP2K2, MAPK1 are moved close to the desired values,

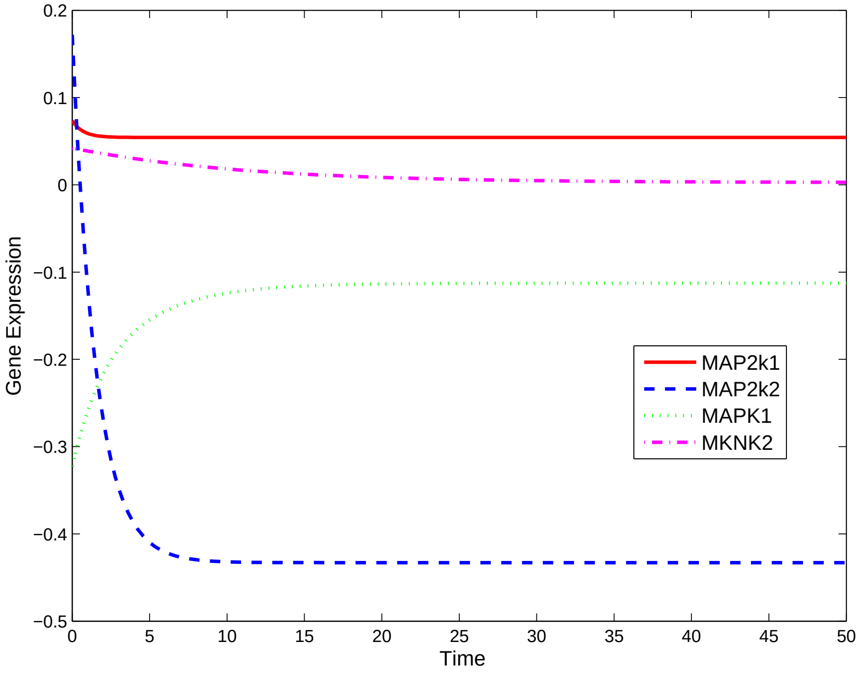


Fig. 29.: Trajectories of genes MAP2K1, MAP2K2, MAPK1, and MKNK2 of MAPK pathway in infinite time-horizon LQT.

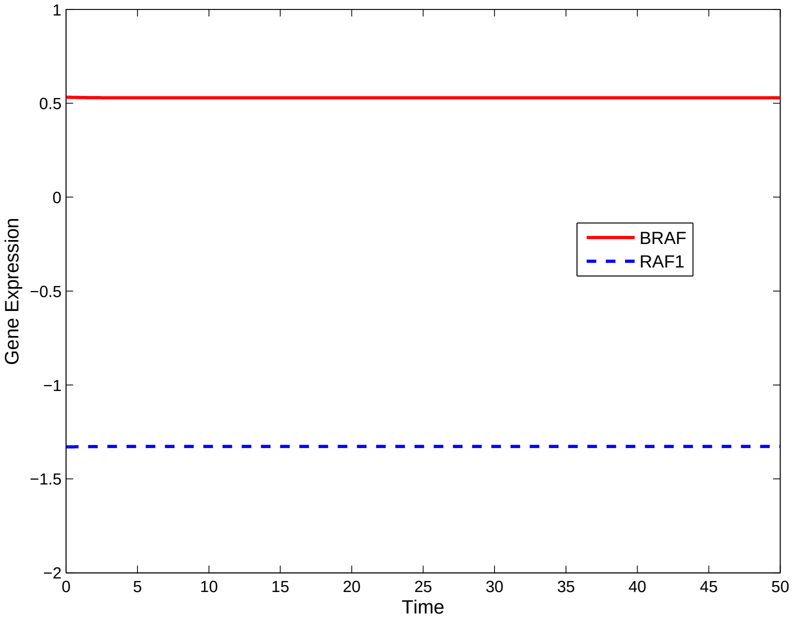


Fig. 30.: Inputs (genes BRAF and RAF1) of MAPK pathway in infinite time-horizon LQT.

0.0542, -0.5601, and -0.1440, but the expression of the gene MKNK2 is far away from the desired value, -0.8013. This again demonstrates the difficulty in regulating the downstream genes in the pathway by LQT.

D. Discussion

In this chapter, regulating gene expression levels in genetic regulatory and signal transduction networks using finite and infinite time-horizon LQT was formulated. To date, only Boolean-network-based controllers have been proposed to regulate the gene expressions. But as Boolean networks gloss considerable biological details, linear state-space models should also be studied for the purpose of regulating gene expressions. Although LQT has been widely used in engineering, to ensure its successful application to gene regulation, I have addressed several important issues that are different from its applications in engineering through adapting of the finite and infinite horizon LQT to regulate TGFβ, GSH, and MAPK pathways.

The first issue is the choice of target trajectory. The purpose of LQT for genetic networks in the abnormal cells is to drive diseased genetic networks to go from undesirable states to desirable states. The undesirable states are, in general, expressions of the genes in abnormal cells. These genes in the abnormal cells are either over-expressed or under-expressed when compared with those in the normal cells. In other words, the genes are differentially expressed. The observed gene expressions are the mean values of the gene expressions in a large number of cells in microarrays. In addition, due to constant perturbations of internal and external environments, gene expression levels are stochastically changed in both space and time. Therefore, the mean expression levels in the abnormal cells is a good value as the initial values of the differential equations in LQT and mean expression levels in the normal cells is a good value as the reference trajectory. This is also related to the precision of control in biological networks. It is not necessary, nor practical, to precisely control the expression levels to be equal to the target values. It is more practical to drive the expression levels to be close to the reference values. Applications of the LQT to three real experimental datasets show that it is difficult to precisely reach the desired levels, but that it is feasible to be close.

Because the genetic regulatory and signal transduction networks are complex, there may be very indirect paths from the control signals to the target genes. The examples of GSH and MAPK pathway indicate that it is difficult to control the expressions of the far away genes in the pathway. In constrast, the example of TGFβ pathway demonstates that drugs may simultaneously target several genes in the networks, including distant genes, and to much better result. In other words, we can use LQT to control the expressions of directly controlled genes to desired levels, but it can be much difficult on indirectly controlled genes. This observation has important implication in practice, for the fact that remote control of target genes is likely to be ineffective for treatments means potential treatment methods such as gene therapy, small RNA interference, and antisense treatment must directly target diease-causing genes in addition to taking into account the response of the networks on a system level.

My results are still preliminary. In this dissertation, constrained optimal control and nonlinear control method have not been developed. In practice, the constraints should be imposed on the states and the control signals. Development of constrained optimal control method will have great implications in both theory and practice. In addition, I have also not considered the impact of stochastic forces in the gene regulatory and signal transduction networks. In the future, application of robust control which takes system and measurement noise into account in gene regulation should be investigated.

CHAPTER VI

PARAMETER ESTIMATION OF NONLINEAR STATE-SPACE MODELS

A. Introduction

So far I have used linear state-space dynamic models to study genetic regulatory networks. Linear models are popular in studying genetic regulatory networks [69, 74, 61, 72, 155] because of its analytic tractability and because a lot of details of gene regulation are still murky and need to be glossed over, but there are attempts at nonlinear modeling of gene regulations [57, 27] that tend to incorporate details of transcriptional factor binding, which can be approximated by nonlinear Hill equations [156]. But in truth, nonlinear state-space modeling is much more common in signal transduction network studies [10, 96, 76, 10, 11, 28], where a bottom-up, mechanistic modeling effort often results in nonlinear ordinary differential equations. Metabolic networks are often studied under a linear approximation scheme called flux-balance analysis [157, 158, 159], although nonlinear dynamic modeling is also possible [160, 161, 162]. A nonlinear approximation approach called S-systems has also been used to study various biological networks [163, 105, 106]. Nonlinear models should be closer to the underlying biochemical reactions because biochemical reactions are nonlinear, and as we collect more data and gain more insights into the various reactions in the cell, there will be more opportunities and more needs for nonlinear modeling.

But no matter whether the model is linear or nonlinear, there are still two key aspects of biological networks, parameter values and structure, that need to be known before the model can be of use. Since nonlinear state-space dynamic modeling tends to be deployed in detailed, mechanistic studies about well studied cellular systems, structures are usually assumed known; Nachman et al. [57] is an exception where they estimated both parameters and structure. This chapter also assume structures are known and that only parameters need to be estimated.

In this chapter, I will study the problem of parameter estimation for nonlinear state-space dynamic systems with additive, white Gaussian noise. I choose this class of systems because state-space systems offer great flexibility in the details of models and approximation of biochemical reactions often lead to state-space forms [67, 70, 164, 66, 69]. The additive, white Gaussian noise is not general as arbitrary noises in arbitrary functional form [165], but it is more tractable analytically, and Gaussian sums can be used to approximate non-Gaussian noises [166].

State-space dynamic systems have hidden variables that cannot be directly measured. This facilitates our modeling efforts because many important quantities about biochemical elements cannot be directly measured or observed, particularly within cells. This is the reason I choose state-space systems. But the presence of hidden states also complicates parameter estimation because hidden states also need to be estimated in some way. The expectation-maximization algorithm (EM) divides estimation in two steps: estimate states using the current estimate of parameters, and estimate parameters using the current estimate of states. I have already used EM algorithm to estimate parameters of linear systems in chapter III; now I will try to do the same for nonlinear systems.

EM algorithm has two steps, the E-step, estimation of hidden states, and the M-step, parameter estimation. These two steps are recursively and repeatedly executed until a local maximum is found. There are a number of choices in state estimation. Extended Kalman filter [167, 80] is an extension of linear Kalman filter by linearizing states around the current best estimate of states. It has the advantage of being simple but the disadvantage of sometimes worse performance than other methods. In this chapter, state estimation is by extended Kalman filter (EKF), because EKF is the the most popular and also the simplest nonlinear extension of Kalman filter, a simplicity that I will leverage when I derive an EM algorithm for systems with time-delays. Unscented Kalman filter (UKF) is another extension of regular Kalman filter. UKF approximates nonlinear state transition by a set of deterministically chosen points, called sigma points [168]. Particle filter on the other hand is a sequential Monte Carlo method that tries to approximate states by repeated sampling [169]. Particle filter tends to be computationally expensive but reportedly it is most accurate of the three nonlinear filters [170]. There is already work on parameter estimation using EM algorithm and particle filter as the E-step [165].

The M-step of nonlinear EM algorithm is very different from the M-step for linear systems. There is no longer the possibility of analytic solution because there is no analytic solution to arbitrary nonlinear equations. Instead, numerical nonlinear optimization methods such as gradient descent or Gauss-Newton's method need to be used to find local maximum. This is the approach Schön et al. [165] took, and the one I will take. Particle filter can approximate nonlinear expectations so Schön et al. used it to compute the gradient directly. EKF cannot do that. Instead, I have to take the first order Taylor approximation of the gradient, just as EKF takes the first order Taylor approximation of nonlinear state transition and output equations. The result is a set of equations that will be detailed in the next section. This is different from the first approach I tried, linearization of state transition and output equations, which resulted in a very complicated set of equations involving multiple second order derivatives.

Because ordinary-differential-equation model of biochemical reactions are actually an approximation of partial differential equations by neglecting to consider distance and pretending that everything is well mixed or at least at the same place, there are often time-delays needed to compensate for the fact that molecules are actually compartmentalized in cells and need to be transported to their destinations. Short of resorting to partial differential equations where transport processes can be explicitly modeled, we can introduce time-delays into ODEs as approximation. Parameter estimation of nonlinear state-space dynamic model appears to be poorly addressed in the literature, particularly regarding its application to biological systems. Quach et al. [168] approximated time-delay with an indicator function in their state-space model and Kim et al. [54] had time-delays but no hidden states.

B. Method

The main complication of nonlinear systems for EM is that the M-step has no analytical solution but must resort to gradient-based approach. We already noted in appendix B that EM can be broadly viewed as alternating improvement of log-likelihood function by holding hidden states and then parameters constant; this is the view that justifies the gradient-based approach.

Suppose we have state at $k+1$ as $\mathbf{x}_{k+1}$ and observation at time k as $\mathbf{y}_k$ evolving nonlinearly as

$$\begin{aligned} \mathbf{x}_{k+1} &= f_\theta(\mathbf{x}_k, u_k) + \mathbf{w}_k \\ \mathbf{y}_k &= h_\varphi(\mathbf{x}_k, u_k) + \mathbf{v}_k, \end{aligned} \tag{6.1}$$

where noise terms $\mathbf{w}_k$ and $\mathbf{v}_k$ are assumed to white Gaussian with zero means and variance Q and R for all time, respectively; and the nonlinear function $f_\theta(\cdot)$ and $h_\varphi(\cdot)$ depend on parameter vectors θ and φ, respectively. We need to estimate θ and φ.

The complete-data probability is

$$\begin{aligned}
\mathrm{P}(\mathbf{X_{N+1}}, \mathbf{Y_N}) &= \mathrm{P}(\mathbf{y}_N|\mathbf{X}_{N+1}, \mathbf{Y}_{N-1})\, \mathrm{P}(\mathbf{X}_{N+1}, \mathbf{Y}_{N-1}) \\
&= \mathrm{P}(\mathbf{y}_N|\mathbf{x}_N)\, \mathrm{P}(\mathbf{x}_{N+1}|\mathbf{X}_N, \mathbf{Y}_{N-1})\, \mathrm{P}(\mathbf{X}_N \mathbf{Y}_{N-1}) \\
&= \mathrm{P}(\mathbf{y}_N|\mathbf{x}_N)\, \mathrm{P}(\mathbf{x}_{N+1}|\mathbf{x}_N)\, \mathrm{P}(\mathbf{X}_{N-1} \mathbf{Y}_{N-1}) \\
&\vdots \\
&= \mathrm{P}(\mathbf{x}_1) \prod_{k=1}^{N} \mathrm{P}(\mathbf{y}_k|\mathbf{x}_k) \prod_{k=1}^{N} \mathrm{P}(\mathbf{x}_{k+1}|\mathbf{x}_k).
\end{aligned} \tag{6.2}$$

Taking logarithm of equation (6.2) to have

$$\log \mathrm{P}(\mathbf{X_{N+1}}, \mathbf{Y_N}) = \log \mathrm{P}(\mathbf{x}_1) + \sum_{k=1}^{N} \log \mathrm{P}(\mathbf{y}_k|\mathbf{x}_k) + \sum_{k=1}^{N} \log \mathrm{P}(\mathbf{x}_{k+1}|\mathbf{x}_k), \tag{6.3}$$

and taking into account the Gaussian nature of noise and retaining only terms relating to the parameters or the states, from equation (6.3) we get

$$\begin{aligned}
&\log\det P_1 + (\mathbf{x}_1 - \mu_1)^T P_1^{-1} (\mathbf{x}_1 - \mu_1) + \sum_{k=1}^{N} (\mathbf{y}_k - h_\varphi(\mathbf{x}_k, u_k))^T R^{-1} (\mathbf{y}_k - h_\varphi(\mathbf{x}_k, u_k)) \\
&+ N \log\det R + \sum_{k=1}^{N} (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))^T Q^{-1} (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k)) + N \log\det Q.
\end{aligned} \tag{6.4}$$

Parameters related to the initial state, μ_1 and P_1 (mean and variance, respectively, of the initial state), are in the same functional form as in linear systems, so the same method applies. But the rest of the parameters and all the states are nonlinear functions, so we cannot simply take expectation because there is no general solution for expectation of nonlinear functions. We could take first-order Taylor expansion of linear equation (6.1) and proceed to estimate parameters in the linearized system, which I did, but the resulting gradient is a very messy set of equations. So instead, I decided to linearize the gradient of the conditional expectation in the E-step, which is the same as taking derivative under the integral in

$$\mathrm{E}_{\mathbf{x}|\mathbf{y}}[\log \mathrm{P}(\mathbf{x},\mathbf{y})] = \int_{\mathbf{x}} \mathrm{P}(\mathbf{x}|\mathbf{y}) \log(\mathrm{P}(\mathbf{y},\mathbf{x})) \mathrm{d}\, \mathbf{x}.$$

Let us make those definitions:

$$l_k^1 = (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))^T Q^{-1} (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k)) \tag{6.5}$$

$$l_k^2 = (\mathbf{y}_k - h_\varphi(\mathbf{x}_k, u_k))^T R^{-1} (\mathbf{y}_k - h_\varphi(\mathbf{x}_k, u_k)), \tag{6.6}$$

then their respective gradients are

$$G_1(\mathbf{x}_k, u_k) = \frac{\partial l_k^1}{\partial \theta} = -2 \frac{\partial f_\theta^T}{\partial \theta} Q^{-1} (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k)) \tag{6.7}$$

$$G_2(\mathbf{x}_k, u_k) = \frac{\partial l_k^2}{\partial \varphi} = -2 \frac{\partial h^T}{\partial \varphi} R^{-1} (\mathbf{y}_k - h_\varphi(\mathbf{x}_k, u_k)). \tag{6.8}$$

Taking Taylor expansion of $G_1(\mathbf{x}_k, u_k)$ and $G_2(\mathbf{x}_k, u_k)$ with respect to $\hat{x}_{k|k}$ and $\hat{x}_{k|k-1}$ respectively, we have

$$G_1(\hat{x}_{k|k}, u_k) + \left.\frac{\partial G_1(\mathbf{x}_k, u_k)}{\partial \mathbf{x}_k^T}\right|_{\mathbf{x}_k = \hat{x}_{k|k}} (\mathbf{x}_k - \hat{x}_{k|k}) \tag{6.9}$$

$$G_2(\hat{x}_{k|k-1}, u_k) + \left.\frac{\partial G_2(\mathbf{x}_k, u_k)}{\partial \mathbf{x}_k^T}\right|_{\mathbf{x}_k = \hat{x}_{k|k-1}} (\mathbf{x}_k - \hat{x}_{k|k-1}). \tag{6.10}$$

Note that

$$\frac{\partial G_1(\mathbf{x}_k, u_k)}{\partial \mathbf{x}_k^T} = \frac{\partial^2 l_k^1}{\partial \theta \partial \mathbf{x}_k^T}.$$

This is not a Hessian matrix but it is a matrix made up of second-order derivatives. Following the steps outlined in appendix A, we will take the Jacobian of the gradient.

Taking the differential first and we get

$$\begin{aligned}
&\mathrm{d}\,\frac{\partial f_\theta^T}{\partial \theta} Q^{-1}(\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k)) \\
&= \frac{\partial f_\theta^T}{\partial \theta} Q^{-1}\mathrm{d}\,(\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k)) + \mathrm{d}\,\left[\frac{\partial f_\theta^T}{\partial \theta}\right] Q^{-1}(\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k)) \\
&= -\frac{\partial f_\theta^T}{\partial \theta} Q^{-1}\mathrm{d}\, f_\theta(\mathbf{x}_k, u_k) + \mathrm{d}\ \mathrm{vec}\left[\frac{\partial f_\theta^T}{\partial \theta} Q^{-1}(\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))\right]
\end{aligned}$$

because $\left[\frac{\partial f_\theta^T}{\partial \theta} Q^{-1}(\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))\right]$ is a row vector, $\mathrm{vec}\left[\frac{\partial f_\theta^T}{\partial \theta} Q^{-1}(\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))\right] = \mathrm{vec}\left[(\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))^T Q^{-1}\frac{\partial f_\theta}{\partial \theta^T}\right]$, so the above is

$$\begin{aligned}
&= -\frac{\partial f_\theta^T}{\partial \theta} Q^{-1}\mathrm{d}\, f_\theta(\mathbf{x}_k, u_k) + \mathrm{d}\ \mathrm{vec}\left[(\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))^T Q^{-1}\frac{\partial f_\theta}{\partial \theta^T}\right] \\
&= -\frac{\partial f_\theta^T}{\partial \theta} Q^{-1}\mathrm{d}\, f_\theta(\mathbf{x}_k, u_k) + (I_m \otimes (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))^T Q^{-1})\mathrm{d}\ \mathrm{vec}\,\frac{\partial f_\theta}{\partial \theta^T},
\end{aligned} \tag{6.11}$$

assuming there are m parameters in θ. The second term above can be written as

$$\begin{aligned}
&\begin{bmatrix} (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))^T Q^{-1}\mathrm{d}\,\frac{\partial f_\theta}{\partial \theta_1} \\ \vdots \\ (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))^T Q^{-1}\mathrm{d}\,\frac{\partial f_\theta}{\partial \theta_m} \end{bmatrix} \\
&= \begin{bmatrix} (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))^T Q^{-1}\frac{\partial f_\theta}{\partial \theta_1 \partial \mathbf{x}^T}\mathrm{d}\,\mathbf{x} \\ \vdots \\ (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))^T Q^{-1}\frac{\partial f_\theta}{\partial \theta_m \partial \mathbf{x}^T}\mathrm{d}\,\mathbf{x} \end{bmatrix} \\
&= \begin{bmatrix} (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))^T Q^{-1}\frac{\partial f_\theta}{\partial \theta_1 \partial \mathbf{x}^T} \\ \vdots \\ (\mathbf{x}_{k+1} - f_\theta(\mathbf{x}_k, u_k))^T Q^{-1}\frac{\partial f_\theta}{\partial \theta_m \partial \mathbf{x}^T} \end{bmatrix}\mathrm{d}\,\mathbf{x}.
\end{aligned} \tag{6.12}$$

Therefore,

$$\left.\frac{\partial^2 l_k^1}{\partial\theta\partial\mathbf{x}_k^T}\right|_{\mathbf{x}_k=\hat{x}_{k|k}} = -\frac{\partial f_\theta^T}{\partial\theta}(\hat{x}_{k|k},u_k)Q^{-1}\frac{f_\theta}{\partial\mathbf{x}_k^T}(\hat{x}_{k|k},u_k) + \begin{bmatrix} (\mathbf{x}_{k+1}-f_\theta(\hat{x}_{k|k},u_k))^T Q^{-1}\frac{\partial f_\theta}{\partial\theta_1\partial\mathbf{x}_k^T}(\hat{x}_{k|k},u_k) \\ \vdots \\ (\mathbf{x}_{k+1}-f_\theta(\hat{x}_{k|k},u_k))^T Q^{-1}\frac{\partial f_\theta}{\partial\theta_m\partial\mathbf{x}_k^T}(\hat{x}_{k|k},u_k) \end{bmatrix}, \quad (6.13)$$

and the conditional expectation of $G_1(\hat{x}_{k|k},u_k)+\frac{\partial G_1(\mathbf{x}_k,u_k)}{\partial\mathbf{x}_k^T}|_{\mathbf{x}_k=\hat{x}_{k|k}}(\mathbf{x}_k-\hat{x}_{k|k})$ is equal to

$$g_{1k} = -2\frac{\partial f_\theta^T}{\partial\theta}(\hat{x}_{k|k},u_k)Q^{-1}(\hat{x}_{k+1|N}-f_\theta(\hat{x}_{k|k},u_k)) + 2\frac{\partial f_\theta^T}{\partial\theta}(\hat{x}_{k|k},u_k)Q^{-1}\left(\frac{\partial f_\theta}{\partial\mathbf{x}_k^T}(\hat{x}_{k|k},u_k)\right)(\hat{x}_{k|N}-\hat{x}_{k|k}) - 2\begin{bmatrix} \mathrm{tr}[Q^{-1}\frac{\partial f_\theta}{\partial\theta_1\partial\mathbf{x}_k^T}\Psi_k] \\ \vdots \\ \mathrm{tr}[Q^{-1}\frac{\partial f_\theta}{\partial\theta_m\partial\mathbf{x}_k^T}\Psi_k] \end{bmatrix}, \quad (6.14)$$

where

$$\Psi_k = \mathrm{E}[\mathbf{x}_k\mathbf{x}_{k+1}^T|\mathbf{Y}_N] - \hat{x}_{k|N}f_\theta^T(\hat{x}_{k|k},u_k) - \hat{x}_{k|k}\hat{x}_{k+1|N}^T + \hat{x}_{k|k}f_\theta^T(\hat{x}_{k|k},u_k).$$

In a similar way we can obtain for $G_2(\mathbf{x}_k,u_k)$ the approximated conditional expectation as

$$g_{2k} = -2\frac{\partial h_\varphi^T}{\partial\varphi}R^{-1}(y_k-h_\varphi(\hat{x}_{k|k-1},u_k)) + 2\frac{\partial h_\varphi^T}{\partial\varphi}R^{-1}\frac{\partial h_\varphi}{\partial\mathbf{x}_k^T}(\hat{x}_{k|N}-\hat{x}_{k|k-1}) - 2\begin{bmatrix} \mathrm{tr}[R^{-1}\frac{\partial h_\varphi}{\partial\varphi_1\partial\mathbf{x}_k^T}\Phi_k] \\ \vdots \\ \mathrm{tr}[R^{-1}\frac{\partial h_\varphi}{\partial\varphi_q\partial\mathbf{x}_k^T}\Phi_k] \end{bmatrix}, \quad (6.15)$$

assuming φ has q elements, and where

$$\Phi_k = \hat{x}_{k|N} y_k^T - \hat{x}_{k|N} h_\varphi^T(\hat{x}_{k|k-1}, u_k) - \hat{x}_{k|k-1}\mathbf{y}_k^T + \hat{x}_{k|k-1} h_\varphi^T(\hat{x}_{k|k-1}, u_k).$$

Therefore, the gradient of parameters $\begin{bmatrix} \theta \\ \varphi \end{bmatrix}$ is

$$\begin{bmatrix} \sum_{k=1}^N g_{1k} \\ \sum_{k=1}^N g_{1k} \end{bmatrix}, \tag{6.16}$$

where g_{1k} and g_{2k} are defined in equation (6.14) and equation (6.15), respectively. With gradient thus obtained we can use numerical optimization algorithms to iteratively estimate the parameters.

Variance matrices of the noise terms are also parameters and in theory should be estimated too, but due to the stringent requirement that they both be symmetric, positive definite, their estimation is not considered here. They are considered known.

1. Time-delay

Introduction of time-delays also introduce complications for state estimation and parameter estimation. The approach I finally settled on is the augmentation of states to include previous states, in order to recover the first-order Markov property. This is a straightforward extension with a computational penalty due to the manifold increase in state dimensions, but my attempts at finding a more efficient method made me realize that time-delays result in exponential increase of terms in Kalman filter and thus it is impractical to write a general formula.

Taking the original nonlinear model with additive noise and adding a time-delay term, we get

$$\begin{aligned}\mathbf{x}_{k+1} &= f_\theta(\mathbf{x}_k, \mathbf{x}_{k-\tau}, u_k) + \mathbf{w}_k \\ \mathbf{y}_k &= h_\varphi(\mathbf{x}_k, \mathbf{x}_{k-\tau}, u_k) + \mathbf{v}_k,\end{aligned} \tag{6.17}$$

where $\mathbf{x}_{k-\tau}$ is the state at time $k-\tau$, and everything else remains same. I have put only one time-delayed state in there to simplify formulas, and multiple time-delayed states will require minor modification because the augmented state vector will include every state between time k and $k-\tau$. It is also not an undue restriction to have state equation and output equation sharing the same time-delayed state, because when they are different, simply choose the one with the largest delay and make that delay τ, with the result that we have a multiple time-delayed state situation.

I will discuss state estimation of time-delay system first and the parameter estimation will follow that. From equation (6.17), I first take the first order Taylor approximation of the state equation to get

$$\begin{aligned}\mathbf{x}_{k+1} &\approx f_\theta(\hat{x}_{k|k}, \hat{x}_{k-\tau|k}, u_k) + F_{0,k}(\mathbf{x}_k - \hat{x}_{k|k}) + F_{1,k}(\mathbf{x}_k - \hat{x}_{k|k}) + \mathbf{w}_k \\ &= F_{0,k}\mathbf{x}_k + F_{1,k}\mathbf{x}_{k-\tau} + f_\theta(\hat{x}_{k|k}, \hat{x}_{k-\tau|k}, u_k) - F_{0,k}\hat{x}_{k|k} - F_{1,k}\hat{x}_{k-\tau|k} + \mathbf{w}_k,\end{aligned} \tag{6.18}$$

where

$$F_{0,k} = \left.\frac{\partial f_\theta}{\partial \mathbf{x}_k}\right|_{\substack{\mathbf{x}_k = \hat{x}_{k|k} \\ \mathbf{x}_{k-\tau} = \hat{x}_{k-\tau|k}}} \text{ and } F_{1,k} = \left.\frac{\partial f_\theta}{\partial \mathbf{x}_{k-\tau}}\right|_{\substack{\mathbf{x}_k = \hat{x}_{k|k} \\ \mathbf{x}_{k-\tau} = \hat{x}_{k-\tau|k}}}.$$

The augmented state vector χ_{k+1} then is equal to

$$\begin{bmatrix} x_{k+1} \\ x_k \\ \vdots \\ x_{k+1-\tau} \end{bmatrix} = \begin{bmatrix} F_{0,k} & 0 & \dots & 0 & F_{1,k} \\ I & 0 & \dots & \vdots & 0 \\ 0 & I & \dots & \vdots & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & I & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \\ \vdots \\ \mathbf{x}_{k-\tau} \end{bmatrix} + \begin{bmatrix} f_\theta(\hat{x}_{k|k}, \hat{x}_{k-\tau|k}, u_k) - F_{0,k}\hat{x}_{k|k} - F_{1,k}\hat{x}_{k-\tau|k} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{w}_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{6.19}$$

which can be rewritten as

$$\chi_{k+1} = F_{\text{new}}\chi_k + f_{\text{new}}(\hat{\chi}_{k|k}, u_k) + G\mathbf{w}_k, \tag{6.20}$$

where F_{new}, f_{new} and G are obvious from equation (6.19). This is the form that one can directly plug in to the extended Kalman filter to estimate states for the state equation.

The output equation can be dealt with in exactly the same way to get

$$\mathbf{y}_k = \begin{bmatrix} H_{0,k} & 0 & \cdots & H_{1,k} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \\ \vdots \\ \mathbf{x}_{k-\tau} \end{bmatrix} \tag{6.21}$$

$$+ h(\hat{x}_{k|k-1}, \hat{x}_{k-\tau|k-1}, u_k) - H_{0,k}\hat{x}_{k|k-1} - H_{1,k}\hat{x}_{k-\tau|k-1} + v_k,$$

where

$$H_{0,k} = \left.\frac{\partial h_\varphi}{\partial \mathbf{x}_k}\right|_{\substack{\mathbf{x}_k = \hat{x}_{k|k-1} \\ \mathbf{x}_{k-\tau} = \hat{x}_{k-\tau|k-1}}} \text{ and } H_{1,k} = \left.\frac{\partial h_\varphi}{\partial \mathbf{x}_{k-\tau}}\right|_{\substack{\mathbf{x}_k = \hat{x}_{k|k-1} \\ \mathbf{x}_{k-\tau} = \hat{x}_{k-\tau|k-1}}}.$$

Put the equation above into an augmented state form we have

$$\mathbf{y}_k = H_{\text{new}}\chi_k + h(\hat{\chi}_{k|k-1}, u_k) - H_{\text{new}}\hat{\chi}_{k|k-1} + \mathbf{v}_k. \tag{6.22}$$

The innovation is a simple equation:

$$\mathbf{e}_k = \mathbf{y}_k - h(\hat{x}_{k|k-1}, \hat{x}_{k-\tau|k-1}, u_k),$$

which is the same as regular extended Kalman filter except for the time-delay term in h. I will not list the extended Kalman filter for augmented state χ_k because they would look exactly the same as the regular Kalman filter. In fact, due to the pattern of f_{new}, F_{new} and H_{new} one can write a program that takes f_θ, $F_{0,k}$, $F_{1,k}$, h_φ, $H_{0,k}$, and $H_{1,k}$ and returns the augmented-state system with new f_{new}, F_{new}, and H_{new}, and then feeds the new system to an existing extended Kalman filter program, which is what I did.

After taking care of state estimation, parameter estimation is straightforward.

First, let us redefine the augmented state

$$\chi_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-\tau} \end{bmatrix}.$$

(One can reuse the augmented state from the state estimation, but the only difference is some additional zero matrices to deal with unwanted states between k and $k-\tau$.) Then everything in this section up to equation (6.14) remains valid if we replace every occurance of $\mathbf{x}_k$ with χ_k, and similiarly its estimators, $\hat{\mathbf{x}}_{k|k}$ and $\hat{\mathbf{x}}_{k|N}$ with $\hat{\chi}_{k|k}$ and $\hat{\chi}_{k|N}$. What does change is Ψ_k, because Ψ_k comes from the expectation of $(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_{k+1} - f(\mathbf{x}_k, u_k))^T$. Replacing $\mathbf{x}_k$ by χ_k and $\hat{\mathbf{x}}_{k|k}$ with $\hat{\chi}_{k|k}$, we get

$$\begin{aligned} &\begin{bmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_{k|k} \\ \mathbf{x}_{k-\tau} - \hat{\mathbf{x}}_{k-\tau|k} \end{bmatrix} (\mathbf{x}_{k+1} - f(\chi_k, u_k))^T \\ &= \begin{bmatrix} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_{k+1} - f(\chi_k, u_k))^T \\ (\mathbf{x}_{k-\tau} - \hat{\mathbf{x}}_{k-\tau|k})(\mathbf{x}_{k+1} - f(\chi_k, u_k))^T \end{bmatrix} \\ &= \begin{bmatrix} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_{k+1} - f(\chi_k, u_k))^T \\ \mathbf{x}_{k-\tau}\mathbf{x}_{k+1}^T - \mathbf{x}_{k-\tau} f(\chi_k, u_k)^T - \hat{\mathbf{x}}_{k-\tau|k}\mathbf{x}_{k+1}^T + \hat{\mathbf{x}}_{k-\tau|k} f(\chi_k, u_k)^T \end{bmatrix}, \end{aligned} \tag{6.23}$$

and taking the conditional expectation of the equation above yields

$$\begin{bmatrix} \mathrm{E}[\mathbf{x}_k \mathbf{x}_{k+1}^T | \mathbf{Y}_N] - \hat{x}_{k|N} f^T(\hat{x}_{k|k}, u_k) - \hat{x}_{k|k}\hat{x}_{k+1|N}^T + \hat{x}_{k|k} f^T(\hat{x}_{k|k}, u_k) \\ \mathrm{E}[\mathbf{x}_{k-\tau}\mathbf{x}_{k+1}^T | \mathbf{Y}_N] - \hat{\mathbf{x}}_{k-\tau|k}\hat{\mathbf{x}}_{k+1|N} - \hat{\mathbf{x}}_{k-\tau|N}\hat{\mathbf{x}}_{k+1|k}^T + \hat{\mathbf{x}}_{k-\tau|k}\hat{\mathbf{x}}_{k+1|k}^T \end{bmatrix}. \tag{6.24}$$

Notice that I have replaced the occurrences of $f(\hat{x}_{k|k}, u_k)$ with $\hat{\mathbf{x}}_{k+1|k}$, an equality that comes from EKF. The term at the top is the old Ψ_k. The term at the bottom can be computed from quantities available from augmented state estimation. Among the estimates at the bottom term, $\hat{\mathbf{x}}_{k-\tau|k}$ can be extracted from the filtered estimator at time k; $\mathrm{E}[\mathbf{x}_{k-\tau}\mathbf{x}_{k+1}^T | \mathbf{Y}_N]$ can be computed from the lag-on covariance returned from the extended Kalman smoother, because

$$\mathrm{cov}\left(\begin{bmatrix}\mathbf{x}_{k+1}\\ \mathbf{x}_k\\ \vdots\\ \mathbf{x}_{k+1-\tau}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k\\ \mathbf{x}_{k-1}\\ \vdots\\ \mathbf{x}_{k-\tau}\end{bmatrix}\middle|\mathbf{Y}_N\right)=\begin{bmatrix}\mathrm{cov}(\mathbf{x}_{k+1},\mathbf{x}_k|\mathbf{Y}_N) & \cdots & \mathrm{cov}(\mathbf{x}_{k+1},\mathbf{x}_{k-\tau}|\mathbf{Y}_N)\\ \vdots & \ddots & \vdots\\ \mathrm{cov}(\mathbf{x}_{k+1-\tau},\mathbf{x}_k|\mathbf{Y}_N) & \cdots & \mathrm{cov}(\mathbf{x}_{k+1-\tau},\mathbf{x}_{k-\tau}|\mathbf{Y}_N)\end{bmatrix}, \tag{6.25}$$

and of course

$$\mathrm{E}[\mathbf{x}_{k-\tau}\mathbf{x}_{k+1}^T|\mathbf{Y}_N]=\mathrm{cov}(\mathbf{x}_{k-\tau},\mathbf{x}_{k+1}|\mathbf{Y}_N)+\hat{\mathbf{x}}_{k-\tau|N}\hat{\mathbf{x}}_{k+1|N}.$$

In a similiar way we can change Φ_k as well.

There is of course a price to pay for the augmentation of states in estimation, and that is computational cost. Kalman filter runs roughly in cubic power of the dimension of states [90], so one time step delay increases the computational time by eight times. The storage cost also goes up, but that is not a major concern in the days of cheap computer memory.

C. Experiments and Results

I applied the EM algorithm above to three datasets in order to gauge its effectiveness. One is a synthetic dataset whose generating functions are taken from Will et al. [171], and another synthetic dataset is generated using an estimated JAK-STAT pathway. Synthetic data afford us flexibility in the amount of data available and an ability to peek behind the curtain to know the true states and true parameters. But synthetic data also has handicaps in that real-world data might be different, which is why data from JAK-STAT pathway from Swameye et al. [172] as the second dataset. By comparing estimated values with observations or with true values, it appears that the

EM algorithm is able to estimate well.

1. Synthetic Data of a Simple System

The nonlinear system that generates the synthetic data is a simple one-state system:

$$x_{k+1} = ax_k + \frac{x_k}{b + x_k^2} + u_k + w_k \tag{6.26}$$
$$y_k = cx_k + dx_k^2 + v_t,$$

where a, b, c, d are all parameters, u_k is scalar input that is generated randomly observing a normal distribution, $\mathcal{N}(0, 1)$, and w_k and v_k are white Gaussian noise with means zero and variance 0.01; x_k and y_k are the state and output, respectively, and they are both scalars. This is a nonlinear state-space model where parameters do not all appear linearly in the model, and it has the functional form that one often encounters in systems biology.

The true parameters are $a = 0.7$, $b = 0.6$, $c = 0.5$, $d = 0.4$. The estimated parameters are $\hat{a} = 0.73$, $\hat{b} = 0.4$, $\hat{c} = 0.32$, $\hat{d} = 0.29$, which are somewhat close but cannot be said to have recovered the true values. Because this is synthetic data, we have the true state values, so in Figure 31 I compare the estimated state values against the true values. The errors, plotted in the bottom panel, are generally small, and the estimated states track true states well. But that is on the training dataset, which is the data used to estimate the model. A tougher test is how the model handles fresh data. In Figure 32 I make the same comparison but use a fresh set of data. The estimated states are computed using extended Kalman smoother and the estimated model. We can see here that errors can be larger than those of the training set, but not much larger, and the estimated states track true states well. In other words, the true parameters are not recovered fully, but all the features of dynamics are represented by the estimated model as evidenced by the ability of estimated states,

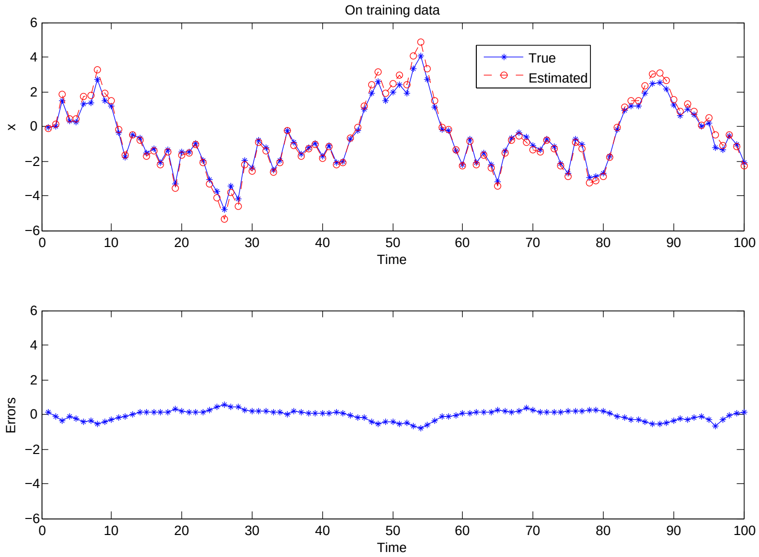


Fig. 31.: Comparison of estimated states and true states on the training data. The top panel has the estimated state values imposed on the plot of true state values. The bottom panel is the plot of the errors of estimated state values. In general, estimation is close to true values

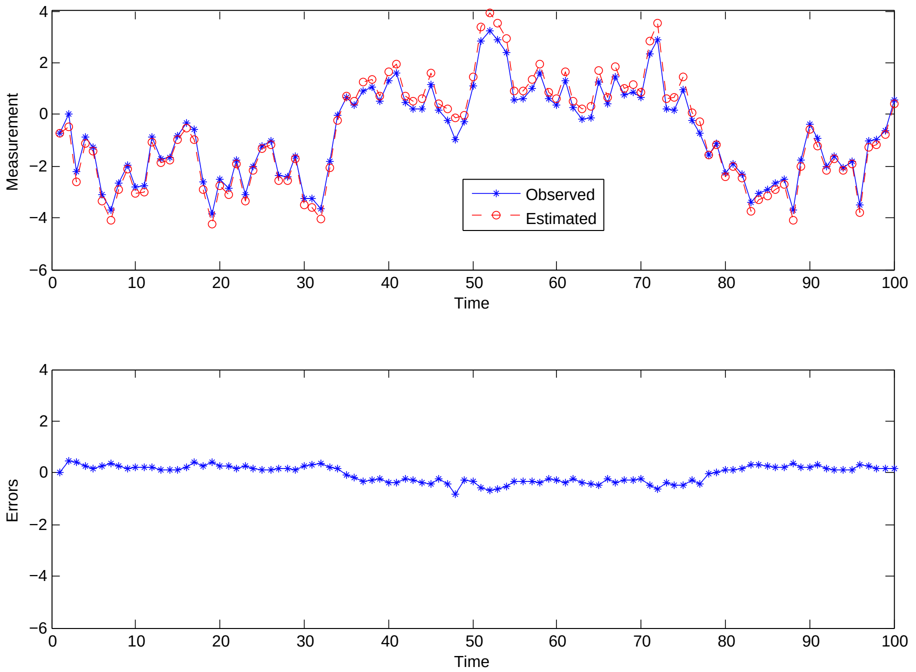


Fig. 32.: Comparison of estimated states and true states on a validating data. The errors are relatively small and estimated states track well true states.

using the estimated model, to track true states well.

2. JAK-STAT Pathway

The Janus family of kinases (JAK)-signal transducer and activator of transcription (STAT) signaling pathway is a well studied pathway that signals through multiple cell surface receptors, of which the erythropoietin receptor (EpoR) is particularly important. Hormone Epo binds to the receptor and activates the receptor-bound tyrosine kinase JAK2, thus creating docking site for down-stream molecules such as latent transcriptor factor STAT5. Once STAT5 is recruited by the activated receptor, it is tyrosine phosphorylated; it then dimerizes and migrates to the nucleus where it stimulates transcription of target genes. A diagram of the part of JAK-STAT pathway

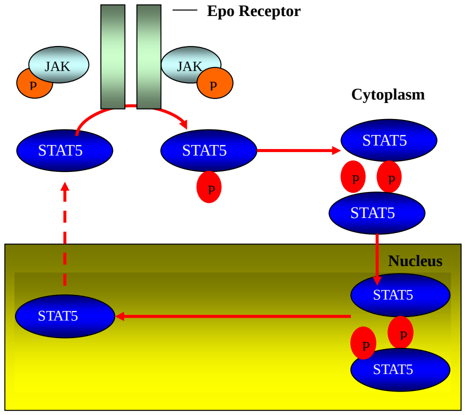


Fig. 33.: A part of JAK-STAT signaling pathway under study. STAT5 is recruited by activated Epo receptor and is phosphorylated, and it dimerizes and migrates to the nucleus to stimulate transcription of target genes [172].

under study is in Figure 33.

The dephosphorylated STAT5 monomer is x_1, x_2 is the phosphorylated STAT5, x_3 is phosphorylated STAT5 dimer, and x_4 is STAT5 dimer in nucleus. According to Swameye et al. [172], the states are governed by a set of continuous ordinary differential equations, which upon simple discretization is taken to be:

$$\begin{aligned} x_1(t+1) &= x_1(t) - k_1 x_1(t)u(t) + 2k_4 x_3(k-\tau) \\ x_2(t+1) &= x_2(t) + k_1 x_1(t)u(t) - 2k_2 x_2(t)^2 \\ x_3(t+1) &= x_3(t) + k_2 x_2(t)^2 - k_3 x_3(t) \\ x_4(t+1) &= x_4(t) + k_3 x_3(t) - k_4 x_3(t-\tau), \end{aligned} \tag{6.27}$$

where the states at time t is represented by $x_i(t)$ for $i = 1 \ldots 4$, and k_1, k_2, k_3, k_4 are parameters to be estimated. Additive noise terms are not explicitly represented here but are assumed present. τ is a natural number representing time-delay of approximately 6 minutes. The input u is the concentration level of EpoR.

The outputs are the phosphorylated STAT5 (y_1) and the total amount of STAT5 (y_2), which are related to states by

$$\begin{aligned} y_1(t) &= k_5(x_2(t) + 2x_3(t)) \\ y_2(t) &= k_6(x_1(t) + x_2(t) + 2x_3(t)), \end{aligned} \tag{6.28}$$

where again noise terms are implicitly assumed.

Gene expressing EpoR was introduced into BaF3 cells, which were starved for 3 hours and then stimulated with 5units/ml Epo. Immunoblotting provided measurements, of which there were 16 time points, but only the first 11 were of uniform time steps, so they are used in parameter estimation.

The data comes with estimate of variances for observation, which are uncorrelated and y_1 has 0.08 and y_2 with 0.1. The state-transition noise is taken to be identity. Because time step is 2 minutes apart, the time-delay is 3 time steps. The estimated parameters are $k_1 = -0.018 \quad k_2 = 2.06 \quad k_3 = 0.04 \quad k_4 = 1.08 \quad k_5 = 1.05 \quad k_6 = 1.05$, which are different from those obtained by Quach et al. [168], but they did not employ a time-delay system and instead used a step-function which is zero until a certain step, after which it becomes a regular state, so a direct comparison is not apt. But for the observation equation, for which my version is the same as theirs, they had k_5 and k_6 both as 1, which is close to 1.05, my estimated value for k_5 and k_6. In addition, we can look at the estimated outputs and observed outputs. The plots of comparison of both outputs y_1 and y_2 of the training dataset are in Figure 34 and 35, respectively. From these we see that the estimated outputs closely follow the observed outputs, thus the major features of observed dynamics are reproduced by the estimated model.

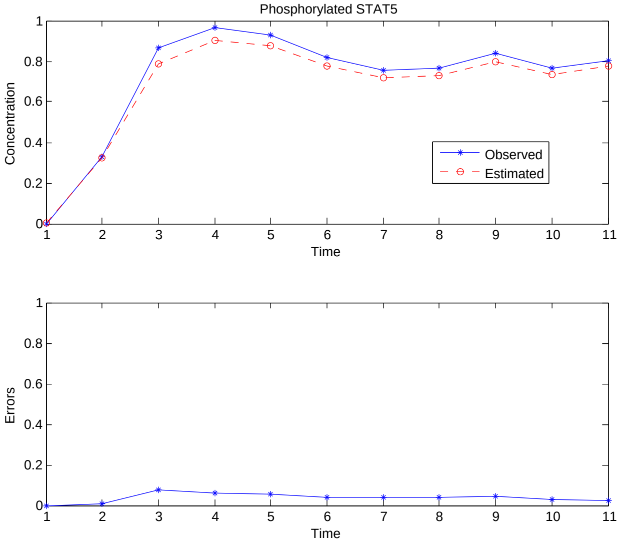


Fig. 34.: Comparison of estimated y_1 and observed values.

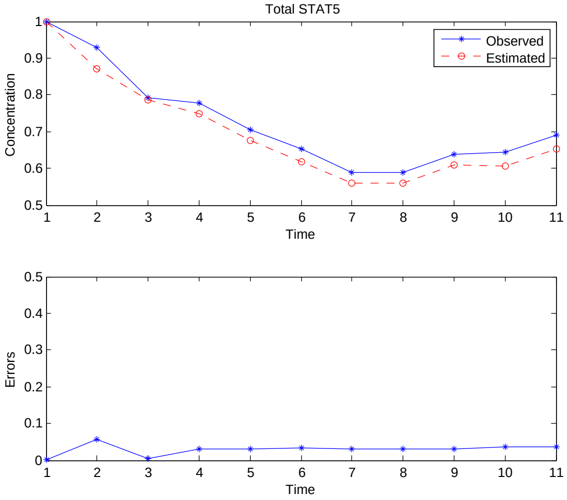


Fig. 35.: Comparison of estimated y_2 and observed values.

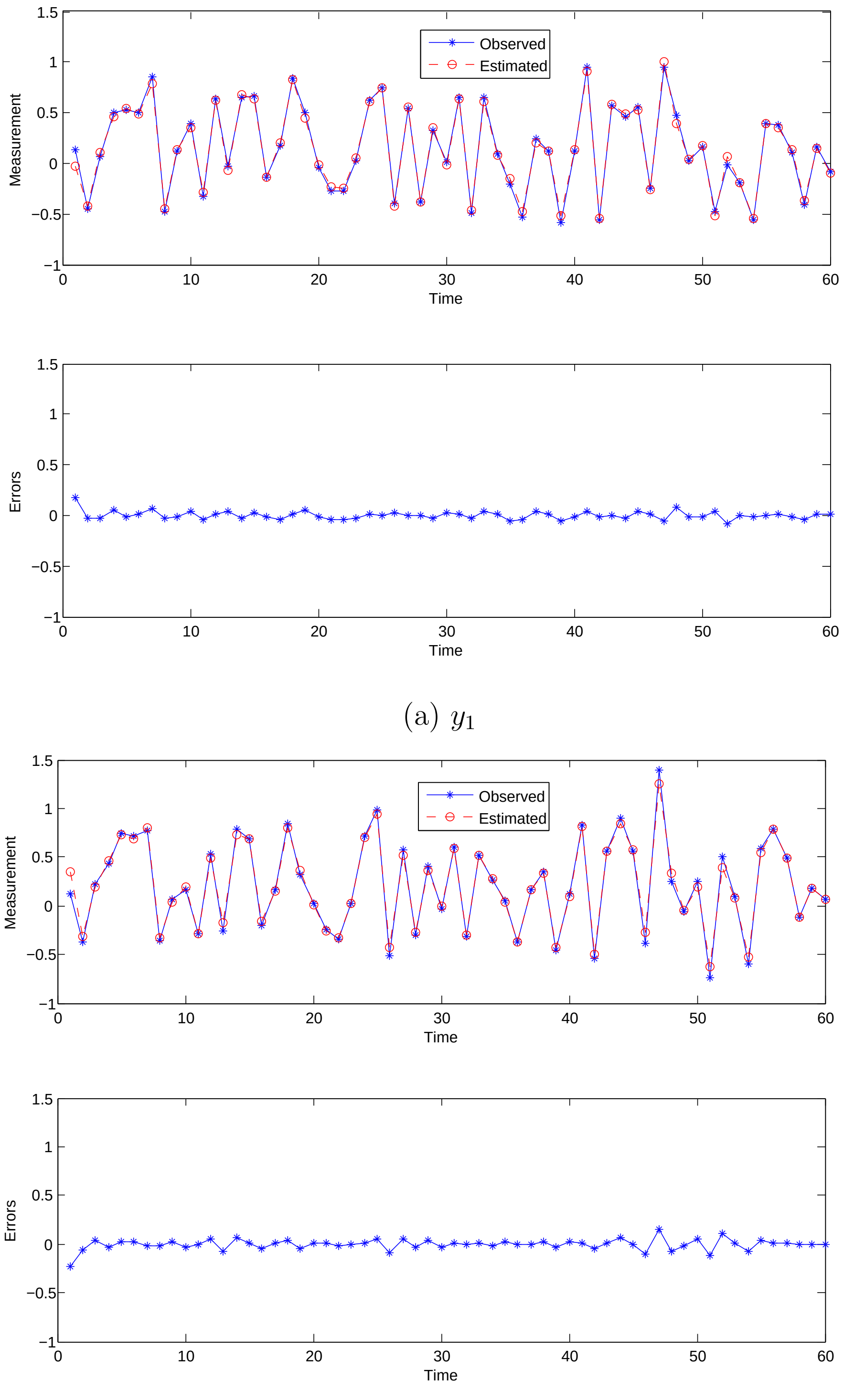
3. Synthetic Data Using JAK-STAT Model

I also tested my method on synthetic data generated using an estimated model of JAK-STAT pathway. The model and code are supplied by Quach et al. [168] for their paper on parameter estimation of biological networks published in *Bioinformatics*. This gives me an opportunity to test on a more complex system with more data, although biological application is likely to see far fewer time points than the 60 I generated.

To ensure that my method performs satisfactorily under diverse noise conditions, I tested it under varying ratio of system and observation noise. The system noise observes a Gaussian distribution with mean zero and variance $0.01 \times I$, where I is an identity matrix. The observation noise is also a Gaussian distribution with mean zero but variance $\sigma \times I$, where σ varies from 0.01 to 0.1.

Here I plot the estimated outputs and observed outputs at noise-ratio of 1, 5, and 10, as well as errors at these ratios. As we can see in Figure 36, for noise-ratio of 1, the estimated values can track observed values well, with small errors. For noise-ratio of 5, the plots in Figure 37 show that there is an increase in errors, but the estimated values can still follow observed values relatively well. However, for noise-ratio of 10, as seen in Figure 38, the errors are noticeable, with estimated values follow in the direction of observed values but not closely in amplitude. This is in accordance with expectation that in situation of high noise observations cannot be entirely trusted and therefore not closely followed. This demonstrates that in situation of very high observation noise, my method can produce models that reproduce the outline of observed trajectory.

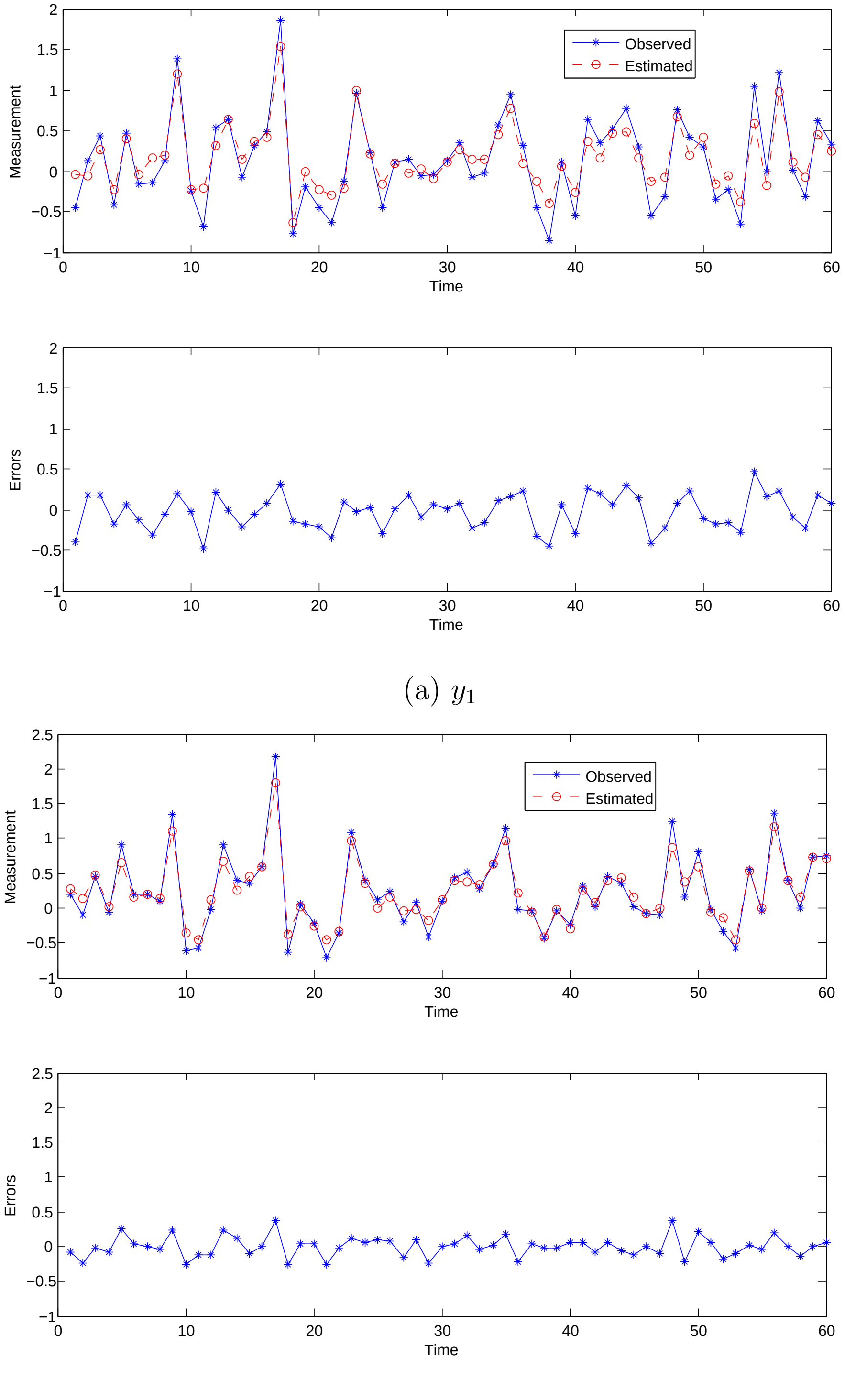
To compare the degradation of performance due to high noise, I also computed the mean square error (MSE) of predicted values, $\hat{y}_{k|k-1}$, against observation, in other



(a) y_1

(b) y_2

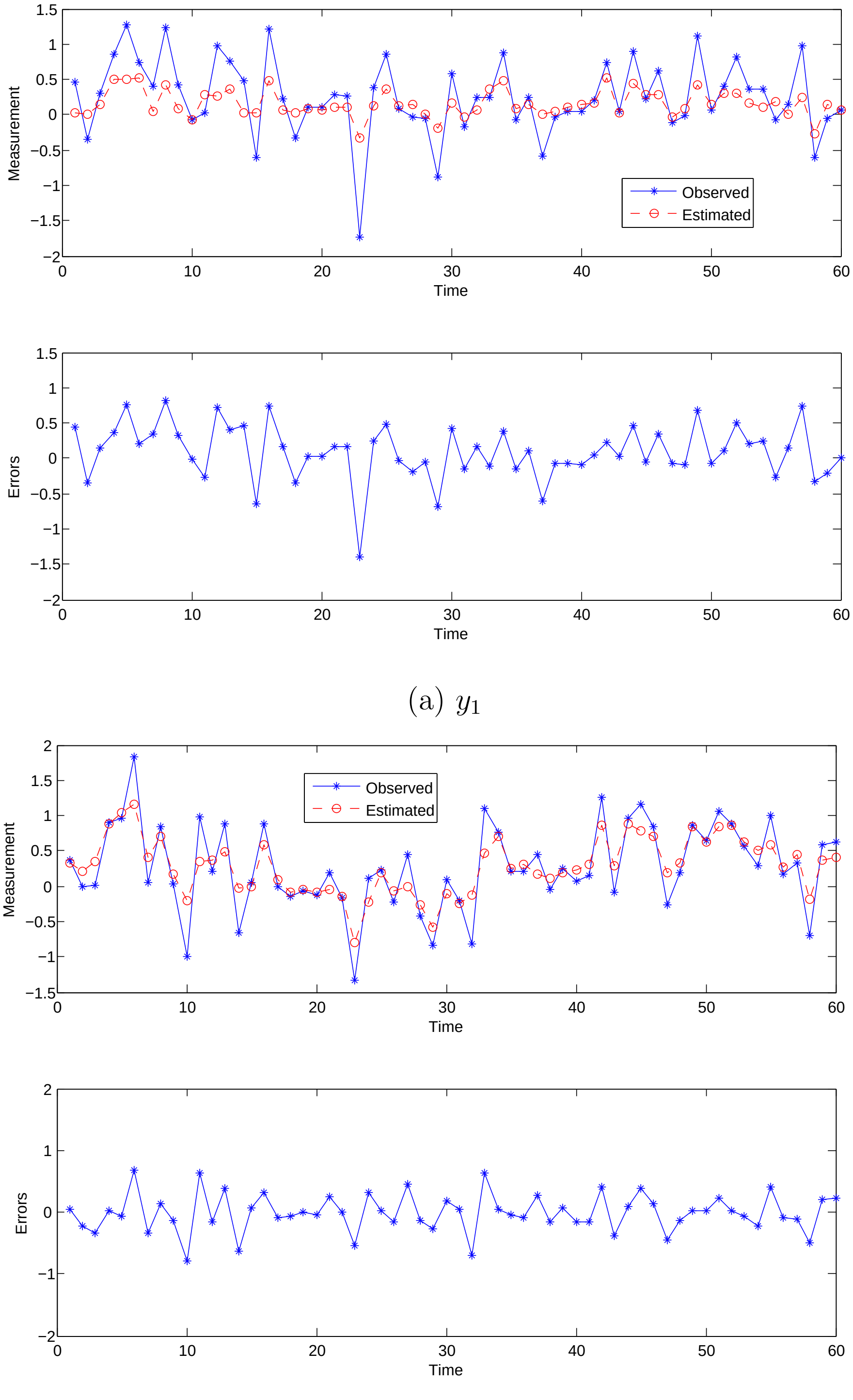
Fig. 36.: Comparison of estimated and observed y_1 and y_2 at noise-ratio of 1. The estimated value track observed values pretty closely.



(a) y_1

(b) y_2

Fig. 37.: Comparison of estimated and observed y_1 and y_2 at noise-ratio of 5. The estimated value track observed values somewhat closely.



(a) y_1

(b) y_2

Fig. 38.: Comparison of estimated and observed y_1 and y_2 at noise-ratio of 10. The estimated value track observed values approximately with significant errors.

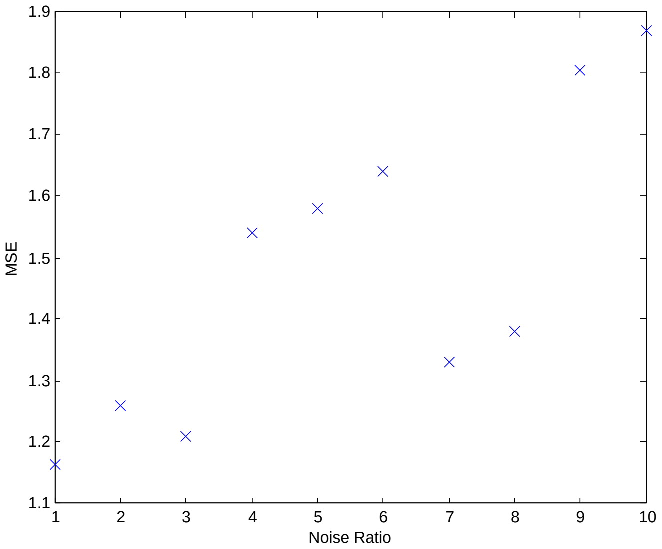


Fig. 39.: MSE as noise ratio varies. In general, MSE increases as noise-ratio increases, with some noticeable fluctuations.

words, innovation. Define MSE as

$$\frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_{k|k-1})\mathrm{Re}_k^{-1}(y_k - \hat{y}_{k|k-1})^T,$$

where $\mathrm{Re}_k =< y_k - \hat{y}_{k|k-1}, y_k - \hat{y}_{k|k-1} >$. The inverse of variance is to normalize MSE so that it is comparable under different noise conditions. The plot of MSE as noise varies from 1 to 10 is plotted in Figure 39, where we can see that in general MSE increases as noise-ratio increases, but there are some fluctuations.

D. Discussion

In this chapter I presented an expectation-maximization method that can estimate a parameters of nonlinear state-space models. Due to the generality of nonlinear state-space models, my method can be applied to detailed, mechanistic studies of genetic regulatory networks, signal transduction networks, and metabolic networks. For instance, chemical rate-equations that are often used to model biochemical pathways [172, 54, 11] can be handled by the method presented here. Of course, there is nothing that would prevent its application outside of biological contexts. Wills et al. [171] and Schön et al. [165] have already proposed EM method that use particle filters as the state and gradient estimation method in engineering context. Compared to their methods, I use the extended Kalman filter, which is simpler than the particle filter, although sometimes at the expense of accuracy. Another recent paper by Quach et al. [168] used unscented Kalman filter to simultaneously estimate states and parameters of biological networks. The benefits of simultaneous estimation of states and parameters are speed and relative simplicity, while the weakness is inaccuracy.

Another benefit of extended Kalman filter is that it allows me to derive a version for systems with time-delays. Time-delays occur in biological context because molecules take time to travel inside and outside of cells, which can be modeled as time-delays in ordinary differential equations [98, 173]. Parameter and state estimation for time-delay systems are active research topics today in engineering [174, 175, 176], and are novel in biological context for state-space models. By a relatively simple extension of the extended Kalman filter, we are now able to handle models with known, constant time-delays, and we can expect more use of time-delays in the future in systems biology because time-delays in biological networks are mostly averages of Stochastic events, thus constants.

This chapter also represents a natural progression of model complexity from linear state-space models, which are mostly used in top-down approach, to nonlinear state-space models, which are more often seen in bottom-up approach. In our quest for functional understanding of biological systems, it is inevitable that we want to have a more detailed, mechanistic look at cells. The generality of state-space allows us to add more details as more data and more knowledge become available, and the general applicability of the nonlinear EM algorithm presented in this chapter means it can accompany us as model complexity increases and nonlinearity creeps into models. For example, many people have tried to model transcriptional factor binding in their genetic regulatory networks [177, 66, 178, 76], but because transcriptional factor binding is a nonlinear biochemical process, linear models have difficulty to capture all the features. Nachman et al. [57] is one of few who attempted a mechanistic model of gene regulation, using dynamic Bayesian networks. Their rate-equations are based on Michaelis-Menten equations for transcription factors, and they did not use equations for protein production and degradation. The method presented in this chapter is more general and can deal with protein production and degradation. In fact, it can model signal transduction networks and metabolic networks, in addition to genetic regulatory networks, as long as they are represented by ordinary differentiation equations in state-space form. This opens up a vast range of possibility for models of biological networks and for the possible details of those networks.

The technical contribution of this work is a new method for a demanding task. Ljung, a highly respected researcher in systems identification and author of the reference book on systems identification, noted in [179], a 2006 survey paper on nonlinear systems identification, that the maximum likelihood "approach is conceptually simple, but could be very demanding in practice, . . . ". He only noted one paper, beside one other using particle filter, that follows EM algorithm for parameter estimation, which is published in the 14th IFAC Symposium on System Identification by Schön et al. [165], and in their method the systems equations are affine functions of parameters. My method is more general in that systems equations can be arbitrary nonlinear differentiable functions of parameters. Also, as my method differs from Schön et al.'s method [165] in that they used particle filter for state estimation while I use extended Kalman filter, which means that they can apply, unmodified, particle filter to gradient approximation while I need a fresh set of formulas, my method has comparable technical contribution as Schön et al.'s paper.

E. Conclusion

The nonlinear EM method presented in this chapter can tackle general nonlinear state-space models with additive noise. It can be applied to parameter estimation in biological networks. Testing the method on synthetic datasets and a real-world dataset demonstrated its ability to generate models that can reproduce major dynamic features of the observed outputs.

The problem of parameter estimation for nonlinear state-space models is a difficult one. As we see in the synthetic data case that multiple parameter values can reproduce the same output trajectory, thus presenting the question of which one to choose. Identifiability should be verified before attempting parameter estimation, but it still does not guarantee a unique estimation. This is also related to the fact that EM algorithm in general is local optimization, not global, so that no matter how many initial values we try, we can never be sure that it is the best estimate. How to take into account the locally optimal nature of EM and to produce a good interval estimate with confidence measure is a worthwhile research topic.

EM algorithm is a local optimization method that could only find local maximum of the log-likelihood function. There are also attempts at using global optimization in parameter estimation [180]. Because nonlinear EM algorithm presented in this chapter has a numerical optimization as the M-step, in theory global optimization approaches such as simulated annealing or genetic algorithms can be used instead in the M-step. But I have not addressed this topic in this dissertation.

Another problem is that extended Kalman filter (EKF) is not necessarily a stable estimator, unlike the regular Kalman filter, and therefore EKF could diverge depending on initial values for the initial state. This problem appeared a number of times in the JAK-STAT pathway experiment, where time-delay causes some estimates to diverge.

Also the problem of estimating variance matrices is not dealt with here. In theory Kalman filter can estimate both the noise and their variance [181], but I only have square-root version for the state-transition noise, not observation noise, thus no way to guarantee their positive definiteness. Here square-root version refers to obtaining the square-root of variance matrices, and then the variance matrices themselves are obtained by squaring the square-roots together, thus guaranteeing the positive definiteness of the result. This is also the way square-root version of Kalman filter estimates variance matrices. Directly taking derivatives of variances offers no guarantee either, unlike the linear case where an analytic, square-root solution exists.

CHAPTER VII

CONCLUSION AND FUTURE WORKS

A. Conclusion

I started with constrained systems identification of the linear state-space models of genetic regulatory networks. Dynamic models of genetic regulatory networks have two properties that impact the networks' dynamics: structure and parameters. Because linear model of genetic networks are coarse-grained models, their parameters do not necessarily have physical meaning, so they are difficult to measure directly, so parameter estimation is often a necessary first step in dynamic study of genetic networks. A large number of approaches are proposed where parameter estimation and structure learning are bound together. They usually start with full connectivity for a network of interest and estimate all possible parameters as being realized, and then they prune their network by cutting away those links whose corresponding parameters are near zero. This takes advantage of a simple correspondence between parameters and structures of linear models, but it is prone to over-fitting. As I demonstrated in my study, full parametrization results in smaller estimation errors of observations for training data but larger errors for fresh, validating data. There is an exception [10] to this joint approach, and in the same spirit, I tackled parameter estimation of linear state-space models of genetic regulatory networks by assuming that the structure is known, thus making it possible to separate parameter estimation and structure learning for linear state-space models of genetic networks. I applied my method to two synthetic datasets and a SOS DNA repair network dataset. In general, fully parameterized estimation method returned worse prediction errors on fresh, validating data than constrained EM algorithm. I further tested the performance of my method using bootstrapping and compared the intervals of estimated parameters and the eigenvalues of estimated systems returned by two methods. The intervals returned by constrained EM algorithm are tighter and, where the true values are known, better centred around the true values. The constrained EM algorithm is general enough that it can be applied to any linear state-space systems with constraints on the parameters, not just genetic regulatory networks, and in fields other than systems biology.

Following parameter estimation, we obtain a dynamic model of genetic regulatory networks. I then asked whether the genetic networks' dynamic properties are differentially expressed in healthy vs. sick cells. Genes play an important role in the health of cells, and genetic networks made up of those important genes should behave differently in healthy vs. sick cells. I studied four dynamical properties, stability, relative stability, controllability, and transient behaviours (overshoot, settling time, and rise time) in three genetic networks, the SOS DNA repair networks, the glutathione (GSH) redox cycle, and the mitogen-activated protein kinase (MAPK) network. I found considerable difference in their dynamic properties when they are healthy compared to when they are sick.

The above suggests one way to treat diseases is to regulate key genetic regulatory networks' dynamics in sick cells so they behave more like they do in healthy cells. So the third part in the dynamic study I applied linear quadratic tracking (LQT) to genetic regulatory networks. I chose the average expression levels of genetic networks in desirable states (healthy) as the target value, due to the inherent Stochastic nature of gene expression. I found in general we could drive genetic networks to be close to the target value but not the precise value. I also found it is difficult to regulate genetic networks where inputs only directly influence a few genes at the top while the outputs are a few rungs downstream. This could have impact on the choice of drug targets.

All the works above are about linear models of genetic networks whose components are mostly genes, which in fact do not directly interact for the most part. If we want to incorporate RNAs or proteins, through which genes regulate themselves and cellular functions, linear models are no longer sufficient. Nonlinear ordinary differential equations have long been used to study signal transduction networks, and they can be used to study genetic regulatory networks too. So the last part of this dissertation was about the parameter estimation of nonlinear state-space dynamic models of biological networks, not just genetic networks, because nonlinear models can also describe signal transduction networks or metabolic networks. I chose the familiar generalized EM algorithm. The nonlinear EM algorithm still has two steps, the E-step for state estimation, and the M-step for parameter estimation. The E-step is performed by extended Kalman filter, a simple extension of linear Kalman filter. The M-step is by gradient-based numerical optimization method, because there is no analytic solution for general nonlinear optimization. When I tried to apply this method to real world data, I found that some signal transduction networks need time-delay to adequately describe observed dynamics and due to compartmentalization of cells. Therefore, I augmented states in both extended Kalman filter and the M-step to account for time-delays, to obtain an EM algorithm that can estimate parameters of nonlinear state-space dynamic models with time-delay. The method is applied to two synthetic dataset and a real world world dataset of JAK-STAT signaling pathway, to verify the nonlinear EM algorithm.

In summary, I developed methods for, or applied existing method to, constrained systems identification, dynamic analysis, and optimal control of linear state-space dynamic models of genetic regulatory networks; and for parameter estimation of nonlinear state-space dynamic models of signal transduction networks, with or without time-delay. The main contributions are the novelty of the very proposal of dynamic analysis of genetic networks, of the mathematical method of parameters estimation with constraints and application of optimal control for linear state-space models, and of the use of EM algorithm for parameter estimation of nonlinear systems, particularly for nonlinear systems with time-delay. I encountered practical problems in my dynamic study of biological networks and I found practical solutions for these problems. I expect the novelties of my work will further stimulate interest in the separation of parameter estimation and structure learning of genetic networks for the sake of improved fidelity, and in the dynamic analysis of biological networks and their optimal control for treatment. I also expect that my different method of nonlinear parameter estimation can add to the repertoire of methods for this difficult problem.

B. Future Works

I have already discussed future works in a lot of previous chapters depending on the chapter's topic, so I will not repeat them here. Instead, I will discuss a little about continuous–discrete models, which have continuous state evolution but discrete-time measurements.

Continuous–discrete models are very useful for irregularly observed longitudinal data where data are collected at arbitrary time points. Stochastic differential equations are equally useful for continuous-discrete models. I plan to study statistical methods for parameter and state estimation in Stochastic differential equations and apply the developed methods to identification of biological networks and longitudinal genetic epidemiology study.

1. Linear Continuous–discrete Kalman Filter

This section follows mostly [182], although any book on Stochastic processes should cover this material.

Consider a linear stochastic differential equation:

$$dX = F(t)X(t)dt + u(t) + L(t)d\beta(t)$$

where $X(0)$ is distributed as $\mathcal{N}(m(0), P(0))$, $F(t)$ and $L(t)$ are matrix-valued functions that serve the same function as time-varying parameters in the discrete systems, $u(t)$ is a known non-random function and $\beta(t)$ is a Brownian motion with diffusion matrix $Q_C(t)$. The observation equation is given by

$$Y_k = C_k X(t_k) + D_k u_k + v_k, \qquad v_k \sim \mathcal{N}(0, R_k),$$

where C_k and D_k are time-varying parameters relating states and inputs to the outputs. Informally, we have

$$\begin{aligned} d[e^{-\int_{t_0}^t F(s)ds}X] &= -F(t)e^{-\int_{t_0}^t F(s)ds}X(t)dt + e^{-\int_{t_0}^t F(s)ds}dX(t) \\ &= e^{-\int_{t_0}^t F(s)ds}u(t) + e^{-\int_{t_0}^t F(s)ds}L(t)d\beta, \end{aligned}$$

which implies that

$$e^{-\int_{t_0}^t F(s)ds}X(t) = X(0) + \int_{t_0}^t e^{-\int_{t_0}^\tau F(s)ds}u(\tau)d\tau + \int_{t_0}^t e^{-\int_{t_0}^\tau F(s)ds}L(\tau)d\beta$$

or

$$X(t) = e^{\int_{t_0}^t F(s)ds}X(0) + \int_{t_0}^t e^{\int_\tau^t F(s)ds}u(\tau)d\tau + \int_{t_0}^t e^{\int_\tau^t F(s)ds}L(\tau)d\beta. \qquad (7.1)$$

Taking expectation on both sides of equation (7.1), we obtain

$$m(t) = e^{\int_{t_0}^t F(s)ds}m(0) + \int_{t_0}^t e^{\int_0^t F(s)ds}u(\tau)d\tau$$

Similarly, we can obtain the variance matrix

$$Q(t) = e^{\int_{t_0}^{t} F(s)ds} Q(0) e^{\int_{t_0}^{t} F^T(s)ds} + \int_{t_0}^{t} e^{\int_{\tau}^{t} F(s)ds} L(\tau) Q(\tau) L^T(\tau) e^{\int_{\tau}^{t} F^T(s)ds} d\tau,$$

which will appear in the prediction part of the filter. Equation (7.1) can be rewritten as

$$X(t_k) = e^{\int_{t_{k-1}}^{t_k} F(s)ds} X(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{\int_{\tau}^{t_k} F(s)ds} u(\tau) d\tau + \int_{t_{k-1}}^{t_k} e^{\int_{\tau}^{t_k} F(s)ds} L(\tau) d\beta.$$

Then, we have

$$\begin{aligned} m_{k|k-1} &= \mathrm{E}[X(t_k)|Y_{k-1}] \\ &= e^{\int_{t_{k-1}}^{t_k} F(s)ds} m_{k-1|k-1} + \int_{t_{k-1}}^{t_k} e^{\int_{\tau}^{t_k} F(s)ds} u(\tau) d\tau \end{aligned}$$

and

$$\begin{aligned} P_{k|k-1} &= \mathrm{cov}[X(t_k), X(t_k)|Y_{k-1}] \\ &= e^{\int_{t_{k-1}}^{t_k} F(s)ds} P_{k-1|k-1} e^{\int_{t_{k-1}}^{t_k} F^T(s)ds} + \int_{t_{k-1}}^{t_k} e^{\int_{\tau}^{t_k} F(s)ds} L(\tau) Q_c(\tau) L^T(\tau) e^{\int_{\tau}^{t_k} F^T(s)ds} d\tau \end{aligned}$$

However,

$$A(t) = e^{\int_{t_{k-1}}^{t} F(s)ds}$$

, the state-transition matrix between t_k and t_{k-1}, is the solution to the following equation

$$\begin{aligned} \frac{dA(t)}{dt} &= F(t)A(t), \\ A(t_{k-1}) &= I. \end{aligned}$$

$$Q(t) = \int_{t_{k-1}}^{t_k} e^{\int_{\tau}^{t_k} F(s)ds} L(\tau) Q_c(\tau) L^T(\tau) e^{\int_{\tau}^{t_k} F^T(s)ds} d\tau$$

is the solution to the following equation:

$$\begin{aligned} \frac{dQ(t)}{dt} &= F(t)Q(t) + Q(t)F^T(t) + L(t)Q_c(t)L^T(t), \\ Q(t_{k-1}) &= 0, \end{aligned}$$

and

$$B(t) = \int_{t_{k-1}}^{t} e^{\int_{\tau}^{t} F(s)ds} u(\tau) d\tau$$

, the cumulative effect of inputs between t_{k-1} and t_k, is the solution to the following equation:

$$\frac{dB(t)}{dt} = F(t)B(t) + u(t).$$

The continuous – discrete Kalman filter can be expressed as follows:

prediction

$$\begin{aligned} m_{k|k-1} &= A(t_k)m_{k-1|k-1} + B(t_k) \\ P_{k|k-1} &= A(t_k)P_{k-1|k-1}A^T(t_k) + Q(t_{k-1}) \end{aligned}$$

filtering

$$\begin{aligned} m_{k|k} &= m_{k|k-1} + K_k(Y_k - C_k m_{k|k-1} - D_k u_k) \\ K_k &= P_{k|k-1}C_k^T[C_k P_{k|k-1}C_k^T + R_k]^{-1} \\ P_{k|k} &= (I - K_k C_k)P_{k|k-1} \end{aligned}$$

Due to the prevalence of uneven time steps in measurements, continuous modeling of biological networks is an important area of research, and I hope to move all of my research results in the discrete systems into the realm of continuous–discrete dynamic systems.

REFERENCES

[1] T Ideker, D.A. Lauffenburger, and R.L. Winslow, "Bioengineering and systems biology," *Ann. Biomed Eng.*, vol. 34, no. 2, pp. 257–264, 2006.

[2] C.F. Sing, J.H. Stengard, and S.L. Kardia, "Genes, environment, and cardiovascular disease," *Arterioscle Throbm Vasc Biol.*, vol. 23, no. 7, pp. 1190–1196, 2003.

[3] B.B. Aldridge, G. Haller, P.K. Sorger, and D.A. Lauffenburger, "Direct Lyapunov exponent analysis enables parametric study of transient signalling governing cell behaviour," *Systems Biology*, vol. 153, no. 6, pp. 425, 2006.

[4] E. Yanagisawa, S.-I. Azuma, and J.-I. Imura, "Controllability analysis for gene regulation networks and its application to luminescence bacterium," in *Proceedings of the First International Conference on Algebraic Biology*, 2005, vol. 1, pp. 51–60.

[5] M.R. Said, T.J. Begley, A.V. Oppenheim, D.A. Lauffenburger, and L.D. Samson, "Global network analysis of phenotypic effects: protain networks and toxicity modulation in saccharomyces cerevisiae," *PNAS*, vol. 101, pp. 18006–18011, 2004.

[6] E.O. Voit, *Compuational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists.* Cambridge : Cambridge University Press, 2000.

[7] M.T. Borisuk and J.J. Tyson, "Bifurcation analysis of a model of mitotic control in frog eggs," *J. Theor. Biol.*, vol. 195, pp. 69–85, 1998.

[8] A. Abbott, "Questions linger about unexplained gene-therapy trial death," *Nature Medicine*, vol. 12, pp. 597–597, 2006.

[9] J. Kaiser, "No easy answers in gene therapy death," *ScienceNOW*, vol. 2007, no. 917, pp. 143–143, 2007.

[10] P. Gennemark and D. Wedelin, "Efficient algorithms for ordinary differential equation model identification of biological systems," *IET Systems Biology*, vol. 1, no. 2, pp. 120–129, 2007.

[11] S.H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Reading, MA : Perseus Books, 1994.

[12] R. A. Monge, E. Roman, C. Nombela, and J. Pla, "The map kinase signal transduction network in candida albicans," *Microbiology*, vol. 152, no. Pt 4, pp. 905–912, 2006.

[13] B.B. Aldridge, J.M. Burke, D.A. Lauffenburger, and P.K. Sorger, "Physicochemical modeling of cell signalling pathways," *NATURE CELL BIOLOGY*, vol. 8, no. 11, pp. 1195–1203, November 2006.

[14] E. D. Sontag, "Molecular systems biology and control," *European Journal of Controj*, vol. 11, no. 4-5, pp. 396–435, 2005.

[15] J.L. DeRisi, V.R. Iyer, and P.O. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, vol. 278, pp. 680–686, 1997.

[16] SP Gygil, B. Rist, S.A. Gerber, F. Turecek, M.H. Gelb, and R. Aebersold, "Quantitative analysis of complex protein mixtures using isotope-coded affinity tags," *Nature Biotech.*, vol. 17, pp. 994–999, 1999.

[17] J.L. Griffin, C.J. Mann, J. Scott, et al., "Choline containing metabolites during cell transfection: an insight into magnetic resonance spectroscopy detectable changes," *FEBS Letters*, vol. 509, pp. 263–266, 2001.

[18] H. Zhou, J.D. Watts, and R. Aebersold, "A systematic approach to the analysis of protein phosphorylation," *Nature Biotech.*, vol. 19, pp. 375–378, 2001.

[19] U.B. Nielsen, M.H. Cardone, A.J. Sinskey, G. MacBeath, and P.K. Sorger, "Profiling receptor tyrosine kinase activation by using ab microarrays," *PNAS*, vol. 100, pp. 9330–9335, 2003.

[20] A.C. Haugen, R. Kelley, J.B. Collins, C.J. Tucker, C. Deng, C.A. Afshari, J.M. Brown, T. Ideker, and B. Van Houten, "Integrating phenotypic and expression profiles to map arsenic-response networks," *Genome Biol.*, vol. 5, pp. R95, 2004, DOI:10.1186/gb-2004-5-12-r95.

[21] R.F. Service, "Gene sequencing: The race for the $1000 genome," *Science*, vol. 311, no. 5767, pp. 1544–1546, 2006.

[22] R. Bashir, "Biomems: state-of-the-art in detection, opportunities and prospects," *Adv. Drug Delivery Reviews*, vol. 56, pp. 1565–1587, 2004.

[23] F.J. Bruggeman and H.V. Westerhoff, "The nature of systems biology," *Trends in Microbiology*, vol. 15, no. 1, pp. 45–50, 2006.

[24] T.S. Gardner and J.J. Faith, "Reverse-engineering transcription control networks," *Physics of Life Reviews*, vol. 2, no. 1, pp. 65–88, 2005.

[25] J.L. Snoep, "The Silicon Cell initiative: working towards a detailed kinetic description at the cellular level," *Curr. Opin. Biotechnol.*, vol. 16, pp. 336–343, Jun 2005.

[26] S. Bornholdt, "Systems biology. Less is more in modeling large genetic networks," *Science*, vol. 310, pp. 449–451, Oct 2005.

[27] T. Chen, H. L. He, and G. M. Church, "Modeling gene expression with differential equations," *Pac Symp Biocomput*, vol. 4, pp. 29–40, 1999.

[28] L. Glass and S. Kauffman, "The logical analysis of continuous, nonlinear biochemical control networks," *Journal of Theoretical Biology*, vol. 39, no. 1, pp. 103–129, 1973.

[29] S.A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*. New York : Oxford University Press, 1993.

[30] A. Cornish-Bowden, *Fundamentals of Enzyme Kinetics*, revised edition. London: Portland Press, 1995.

[31] R. Heinrich and S. Schuster, *The Regulation of Cellular Systems*. New York : Springer, 1996.

[32] B. C. Goodwin and S.A. Kauffman, *Cell to Cell Signaling: From Experiments to Theoretical Models*, chapter Bifurcations, harmonics, and the four color wheel model of Drosophila development, London : Academic Press, 1989.

[33] J. Griffith, "Mathematics of cellular control processes: I. negative feedback to one gene," *Journal of Theoretical Biology*, vol. 20, no. 2, pp. 202–208, 1968.

[34] R. Thomas and R.dÁri, *Biological Feedback*. Boca Raton, FL : CRC Press, 1990.

[35] R. Thomas, "On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations," in *Numeri-*

cal Methods in the Study of Critical Phenomena, J. Della Dora, J. Demongeot, and B. Lacolle, Eds. Berlin : Springer Verlag, 1981.

[36] E. Plahte, T. Mestl, and S.W. Omholt, "Feedback loops, stability and multistationarity in dynamical systems," *Journal of Biological Systems*, vol. 3, no. 2, pp. 409–413, 1995.

[37] S. E. Houssine, "Necessary conditions for multistationarity and stable periodicity," *Journal of Biological Systems*, vol. 6, no. 1, pp. 2403–2413, 1998.

[38] R. Thomas, "Laws for the dynamics of regulatory networks," *International Journal of Developmental Biology*, vol. 42, pp. 479–485, 1998.

[39] Y. Wang, C.L. Liu, J.D. Storey, R.J. Tibshirani, D. Herschlag, and P.O. Brown, "Precision and functional specificity in mRNA decay," *Proceedings of the National Academy of Sciences*, vol. 99, no. 9, pp. 5860–5865, 2002.

[40] M. Ronen, R. Rosenberg, B. I. Shraiman, and U. Alon, "Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics," *PNAS*, vol. 99, pp. 10555–10560, 2002.

[41] R.D. Bliss, P.R. Painter, and A.G. Marr, "Role of feedback inhibition in stabilizing the classical operon," *J. Theor. Biol.*, vol. 97, pp. 177–193, 1982.

[42] J. Reinitz and J.R. Vaisnys, "Theoretical and experimental analysis of the phage lambda genetic switch implies missing levels of co-operativity," *J. Theor. Biol.*, vol. 145, pp. 295–318, 1990.

[43] B. Novak and J.J. Tyson, "Numerical analysis of a comprehensive model of m-phase control xenopus oocyte extracts and intact embryos," *J. Cell Sci.*, vol. 106, pp. 1153–1168, 1993.

[44] B. Bunow, J.-P. Kernevez, G. Joly, and D.Thomas, "Pattern formation by reaction-diffusion instabilities: Application ot morphogenesis in drosophila," *Journal of Theoretical Biology*, vol. 84, pp. 629–649, 1980.

[45] T.E. Ideker and V. Thorsson, "Discovery of regulatory interactions through perturbation: inference and experimental design," in *Pacific Symposium on Biocomputing*, 2000, pp. 302–313.

[46] D.J. Irons and N.A. Monk, "Identifying dynamical modules from genetic regulatory systems: applications to the segment polarity network," *BMC Bioinformatics*, vol. 8, pp. 413, Oct 2007, doi: 10.1186/1471-2105-8-413.

[47] K. Willadsen and J. Wiles, "Robustness and state-space structure of Boolean gene regulatory models," *J. Theor. Biol.*, vol. 249, pp. 749–765, Dec 2007.

[48] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, "Genetic networks with canalyzing Boolean rules are always stable," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 101, pp. 17102–17107, Dec 2004.

[49] I. Shmulevich, E.R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, pp. 1319–1331, Oct 2002.

[50] I. Shmulevich, 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, Feb 2002.

[51] G. Sanguinetti, M. Rattray, and N. D. Lawrence, "A probabilistic dynamical model for quantitative inference of the regulatory mechanism of transcription," *Bioinformatics*, vol. 22, no. 14, pp. 1753–1759, 2006.

[52] M. Ronen, R. Rosenberg, B.I. Shraiman, and U. Alon, "Parameterizing a gene regulation network by using accurate expression kinetics," *PNAS*, vol. 99, no. 16, pp. 10555–10560, 8 2002.

[53] S. Rogers and M. Girolami, "A Bayesian regression approach to the inference of regulatory networks from gene expression data," *Bioinformatics*, vol. 21, no. 14, pp. 3131–3137, 2005.

[54] S. Kim, S. Imoto, and S. Miyand, "Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data," *Biosystems*, vol. 75, pp. 57–65, 2004.

[55] N. Friedman and D. Koller, "Being Bayesian about network structure. a Bayesian approach to structure discover in Bayesian networks," *Machine Learning*, vol. 50, no. 1-2, pp. 95–125, 1 2003.

[56] K. Murphy and S. Mian, "Modeling gene expression data using dynamic Bayesian networks," *Technical report CS Division UC Berkeley*, 1999.

[57] I. Nachman, A. Regev, and N. Friedman, "Inferring quantitative models of regulatory networks from expression data," *Bioinformatics*, vol. 20 Suppl 1, pp. 248–256, 2004.

[58] N. Friedman, K. Murphy, and S. Russel, "Learning the structure of dynamic probabilistic networks," in *Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 129–138.

[59] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model," *Pac Symp Biocomput*, vol. 4, pp. 17–28, 1999.

[60] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, "Random Boolean network models and the yeast transcriptional network," *Proc Natl Acad Sci USA*, vol. 100, no. 25, pp. 14796–9, 2003.

[61] P. D'Haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear modeling of mRNA expression levels during CNS development and injury," *Pac Symp Biocomput*, vol. 4, pp. 41–52, 1999.

[62] N. Dojer, A. Gambin, A. Mizera, B. Wilczynski, and J. Tiuryn, "Applying dynamic Bayesian networks to perturbed gene expression data," *BMC Bioinformatics*, vol. 7, pp. 249, 2006.

[63] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *J Computional Biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[64] D. Husmeier, "Reverse engineering of genetic networks with Bayesian networks," *Biochem Soc Trans*, vol. 31, no. Pt 6, pp. 1516–1518, 2003.

[65] K. Murphy and S. Mian, "Modelling gene expression data using dynamic Bayesian networks," Tech. Rep., University of California, 1999.

[66] M. J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild, "A Bayesian approach to reconstructing genetic regulatory networks with hidden factors," *Bioinformatics*, vol. 21, no. 3, pp. 349–356, 2005.

[67] Z. Li, S. M. Shaw, M. J. Yedwabnick, and C. Chan, "Using a state-space model with hidden variables to infer transcription factor activities," *Bioinformatics*, vol. 22, no. 6, pp. 747–754, 2006.

[68] D. Pe'er, A. Regev, G. Elidan, and N. Friedman, "Inferring subnetworks from perturbed expression profiles," *Bioinformatics*, vol. 17 Suppl 1, pp. S215–S224, 2001.

[69] F. X. Wu, W. J. Zhang, and A. J. Kusalik, "Modeling gene expression from microarray expression data with state-space equations," *Pac Symp Biocomput*, vol. 9, pp. 581–592, 2004.

[70] R. Yamaguchi and T. Higuchi, "State-space aproach with the maximum likelihood principle to identify the system generating time-course gene expression data of yeast," *International Journal of Data Mining and Bioinformatics*, vol. 1, no. 1, pp. 77–87, 2006.

[71] R. Yamaguchi, R. Yoshida, S. Imoto, T. Higuchi, and S. Miyano, "Finding module-based gene networks with state-space models," *IEEE Signal Processing Magazine*, vol. 24, no. 1, pp. 37–46, 2007.

[72] C. Rangel, J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. L. Wild, and F. Falciani, "Modeling t-cell activation using gene expression profiling and state-space models," *Bioinformatics*, vol. 20, no. 9, pp. 1361–1372, 2004.

[73] C. Rangel, J. Angus, Z. Ghahramani, and D.L. Wild, "Modeling genetic regulatory networks using gene expression profiling and state space models," in *Applications of Probabilistic Modelling in Medical Informatics and Bioinformatics*, D. Husmeier, S. Robert, and R. Dybowski, Eds. New York : Springer-Verlag, 2004, pp. 269–293.

[74] C. Rangel, D.L. Wild, F. Falciani, Z. Ghahramani, and A. Gaiba, "Modeling biological responses using gene expression profiling and linear dynamical sys-

tems," in *Proceedings of the 2nd International Conference on Systems Biology*. 2001, pp. 248–256, Omipress.

[75] V. R. Iyer, C. E. Horak, C. S. Scafe, D. Botstein, M. Snyder, and P. O. Brown, "Genomic binding sites of the yeast cell-cycle transcription factors sbf and mbf," *Nature*, vol. 409, no. 6819, pp. 533–538, 2001.

[76] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young, "Transcriptional regulatory networks in saccharomyces cerevisiae," *Science*, vol. 298, no. 5594, pp. 799–804, 2002.

[77] B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. L. Volkert, C. J. Wilson, S. P. Bell, and R. A. Young, "Genome-wide location and function of DNA binding proteins," *Science*, vol. 290, no. 5500, pp. 2306–2309, 2000.

[78] M. Welling and M. Weber, "A constrained EM algorithm for independent component analysis," *Neural Computation*, vol. 13, no. 3, pp. 677–689, 2001.

[79] J. H. Ahn and J. H. Oh, "A constrained EM algorithm for principal component analysis," *Neural Computation*, vol. 15, no. 1, pp. 57–65, 2003.

[80] L. S. Y. Wu, J. S. Pai, and J. R. M. Hosking, "An algorithm for estimating parameters of state-space models," *Statistics & Probability Letters*, vol. 28, no. 2, pp. 99–106, 1996.

[81] K. Murphy and S. Mian, "Dynamic Bayesian networks: Representation, inference and learning," Ph.D. dissertation, UC Berkeley, 2002.

[82] K. Lemmens, T. Dhollander, T. De Bie, P. Monsieurs, K. Engelen, B. Smets, J. Winderickx, B. De Moor, and K. Marchal, "Inferring transcriptional modules from chip-chip, motif and microarray data," *Genome Biol*, vol. 7, no. 5, pp. R37, 2006, doi:10.1186/gb-2006-7-5-r37.

[83] D. E. Zak, G. E. Gonye, J. S. Schwaber, and 3rd Doyle, F. J., "Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: insights from an identifiability analysis of an in silico network," *Genome Res*, vol. 13, no. 11, pp. 2396–2405, 2003.

[84] C.-T. Chen, *Linear System Theory and Design*, 3rd edition. The Oxford series in electrical and computer engineering. New York : Oxford University Press, 1999.

[85] F. d'Alch Buc, P.-J. Lahaye, B.-E. Perrin, L. Ralaviola, T. Vujasinovic, A. Mazurie, and S. Bottani, "A dynamic model of gene regulatory networks based on inertia principle," in *Bioinformatics Using Computational Intelligence Paradigms*, Udo Seiffert, Lakhmi C. Jain, and Patrick Schweizer, Eds. Berlin : Springer, 2005, pp. 93–118.

[86] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[87] Z. Ghahramani and G.E. Hinton, "Parameter estimation for linear dynamical systems," Tech. Rep., University of Toronto, 1996.

[88] S. Gibson and B. Ninness, "Robust maximum-likelihood estimation of multivariable dynamic systems," *Automatica*, vol. 41, no. 10, pp. 1667–1682, 2005.

[89] R.H. Shumway and D.S. Stoffer, "An approach to time series smoothing and forecasting using the em algorithm," *Journal of Time Series Analysis*, vol. 3, no. 4, pp. 253–264, 1982.

[90] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Prentice Hall Information and System Sciences Series. Upper Saddle River, NJ : Prentice Hall, 2000.

[91] S. Gibson and B. Ninness, "Robust maximum-likelihood estimation of multivariable dynamic systems," *Automatica*, vol. 41, no. 10, pp. 1667–1682, 2005.

[92] M. Ronen, R. Rosenberg, B. I. Shraiman, and U. Alon, "Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics," *Proc Natl Acad Sci USA*, vol. 99, no. 16, pp. 10555–10560, 2002.

[93] F. M. Camas, J. Blazquez, and J. F. Poyatos, "Autogenous and nonautogenous control of response in a genetic network," *Proc Natl Acad Sci USA*, vol. 103, no. 34, pp. 12718–12723, 2006.

[94] N. Friedman, S. Vardi, M. Ronen, U. Alon, and J. Stavans, "Precise temporal modulation in the response of the sos dna repair network in individual bacteria," *PLoS Biology*, vol. 3, no. 7, pp. e238, 2005.

[95] R. Strohman, "Maneuvering in the complex path from genotype to phenotype," *Science*, vol. 296, no. 5568, pp. 701–703, 2002.

[96] O. Wolkenhauer and M. Mesarovic, "Feedback dynamics and cell function: Why systems biology is called systems biology," *Mol Biosyst*, vol. 1, no. 1, pp. 14–16, 2005.

[97] J. Herrero, J. M. Vaquerizas, F. Al-Shahrour, L. Conde, A. Mateos, J. S. Diaz-Uriarte, and J. Dopazo, "New challenges in gene expression data analysis and the extended gepas," *Nucleic Acids Res*, vol. 32, no. Web Server issue, pp. W485–491, 2004.

[98] N. Rosenfeld and U. Alon, "Response delays and the structure of transcription networks," *J Mol Biol*, vol. 329, no. 4, pp. 645–654, 2003.

[99] N. Rosenfeld, M. B. Elowitz, and U. Alon, "Negative autoregulation speeds the response times of transcription networks," *J Mol Biol*, vol. 323, no. 5, pp. 785–793, 2002.

[100] R. J. Prill, P. A. Iglesias, and A. Levchenko, "Dynamic properties of network motifs contribute to biological network organization," *PLoS Biol*, vol. 3, no. 11, pp. e343, 2005.

[101] V. de Lorenzo, L. Serrano, and A. Valencia, "Synthetic biology: challenges ahead," *Bioinformatics*, vol. 22, no. 2, pp. 127–128, 2006.

[102] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ : Prentice-Hall, Inc., 1980.

[103] S. Martin, Z. Zhang, A. Martino, and J. L. Faulon, "Boolean dynamics of genetic regulatory networks inferred from microarray time series data," *Bioinformatics*, vol. 23, no. 7, pp. 866–874, 2007.

[104] Z. Szallasi and S. Liang, "Modeling the normal and neoplastic cell cycle with realistic Boolean genetic networks: their application for understanding carcino-

genesis and assessing therapeutic strategies," *Pac Symp Biocomput*, pp. 66–76, 1998.

[105] S. Kimura, K. Ide, A. Kashihara, M. Kano, M. Hatakeyama, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, and A. Konagaya, "Inference of s-system models of genetic networks using a cooperative coevolutionary algorithm," *Bioinformatics*, vol. 21, no. 7, pp. 1154–1163, 2005.

[106] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita, "Dynamic modeling of genetic networks using genetic algorithm and s-system," *Bioinformatics*, vol. 19, no. 5, pp. 643–650, 2003.

[107] R. Yamaguchi and T. Higuchi, "State-space aproach with the maximum likelihood principle to identify the system generating time-course gene expression data of yeast," *International Journal of Data Mining and Bioinformatics*, vol. 1, pp. 77–87, 2006.

[108] B.M. Broom, T.J. McDonnell, and D. Subramanian, "Predicting altered pathways using extendable scaffolds," *International Journal of Bioinformatics Research and Applications*, vol. 2, no. 1, pp. 3–18, 2006.

[109] A. M. Sciuto, C. S. Phillips, L. D. Orzolek, A. I. Hege, T. S. Moran, and 3rd Dillman, J. F., "Genomic analysis of murine pulmonary tissue following carbonyl chloride inhalation," *Chem Res Toxicol*, vol. 18, no. 11, pp. 1654–1660, 2005.

[110] N. Yoshizuka, Y. Yoshizuka-Chadani, V. Krishnan, and S.L. Zeichner, "Human immunodeficiency virus type 1 Vpr-dependent cell cycle arrest through a mitogen-activated protein kinase signal transduction pathway," *J. Virol*, vol. 79, no. 17, pp. 11366–11381, 2005.

[111] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 10th edition. Upper Saddle River, NJ : Pearson Prentice Hall, 2005.

[112] K. Zhou and J. C. Doyle, *Essentials of Robust Control*. Upper Saddle River, NJ : Prentice Hall, 1998.

[113] S. P. Bhattacharyya and L. H. Keel, *Control of Uncertain Dynamic Systems*. Boca Raton, FL : CRC Press, 1991.

[114] H. Kitano, "Opinon - cancer as a robust system: implications for anticancer therapy," *Nature Reviews Cancer*, vol. 4, no. 3, pp. 227–235, 2004.

[115] E. Hornstein and N. Shomron, "Canalization of development by micrornas," *Nat Genet*, vol. 38 Suppl, pp. S20–24, 2006.

[116] H. Kitano, "Biological robustness," *Nature Reviews Genetics*, vol. 5, no. 11, pp. 826–837, 2004.

[117] H. Kitano, "Biological robustness in complex host-pathogen systems," *Prog Drug Res*, vol. 64, pp. 239, 241–263, 2007.

[118] F. Ding and T. Chen, "Hierarchical least squares identification methods for multivariable systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 397–402, 2005.

[119] O. Wolkenhauer, "Defining systems biology: an engineering perspective," *IET Syst Biol*, vol. 1, no. 4, pp. 204–206, 2007.

[120] R. Chakrabarti, H. Rabitz, S.L. Springs, and G.L. McLendon, "Mutagenic evidence for the optimal control of evolutionary dynamics," *Physical Review Letters*, vol. 100, no. 25, pp. 258103, 2008, doi:10.1103/PhysRevLett.100.258103.

[121] O. Slaby, S. Sager, O.S. Shaik, U. Kummer, and D. Lebiedz, "Optimal control of self-organized dynamics in cellular signal transduction," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 13, no. 5, pp. 487–502, 2007.

[122] B. Faryabi, G. Vahedi, J.F. Chamberland, A. Datta, and E.R. Dougherty, "Optimal constrained stationary intervention in gene regulatory networks," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2008, 2008.

[123] J Karrakchou, M Rachik, and S Gourari, "Optimal control and infectiology : Application to an HIV/AIDS model," *Applied Mathematics and Computation*, vol. 177, pp. 807–818, 2006.

[124] R. Pal, A. Datta, and E.R. Dougherty, "Optimal infinite-horizon control for probabilistic Boolean networks," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2375–2387, 2006.

[125] M. Joly and J. M. Pinto, "Role of mathematical modeling on the optimal control of hiv-1 pathogenesis," *AIChE Journal*, vol. 52, no. 3, pp. 856–884, 2005.

[126] Y. Liu, H. Bin Sun, and H. Yokota, "Regulating gene expression using optimal control theory," in *Proc. 3rd. Sym. Bioinformatics and Bioeng.* IEEE, 2003, vol. 1, pp. 313–318.

[127] ZG Goldsmith and N. Dhanasekaran, "The microrevolution: applications and impacts of microarray technology on molecular biology and medicine (review)," *Int J Mol Med*, vol. 13, no. 4, pp. 483–495, 2004.

[128] E.C. Gunther, D.J. Stone, R.W. Gerwien, P. Bento, and M.P. Heyes, "Prediction of clinical drug efficacy by classification of drug-induced genomic expression

profiles in vitro," *Proceedings of the National Academy of Sciences*, vol. 100, no. 16, pp. 9608–9613, 2003.

[129] P.C.Y. Chen and J.W. Chen, "Probabilistic control of gene networks," *Engineering in Medicine and Biology Society, 27th Annual International Conference of the IEEE-EMBS*, pp. 2966–2970, 2005.

[130] I. Shmulevich, E.R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[131] I. Shmulevich, E.R. Dougherty, and W. Zhang, "Control of stationary behavior in probabilistic Boolean networks by means of structural intervention," *Journal of Biological Systems*, vol. 10, no. 4, pp. 431–445, 2002.

[132] I. Shmulevich and E.R. Dougherty, *Genomic Signal Processing (Princeton Series in Applied Mathematics)*. Princeton,NJ : Princeton University Press Princeton, 2007.

[133] A. Datta, A. Choudhary, M.L. Bittner, and E.R. Dougherty, "External Control in Markovian Genetic Regulatory Networks," *Machine Learning*, vol. 52, no. 1, pp. 169–191, 2003.

[134] J. Hasty, D. McMillen, and J.J. Collins, "Engineered gene circuits," *Nature*, vol. 420, no. 6912, pp. 224–230, 2002.

[135] J.J. Tyson and H.G. Othmer, "The dynamics of feedback control circuits in biochemical pathways," *Progress in Theoretical Biology*, vol. 5, pp. 1–62, 1978.

[136] E. Plahte, T. Mestl, and S.W. Omholt, "A methodological basis for description and analysis of systems with complex switch-like interactions," *Journal of*

Mathematical Biology, vol. 36, no. 4, pp. 321–348, 1998.

[137] A. Arkin, J. Ross, and H.H. McAdams, "Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected escherichia coli cells," *Genetics*, vol. 149, no. 4, pp. 1633–1648, 1998.

[138] G. Siouris, *An Engineering Approach to Optimal Control and Parameter Theory*. New York : John Wiley, 1996.

[139] P. Ganju and J. Hall, "Potential applications of siRNA for pain therapy," *Expert Opinion on Biological Therapy*, vol. 4, no. 4, pp. 531–542, 2004.

[140] M.J. Wood, B. Trulzsch, A. Abdelgany, and D. Beeson, "Ribozymes and siRNA for the treatment of diseases of the nervous system.," *Curr Opin Mol Ther*, vol. 5, no. 4, pp. 383–388, 2003.

[141] M.E. Wall, W.S. Hlavacek, and M.A. Savageau, "Design of gene circuits: lessons from bacteria," *Nature Reviews Genetics*, vol. 5, no. 1, pp. 34–42, 2004.

[142] C.C. Guet, M.B. Elowitz, W. Hsing, and S. Leibler, "Combinatorial Synthesis of Genetic Networks," *Science*, vol. 296, no. 5572, pp. 1466–1470, 2002.

[143] D. Ferber, "Synthetic biology: Time for a synthetic biology asilomar?," *Science*, vol. 303, no. 5655, pp. 159–159, 2004.

[144] I.H. Tarner, U. Müller-Ladner, and C.G. Fathman, "Targeted gene therapy: frontiers in the development of smart drugs," *Trends in Biotechnology*, vol. 22, no. 6, pp. 304–310, 2004.

[145] W.E. Bunney, B.G. Bunney, M.P. Vawter, H. Tomita, J. Li, S.J. Evans, P.V. Choudary, R.M. Myers, E.G. Jones, S.J. Watson, et al., "Microarray Technology: A Review of New Strategies to Discover Candidate Vulnerability Genes

in Psychiatric Disorders," *American Journal of Psychiatry*, vol. 160, no. 4, pp. 657–666, 2003.

[146] D.R. Bentley, "Genomes for medicine," *Nature*, vol. 429, pp. 440–445, 2004.

[147] R.L. Strausberg, A.J.G. Simpson, L.J. Old, and G.J. Riggins, "Oncogenomics and the development of new cancer therapies," *Nature*, vol. 429, pp. 469–474, 2004.

[148] L. Wu, M. Johnson, and M. Sato, "Transcriptionally targeted gene therapy to detect and treat cancer," *Trends in Molecular Medicine*, vol. 9, no. 10, pp. 421–429, 2003.

[149] F. L. Lewis and V. L. Syrmos, *Optimal Control*. New York : John Wiley & Sons, Inc., 1995.

[150] K. Ogata, *Modern Control Engineering*. Upper Saddle River, NJ : Prentice-Hall, Inc., 1996.

[151] S.M. Shinners, *Advanced Modern Control System Theory and Design*. New York : John Wiley & Sons, Inc., 1998.

[152] A. E. Bryson, Jr. and Y.C. Ho, *Applied Optimal Control*. Washington : Hemisphere Pub. Corp., 1975.

[153] H.N. Claman, R.C. Giorno, and J.R. Seibold, "Endothelial and fibroblastic activation in scleroderma. the myth of the" uninvolved skin"," *Arthritis Rheum*, vol. 34, no. 12, pp. 1495–1501, 1991.

[154] R. Alba-Florest and E. Barbieri, "Real-time infinite horizon linear-quadratic tracking controller for vibration quenching in flexible beams," in *IEEE International Conference on Systems, Man and Cybernetics*, 2006, pp. 38–43.

[155] C. Rangel, J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. L. Wild, and F. Falciani, "Modeling t-cell activation using gene expression profiling and state-space models," *Bioinformatics*, vol. 20, no. 9, pp. 1361–1372, 2004.

[156] H. de Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 67–103, 2002.

[157] K.J. Kauffman, P. Prakash, and J.S. Edwards, "Advances in flux balance analysis," *Current Opinion in Biotechnology*, vol. 14, no. 5, pp. 491–496, 2003.

[158] C.H. Schilling, J.S. Edwards, D. Letscher, and B.O. Palsson, "Combining pathway analysis with flux balance analysis for the comprehensive study of metabolic systems," *Biotechnology and Bioengineering*, vol. 71, no. 4, pp. 286–306, 2000.

[159] A. Varma and B.O. Palsson, "Metabolic flux balancing: Basic concepts, scientific and practical use," *Nature Biotechnology*, vol. 12, no. 10, pp. 994–998, 1994.

[160] J.R. Koza, W. Mydlowec, G. Lanza, J. Yu, and M.A. Keane, "Reverse engineering of metabolic pathways from observed data using genetic programming.," *Pac Symp Biocomput*, vol. 6, pp. 434–445, 2001.

[161] V. Hatzimanikatis, "Nonlinear metabolic control analysis," *Metabolic Engineering*, vol. 1, no. 1, pp. 75–87, 1999.

[162] Z.P. Gerdtzen, P. Daoutidis, and W.S. Hu, "Non-linear reduction for kinetic models of metabolic reaction networks," *Metabolic Engineering*, vol. 6, no. 2, pp. 140–154, 2004.

[163] O. R. Gonzalez, C. Kuper, K. Jung, Jr. Naval, P. C., and E. Mendoza, "Parameter estimation using simulated annealing for s-system models of biochemical networks," *Bioinformatics*, vol. 23, no. 4, pp. 480–486, 2007.

[164] Z. Li, S. M. Shaw, M. J. Yedwabnick, and C. Chan, "Using a state-space model with hidden variables to infer transcription factor activities," *Bioinformatics*, vol. 22, no. 6, pp. 747–754, 2006.

[165] T.B. Schön, A. Wills, and B. Ninness, "Maximum likelihood nonlinear system estimation," *Proceedings of the 14th IFAC Symposium on System Identification*, pp. 1003–1008, 2006.

[166] J.H. Kotecha and P.M. Djuric, "Gaussian sum particle filtering," *Signal Processing, IEEE Transactions on*, vol. 51, no. 10, pp. 2602–2612, 2003.

[167] E. Haseltine and J. Rawlings, "Critical evaluation of extended Kalman filtering and moving-horizon estimation," *Ind. Eng. Chem. Res.*, vol. 44, no. 8, pp. 2451–2460, 2004.

[168] M. Quach and N. Brunel, "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.

[169] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and S.A. Adelaide, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, 2002.

[170] N. Cui, L. Hong, and J.R. Layne, "A comparison of nonlinear filtering approaches with an application to ground target tracking," *Signal Processing*,

vol. 85, no. 8, pp. 1469–1492, 2005.

[171] A. Wills, T. Schon, and B. Ninness, "Parameter estimation for discrete-time nonlinear systems using EM," in *Proceedings of the 17th IFAC World Congress, Seoul, Korea*, Jul 2008, pp. 1–6.

[172] I. Swameye, T.G. Muller, J. Timmer, O. Sandra, and U. Klingmuller, "Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by databased modeling," *Proceedings of the National Academy of Sciences*, vol. 100, no. 3, pp. 1028–1033, 2003.

[173] J.M. Mahaffy, "Cellular control models with linked positive and negative feedback and delays: I. the models," *J. Math. Biol.*, vol. 106, pp. 89–102, 1984.

[174] C.H. Lien, K.W. Yu, J.S. Lin, and M.L. Hung, "Exponential estimation of generalized state-space time-delay systems," in *Journal of Physics: Conference Series*. 2008, vol. 96, p. 012100, Institute of Physics Publishing, doi: 10.1088/1742-6596/96/1/012100.

[175] Z. Gao and S.X. Ding, "State and disturbance estimator for time-delay systems with application to fault estimation and signal compensation," *Signal Processing, IEEE Transactions on*, vol. 55, no. 12, pp. 5541–5551, 2007.

[176] Y. Tang and X. Guan, "Parameter estimation for time-delay chaotic system by particle swarm optimization," *Chaos, Solitons and Fractals*, vol. 34, no. 2, pp. 654–661, 2007.

[177] G. Sanguinetti, M. Rattray, and N. D. Lawrence, "A probabilistic dynamical model for quantitative inference of the regulatory mechanism of transcription," *Bioinformatics*, vol. 22, no. 14, pp. 1753–1759, 2006.

[178] K. C. Kao, Y. L. Yang, R. Boscolo, C. Sabatti, V. Roychowdhury, and J. C. Liao, "Transcriptome-based determination of multiple transcription regulator activities in Escherichia coli by using network component analysis," *Proc Natl Acad Sci USA*, vol. 101, no. 2, pp. 641–646, 2004.

[179] L. Ljung, "Identification of nonlinear systems," in *The Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision 2006*, 2006, pp. 1–10.

[180] S. C. Wang, "Reconstructing genetic networks from time ordered gene expression data using Bayesian method with global search algorithm," *J Bioinform Comput Biol*, vol. 2, no. 3, pp. 441–458, 2004.

[181] J. Durbin and S.J. Koopman, *Time Series Analysis by State Space Methods.* Oxford ; New York : Oxford University Press, 2001.

[182] A.H. Jazwinski, *Stochastic Processes and Filtering Theory.* New York : Academic Press, 1970.

[183] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions on Circuits and Systems*, vol. 25, no. 9, pp. 772–781, 1978.

[184] E. Bodewig, *Matrix Calculus.* New York : Interscience Publishers, 1959.

[185] A. Graham, *Kronecker Products and Matrix Calculus.* New York : Ellis Horwood Ltd, 1981.

[186] J.R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics.* Chichester [England] ; New York : Wiley Series in Proba. Math. Stat, 1988.

[187] W.J. Vetter, "Matrix calculus operations and Taylor expansions," *SIAM Review*, vol. 15, no. 2, pp. 352–369, 1973.

[188] D.A. Turkington, *Matrix Calculus and Zero-One Matrices: Statistical and Econometric Applications*. Cambridge ; New York : Cambridge University Press, 2002.

[189] C.D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia : Society for Industrial Mathematics, 2000.

[190] R. A. Horn, *Topics in Matrix Analysis*, 1 edition. New York : Cambridge University Press, 1991.

[191] R. Courant and F. John, *Introduction to Calculus and Analysis*. New York : Springer, 1999.

[192] T.P. Minka, "Expectation-maximization as lower bound maximization," *Tutorial published on the web at http://www-white. media. mit. edu/tpminka/papers/em. html*, 1998.

[193] R.K. Olsson, K.B. Petersen, and T. Lehn-Schioler, "State-space models: from the EM algorithm to a gradient approach," *Neural Computation*, vol. 19, no. 4, pp. 1097–1111, 2007.

[194] R.E. Kalman and R.S. Bucy, "New Results in Linear Filtering and Prediction Theory.," *Tans. ASME J. Basic Eng.*, pp. 95–108, 1961.

APPENDIX A

MATRIX CALCULUS

In this appendix, some of basic matrix calculus definitions and theorems utilized in the dissertation are listed so that the reader does not need to hunt for them in other books and papers. No attempt at thoroughness is made, nor are we concerned with mathematical rigor. For those who are interested in more detailed treatment, these papers and books, [183, 184, 185], are recommended. Magnus and Neudecker [186] is a book that demands considerable mathematical maturity, because the text is concise and skips steps in proofs and derivations, but it is invaluable as a reference for its reasonable completeness.

Before we start I should say a word about the definition of derivative for vectors and matrices. The truth is there is no consensus about them. Derivative of a scalar with respect to vectors can be either a column vector as in [149] or a row vector as in [152], both of which enjoy popularity in the literature. But at least the difference is a mere transpose. It is much worse for matrices. Although no disagreement exists about the derivative of a scalar with respect to a matrix (except possibly a transpose), several definitions of the derivative of a vector with respect to a matrix (see [187] for an early summary and [188] for a more recent book), the equivalence of them being far more complicated than mere transpose. The disagreement is about the arrangement of partial derivatives as a matrix. Without getting into the strength and weakness of the respective approaches, I will here only state the reason for the particular choice I made.

The main reason I chose the differential approach to matrix calculus is because this presents an easy-to-remember chain rule and a product rule. We can all recall the chain rule and product rule in single-variable calculus as simple and yet powerful and fundamental to taking derivatives. I wanted to have something similar in matrix calculus. So when each of the other definitions has slightly or not so slightly different chain rules and product rules for different functions, the clarity and elegance of the differential approach become attractive.

This appendix is organized to start with elementary vectors and matrices, which are then used in proving some properties of vec and Kronecker product. With these indispensable tools thus defined, we then proceed to define the differential of a function and associated derivatives. As examples, differentials and derivatives of common functions such as traces, determinants, matrix inverses are provided. Finally, a formula for the Hessian matrix of vector-valued functions is provided.

A vector that has zeroes everywhere except the ith element which is equal to 1 is called an elementary vector, i.e.,

$$e_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} .$$

We usually denote an elementary vector by a subscripted lower-case e_i. An elementary matrix is a square matrix that also has only one element equal to1 and every other element equal to zero. We can also define elementary matrix as a product of elementary vectors as

$$E_{ij} = e_i \times e_j^T ,$$

a matrix whose entries are all zero except the ijth element, which is equal to one.

Some immediate and obvious properties of elementary vector and elementary matrix are

$$\delta_{ij} = e_i^T e_j, \tag{A.1}$$

$$E_{ij}E_{kl} = \delta_{jk}E_{il}, \tag{A.2}$$

$$A = \sum_{ij} a_{ij}E_{ij}, \tag{A.3}$$

$$A_{i\cdot} = e_i^T A, \tag{A.4}$$

$$A_{\cdot j} = Ae_j, \tag{A.5}$$

where δ_{ij} is the Kronecker delta function (the Kronecker delta function returns 1 if $i = j$, otherwise zero), and A is any matrix whose ith row is represented by $A_{i\cdot}$ and jth column by $A_{\cdot j}$. Replacing matrix A with matrix product BC in (A.3) (A.4)(A.5) results in the element-wise view, the row view, and column view of matrix product:

1. the ijth element of BC is the product of the ith row of B and jth column of C;

2. the ith row of BC is C multiplied by the ith row of B;

3. the jth column of BC is equal to B multiplied by the jth column of C.

Not as commonly known is a forth view of matrix product,

$$BC = \sum_i B_{\cdot i}C_{i\cdot}, \tag{A.6}$$

i.e., sum of corresponding row and column products. This involves rearranging terms, but a direct proof is unintuitive, so, instead, we can use the fact that an identity matrix

$$I = \sum_i e_i e_i^T$$

to get

$$\begin{aligned}BC = BIC &= B(\sum_i e_i e_i^T)C \\ &= \sum_i B e_i e_i^T C \\ &= \sum_i (Be_i)(e_i^T C) \\ &= \sum_i B_{\cdot i} C_{i\cdot}.\end{aligned} \tag{A.7}$$

Elementary vector and matrix are versatile and powerful tools (for its use in LU decomposition see [189]) , but for the sake of space we will stop here.

Kronecker operation and vec product are related and are primarily used for rearranging entries of matrices and vectors. We need it primarily to rearrange a matrix into a vector, and others use it to rearrange the derivative operator. They are more useful than one at first supposes.

Given a matrix A,

$$\operatorname{vec} A = \begin{bmatrix} A_{\cdot 1} \\ A_{\cdot 2} \\ \vdots \end{bmatrix},$$

that is, $\operatorname{vec} A$ is a vector with columns of A (represented by $A_{\cdot i}$) stacked in order. It is clear vec is a linear operator, that is, it is closed under addition and scalar multiplication. Kronecker product, on the other hand, requires two matrices, say B and C, so that $B \otimes C$ is equal to

$$\begin{bmatrix} b_{11}C & b_{12}C & \ldots \\ \vdots & \ddots & \vdots \end{bmatrix},$$

that is, each element of B multiplied by C. It is clear that the Kronecker product is defined for any two matrices, unlike the regular matrix product which requires comforming dimensions. Kronecker product is associative and distributive with respective to matrix sum:

$$(A + B) \otimes C = A \otimes C + B \otimes C,$$

$$A \otimes (B + C) = A \otimes B + A \otimes C,$$

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C.$$

The so-called mixed product rule is fundamental to most of properties associated with Kronecker product,

$$(A \otimes B)(C \otimes D) = AC \otimes BD.$$

Its proof is tedious but straightforward, just multiplying both sides to find the ijth element to be equal. An immediate consequence of mixed product rule is the inverse of Kronecker product,

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1},$$

assuming the inverse exists, which by this formula is same as A and B are invertible. The proof is one line

$$(A \otimes B)(A^{-1} \otimes B^{-1}) = AA^{-1} \otimes BB^{-1} = I.$$

The relation between vec and Kronecker product can be seen at the most basic level when matrices are degenerate and in fact are vectors. Let x and y be vectors of length greater than 1,

$$xy^T = x \otimes y^T = y^T \otimes x \tag{A.8}$$

$$\operatorname{vec} xy^T = y \otimes x. \tag{A.9}$$

With these and elementary vector and matrix operations we can prove succinctly the following important formula:

$$\operatorname{vec} AXB = (B^T \otimes A) \operatorname{vec} X, \tag{A.10}$$

because

$$\begin{aligned}
\operatorname{vec} AXY &= \operatorname{vec} A(\sum_{ij} x_{ij} E_{ij})B \\
&= \operatorname{vec} \sum_{ij} x_{ij} A e_i e_j^T B \\
&= \sum_{ij} x_{ij} \operatorname{vec}[(Ae_i)(e_j^T B)] \qquad \text{by linearity of } \operatorname{vec} \text{ operation} \\
&= \sum_{ij} x_{ij} (e_j^T B)^T \otimes (Ae_i) \qquad \text{by equation (A.9)} \\
&= \sum_{ij} x_{ij} (B^T \otimes A)(e_j \otimes e_i) \qquad \text{by mixed product rule} \\
&= (B^T \otimes A) \sum_{ij} x_{ij} e_j \otimes e_i \\
&= (B^T \otimes A) \sum_{ij} x_{ij} \operatorname{vec} e_i e_j^T \\
&= (B^T \otimes A) \operatorname{vec} \sum_{ij} x_{ij} e_i e_j^T \\
&= (B^T \otimes A) \operatorname{vec} \sum_{ij} x_{ij} E_{ij} \\
&= (B^T \otimes A) \operatorname{vec} X,
\end{aligned}$$

assuming all matrix products involving comforming matrices.

The last useful formula we will list in this section is the vec version of trace,

$$\operatorname{tr} AB = (\operatorname{vec} A^T)^T \operatorname{vec} B.$$

This can be proved by noting that trace of a matrix can be written as

$$\sum_i e_i^T A e_i,$$

and therefore we can write

$$\begin{aligned}\operatorname{tr} AB &= \sum_i e_i^T ABe_i \\ &= \sum_i e_i^T AIBe_i \\ &= \sum_i e_i^T A(\sum_j e_j e_j^T)Be_i \\ &= \sum_{ij} e_i^T Ae_j(e_j^T Be_j) \\ &= \sum_{ij} [A]_{ij}[B]_{ji},\end{aligned}$$

the last term above is the sum of products of corresponding elements of B with A transposed, which can be written as a dot product of two vectors $(\operatorname{vec} A^T)^T$ and $\operatorname{vec} B$.

There are a lot more about Kronecker product and vec operation we are not covering here, in particular the eigenvalues and eigenvectors of Kronecker product, which are very important. Please see Brewer's excellent paper [183] for a summary.

This section only covers derivative of matrix and neglects integration, and then only the very basics of derivative, mostly definitions with a few examples.

First we must make a note about notations, especially since notations tend to be confusing in matrix calculus. We denote a differential by $\mathrm{d}\, X$, and derivative of a vector with respect to another vector as

$$\frac{\mathrm{d}\, Y}{\mathrm{d}\, X^T} \quad \text{derivative of a column vector with respect to a row vector,} \tag{A.11}$$

$$\frac{\mathrm{d}\, Y^T}{\mathrm{d}\, X} \quad \text{derivative of a row vector with respect to a column vector.} \tag{A.12}$$

An immediate consequence of above definition is that

$$\left(\frac{\mathrm{d}\,Y}{\mathrm{d}\,X^T}\right)^T = \frac{\mathrm{d}\,Y^T}{\mathrm{d}\,X},$$

the transpose of derivative is equal to the derivative of both dependent and independent variables transposed, and that if

$$\mathrm{d}\,Y = f(X)\mathrm{d}\,X,$$

then the derivative is

$$\frac{\mathrm{d}\,Y}{\mathrm{d}\,X^T} = f(X).$$

Now let us define differentials. The differential of a function of vectors or matrices, $f(X)$, is the linear part of $f(X + \mathrm{d}\,X) - f(X)$, assuming the higher order terms vanish when divided by $\mathrm{d}\,X$ and $\mathrm{d}\,X \to 0$. This is a case of definition by Taylor series. The differentials of some common functions are listed here:

$$\mathrm{d}\,A = 0 \qquad A \text{ is constant} \tag{A.13}$$

$$\mathrm{d}\,\alpha X = \alpha \mathrm{d}\,X \qquad \alpha \text{ is a scalar} \tag{A.14}$$

$$\mathrm{d}\,\mathrm{tr}\,X = \mathrm{tr}[\mathrm{d}\,X] \tag{A.15}$$

$$\mathrm{d}\,XY = X\mathrm{d}\,Y + \mathrm{d}\,XY \qquad \text{product rule} \tag{A.16}$$

$$\mathrm{d}\,X^{-1} = -X^{-1}(\mathrm{d}\,X)X^{-1} \tag{A.17}$$

$$\mathrm{d}\,|X| = |X|\,\mathrm{tr}(X^{-1}\mathrm{d}\,X). \tag{A.18}$$

The rule for inverse is simple to prove:

$$0 = \mathrm{d}\,I = \mathrm{d}\,XX^{-1} = (\mathrm{d}\,X^{-1})X + X^{-1}\mathrm{d}\,X,$$

and equation (A.17) follows by rearranging terms. The differential of a determinant is a little more complicated to prove.

Here we shall use a definition of determinant that is not usually taught at undergraduate level but is nevertheless very useful. The definition is mentioned in [190] along with other definitions, and a particularly good exposition of it can be found in Courant's venerable book [191]. We shall define the determinant as a multi-linear, alternating multivariable function. The multi-linear part is easy to understand: $f(x, y, z)$ is multi-linear function is $f(\cdot)$ is linear in one variable while other variables are fixed. An alternating function is a function of at least two variables, say $g(x, y)$, that switches sign when two variables are switched, that is $g(x, y) = -g(y, x)$. An immediate consequence is that $g(x, x) = 0$ because $g(x, x) = -g(x, x)$, but we digress. The point is we write the determinant of X as a multi-linear, alternating function of its columns (rows work too). The differential of a determinant then is

$$\begin{aligned}
\mathrm{d}\,|X| &= \mathrm{d}\,f(X_{\cdot 1}, \ldots, X_{\cdot n}) \\
&= \text{ the linear part of } f(X_{\cdot 1} + \mathrm{d}\,X_{\cdot 1}, \ldots, X_{\cdot n} + \mathrm{d}\,X_{\cdot n}) \\
&= \sum_i f(X_{\cdot 1}, \ldots, \mathrm{d}\,X_{\cdot n}, \ldots, X_{\cdot n}) \\
&= f(X) \sum_i \frac{f(X_{\cdot 1}, \ldots, \mathrm{d}\,X_{\cdot n}, \ldots, X_{\cdot n})}{f(X)},
\end{aligned}$$

and by Cramer's rule,

$$\frac{f(X_{\cdot 1}, \ldots, \mathrm{d}\,X_{\cdot n}, \ldots, X_{\cdot n})}{f(X)}$$

is the *ith* diagonal element of $X^{-1}\mathrm{d}\,X$, so it continues

$$= f(X)\,\mathrm{tr}[X^{-1}\mathrm{d}\,X] = |X|\,\mathrm{tr}[X^{-1}\mathrm{d}\,X]. \tag{A.19}$$

The rules and formulas in this section can be used to find gradiants and Jacobian matrix. But if we have to deal with higher order derivatives, it is slightly more complicated. The second order differential of a function can be defined as the differential of the first order differential. Likewise, the Hessian matrix packs all the second order derivatives of a function, as the Jacobian matrix packs all the first order derivatives.

First, the second order differential of scalar functions. Because the first order differential of a scalar function is still a scalar function, the second order differential of a scalar function is also a scalar function. From multi-variable calculus we known the third term of a Taylor expansion is

$$\frac{1}{2}\mathrm{d}\,x^T \mathrm{H}f(x)\mathrm{d}\,x,$$

where $\mathrm{H}f(x)$ stands for the Hessian matrix of $f(x)$.

The second order differential of a vector-valued function is not so easy because the first order differential is a vector and taking differential again should preserve the dimension but there are a lot more terms we need park somewhere. The key turns out to be taking the transpose of the first order differential and then take differential again. This is same as treating each element of the first order differential as a scalar and take differential element-wise; in other words, the second order differential of a vector-valued function is equal to a vector of element-wise second order differential of the function:

$$\begin{aligned}
\mathrm{d}^2 f(x) &= \begin{bmatrix} \mathrm{d}^2 f_1(x) \\ \vdots \\ \mathrm{d}^2 f_n(x) \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{2}\mathrm{d}\, x^T \mathrm{H} f_1 \mathrm{d}\, x \\ \vdots \frac{1}{2}\mathrm{d}\, x^T \mathrm{H} f_n \mathrm{d}\, x \end{bmatrix} \\
&= \frac{1}{2} \begin{bmatrix} \mathrm{d}\, x^T & 0 & \cdots \\ 0 & \mathrm{d}\, x^T & \cdots \\ \vdots & \ddots & \vdots \end{bmatrix} \begin{bmatrix} \mathrm{H} f_1 \\ \vdots \\ \mathrm{H} f_n \end{bmatrix} \mathrm{d}\, x \\
&= \frac{1}{2}(I \otimes \mathrm{d}\, x^T)(\mathrm{H} f(x))\mathrm{d}\, x,
\end{aligned} \tag{A.20}$$

assuming $f(x)$ is a n-dimensional vector-valued function. On the other hand, if we start with the first differential

$$\mathrm{d}\, f(x) = \frac{\partial\, f}{\partial\, x^T}\mathrm{d}\, x,$$

then we have the problem that the Jacobian is a matrix and not a vector. The solution is use vec operator. Recalling that the Hessian matrix is the Jacobian of the gradient, we take transpose first:

$$\begin{aligned}
\mathrm{d}\ \mathrm{vec}\,\mathrm{d}\,f(x)^T &= \mathrm{d}\ \mathrm{vec}\,\mathrm{d}\,x^T \frac{\partial\, f^T}{\partial\, x} \\
&= \mathrm{d}\,[(I \otimes \mathrm{d}\,x^T)\,\mathrm{vec}\,\frac{\partial\, f^T}{\partial\, x}] \\
&= (I \otimes \mathrm{d}\,x^T)\mathrm{d}\ \mathrm{vec}\,\frac{\partial\, f^T}{\partial\, x} \\
&= (I \otimes \mathrm{d}\,x^T) \begin{bmatrix} \mathrm{d}\,\frac{\partial\, f_1^T}{\partial\, x} \\ \vdots \\ \mathrm{d}\,\frac{\partial\, f_1^T}{\partial\, x} \end{bmatrix} \\
&= (I \otimes \mathrm{d}\,x^T) \begin{bmatrix} \mathrm{H}f_1 \\ \vdots \\ \mathrm{H}f_n \end{bmatrix} \mathrm{d}\,x,
\end{aligned} \tag{A.21}$$

which is same as equation (A.20) except constant 1/2.

One last thing before we conclude this appendix: the Hessian matrix is symmetric if all the second derivatives are continuous, but no guarantee the matrix in equation (A.20) is symmetric. So according to [186], we need $B + (B)_v$ to get the real Hessian if we find B in the middle of equation (A.20), where

$$(B)_v = \begin{bmatrix} B_1^T \\ B_2^T \\ \vdots \\ B_n^T \end{bmatrix},$$

and B_i for $1 \leq i \leq n$ is the $n \times n$ block of B in order from the top.

APPENDIX B

EXPECTATION-MAXIMIZATION ALGORITHM

The expectation-maximization algorithm, abbreviated EM algorithm, is a popular tool in statistics to deal with hidden variables. In the context of parameter estimation for state-space models, it is usually represented as two steps, the E-step estimates hidden states using existing estimated parameters, and the M-step estimates the parameters using the estimated states in the E-step. While this is true for the regular EM, there are variations on the basic EM that require a deeper understanding of EM algorithm as a maximum likelihood method. So in this appendix, we will give an outline of EM algorithm as maximizing a lower-bound function, and then, instead of giving a standard version of EM in parameter estimation for linear state-space models, we will give one with a time-delay in it. This problem turns out to be almost the same as that for a linear model without time-delay.

This section closely follows Minka's tutorial [192].

Before we start, we need to state Jensen's inequality, which we will make use of immediately. If $f(\cdot)$ is a convex function, then

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y), \quad 0 \leq \alpha \leq 1.$$

For concave functions such as logarithm, $\leq$ is changed to $\geq$. Extending sum to integrals we have

$$\log \int f(x)\, \mathrm{P}(x) \mathrm{d}\, x \geq \int \mathrm{P}(x) \log[f(x)] \mathrm{d}\, x, \qquad \text{where} \int \mathrm{P}(x) \mathrm{d}\, x = 1.$$

Suppose we wish to estimate the parameter vector θ in

$$\mathrm{P}_\theta(\mathbf{y}) = \int_x \mathrm{P}_\theta(\mathbf{y}, \mathbf{x}) \mathrm{d}\, \mathbf{x},$$

with $\mathbf{y}$ as observations and $\mathbf{x}$ as hidden variables. In order to use Jensen's inequality we add a function $q(\mathbf{x})$,

$$\int_x \mathrm{P}_\theta(\mathbf{y},\mathbf{x})\mathrm{d}\,\mathbf{x} = \int_x \frac{\mathrm{P}_\theta(\mathbf{y},\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x})\mathrm{d}\,\mathbf{x},$$

such that $\int_\mathbf{x} q(\mathbf{x})\mathrm{d}\,\mathbf{x}$ and $q(\mathbf{x}) \neq 0$. By Jensen's inequality, if we take logarithm we have

$$\log \int_x \frac{\mathrm{P}_\theta(\mathbf{y},\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x})\mathrm{d}\,\mathbf{x} \geq \int_x q(\mathbf{x}) \log \left[\frac{\mathrm{P}_\theta(\mathbf{y},\mathbf{x})}{q(\mathbf{x})}\right] \mathrm{d}\,\mathbf{x}. \tag{B.1}$$

Equation (B.1) is a lower bound of the log-likelihood $\log \int_x \mathrm{P}_\theta(\mathbf{y},\mathbf{x})\mathrm{d}\,\mathbf{x}$. But we want to get close to the true likelihood, so we maximize the lower-bound.

First, we augment equation (B.1) with a Lagrange-multiplier term in order to enforce $\int_\mathbf{x} q(\mathbf{x})\mathrm{d}\,\mathbf{x} = 1$,

$$\lambda(1 - \int_\mathbf{x} q(\mathbf{x})\mathrm{d}\,\mathbf{x}) + \int_\mathbf{x} q(\mathbf{x}) \log[\mathrm{P}_\theta(\mathbf{y},\mathbf{x})]\mathrm{d}\,\mathbf{x} - \int_\mathbf{x} q(\mathbf{x}) \log[q(\mathbf{x})]\mathrm{d}\,\mathbf{x}. \tag{B.2}$$

Take derivative of equation (B.2) with respect to $q(\mathbf{x})$ to get

$$-\lambda q(\mathbf{x}) + q(\mathbf{x}) \log[\mathrm{P}_\theta(\mathbf{y},\mathbf{x})] - [q(\mathbf{x}) \log(q(\mathbf{x})) - q(\mathbf{x})] = 0, \tag{B.3}$$

but since $q(\mathbf{x}) \neq 0$, we have the following two equations

$$\int_\mathbf{x} q(\mathbf{x})\mathrm{d}\,\mathbf{x} = 1 \tag{B.4}$$

$$-\lambda - 1 + \log(\mathrm{P}_\theta(\mathbf{y},\mathbf{x})) - \log(q(\mathbf{x})) = 0. \tag{B.5}$$

From equation (B.5), we have

$$\lambda + 1 = \log \frac{\mathrm{P}_\theta(\mathbf{y}, \mathbf{x})}{q(\mathbf{x})}, \tag{B.6}$$

$$\text{taking exponential to get } e^{\lambda+1} = \frac{\mathrm{P}_\theta(\mathbf{y}, \mathbf{x})}{q(\mathbf{x})}, \tag{B.7}$$

$$\text{which is same as } q(\mathbf{x}) = \frac{\mathrm{P}_\theta(\mathbf{y}, \mathbf{x})}{e^{\lambda+1}} \tag{B.8}$$

$$\text{taking integral, and we have } 1 = \int_{\mathbf{x}} q(\mathbf{x}) \mathrm{d}\,\mathbf{x} = \int_{\mathbf{x}} \frac{\mathrm{P}_\theta(\mathbf{y}, \mathbf{x})}{e^{-\lambda-1}} \mathrm{d}\,\mathbf{x} = \frac{\int_{\mathbf{x}} \mathrm{P}_\theta(\mathbf{y}, \mathbf{x}) \mathrm{d}\,\mathbf{x}}{e^{-\lambda-1}}, \tag{B.9}$$

$$\text{taking logarithms to get } \lambda + 1 = \log[\int_{\mathbf{x}} \mathrm{P}_\theta(\mathbf{y}, \mathbf{x}) \mathrm{d}\,\mathbf{x}] \tag{B.10}$$

plugging equation (B.10) into (B.5) to get

$$-\log[\int_{\mathbf{x}} \mathrm{P}_\theta(\mathbf{y}, \mathbf{x}) \mathrm{d}\,\mathbf{x}] + \log(\mathrm{P}_\theta(\mathbf{y}, \mathbf{x})) - \log(q(\mathbf{x})) = 0, \tag{B.11}$$

$$\text{and finally we have } q(\mathbf{x}) = \frac{\mathrm{P}_\theta(\mathbf{y}, \mathbf{x})}{\int_{\mathbf{x}} \mathrm{P}_\theta(\mathbf{y}, \mathbf{x}) \mathrm{d}\,\mathbf{x}} = \mathrm{P}_\theta(\mathbf{x}|\mathbf{y}) \tag{B.12}$$

This means that function $q(\mathbf{x})$ is indeed a probability density function, and because

$$\int_{x} q(\mathbf{x}) \log \left[\frac{\mathrm{P}_\theta(\mathbf{y}, \mathbf{x})}{q(\mathbf{x})} \right] \mathrm{d}\,\mathbf{x} = \int_{\mathbf{x}} \mathrm{P}_\theta(\mathbf{x}|\mathbf{y}) \log \mathrm{P}_\theta(\mathbf{y}) \mathrm{d}\,\mathbf{x} = \log \mathrm{P}_\theta(\mathbf{y}), \tag{B.13}$$

we say the maximum lower-bound touches the true log-likelihood at the current value of θ. The lower-bound can also be written as

$$\int_{\mathbf{x}} q(\mathbf{x}) \log(\mathrm{P}_\theta(\mathbf{y}, \mathbf{x})) \mathrm{d}\,\mathbf{x} - \int_{\mathbf{x}} q(\mathbf{x}) \log q(\mathbf{x}) \mathrm{d}\,\mathbf{x}, \tag{B.14}$$

of which only the first term has to do with observations $\mathbf{y}$, so we retain that to get the conditional expectation whence the E-step gets its name,

$$\int_{\mathbf{x}} q(\mathbf{x}) \log(\mathrm{P}_\theta(\mathbf{y}, \mathbf{x})) \mathrm{d}\,\mathbf{x} = \mathrm{E}_{\mathbf{x}|\mathbf{y}} \log \mathrm{P}_\theta(\mathbf{x}, \mathbf{y}). \tag{B.15}$$

The M-step maximize this bound by changing θ. From this, one can formulate an alternative definition of EM algorithm [193]:

E-step maximize the lower-bound with respect to $\mathbf{x}$ while holding θ constant;

M-step maximize the lower-bound with respect to θ while holding $\mathbf{x}$ constant.

when maximization proves difficult, we need only to seek an improved estimated of $\mathbf{x}$ and θ. This justifies the use of gradient method in EM [193].

Time delays introduce a great deal of complexity into a model. Here we are going to derive EM algorithm for linear state-space models that have time-delays. As it turns out for one kind of models with delays, the kind whose state transition and output functions depend on same delayed states, the modification is minimal in the M-step, with most of the added complexity residing in the E-step. Suppose the system is defined as follows:

$$\mathbf{x}_{k+1} = A_0\mathbf{x}_k + A_1\mathbf{x}_{k-\tau} + Bu_k + \mathbf{w}_k \tag{B.16}$$

$$y_k = C_0\mathbf{x}_k + C_1\mathbf{x}_{k-\tau} + Du_k + \mathbf{v}_k, \tag{B.17}$$

$$\mathbf{x}_1 \sim \mathcal{N}[\mu_1, P_1], \ldots, \mathbf{x}_{1-\tau} \sim \mathcal{N}[\mu_{1-\tau}, P_{1-\tau}] \tag{B.18}$$

where the state at time k is represented by $\mathbf{x}_k$ and $\mathbf{x}_1$ to $\mathbf{x}_{1-\tau}$ are the initial values whose distributions are known Gaussian distributions; the output at the same time are represented by $\mathbf{y}_k$; $\mathbf{w}_k$ and $\mathbf{v}_k$ are state and output noises assumed to be white Gaussian, and all of the initial states' mean and variances, B, D, A_0, A_1, C_0, and C_1 are parameters of the model along with the noise variance of w_k and v_k, which are Q and R respectively. The inputs u_k is assumed to be known constants and not random variables, so we will not put it in the condition of the conditional probability below.

Let us denote all states from $k = 1-\tau$ to $k = N$ as X_N, and likewise for U_N, but for Y_N, it is $1 \ldots N$. For the sake of convenience we shall make following notational changes:

$$\xi_{\mathbf{k}} = \begin{bmatrix} \mathbf{x_{k+1}} \\ \mathbf{y_k} \end{bmatrix}, \mathbf{z_k} = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-\tau} \\ u_k \end{bmatrix}, \Gamma = \begin{bmatrix} A_0 & A_1 & B \\ C_0 & C_1 & D \end{bmatrix}, \Pi = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \tag{B.19}$$

Starting with the joint probability,

$$\begin{aligned} \mathrm{P}(\mathbf{X_{N+1}}, \mathbf{Y_N}) &= \mathrm{P}(\mathbf{x_{N+1}}, \mathbf{y_N} | \mathbf{X_N}, \mathbf{Y_{N-1}})\, \mathrm{P}(\mathbf{X_N}, \mathbf{Y_{N-1}}) \\ &= \mathrm{P}(\mathbf{x_{N+1}}, \mathbf{y_N} | \mathbf{x_N}, \mathbf{x_{N-\tau}})\, \mathrm{P}(\mathbf{X_N}, \mathbf{Y_{N-1}}) \\ &\vdots \\ &= \mathrm{P}(\mathbf{x_1})\, \mathrm{P}(\mathbf{x_0}) \ldots \mathrm{P}(\mathbf{x_{1-\tau}}) \prod_{k=1}^{T} \mathrm{P}(\mathbf{x_{k+1}}, \mathbf{y_k} | \mathbf{x_k}, \mathbf{x_{k-\tau}}). \end{aligned} \tag{B.20}$$

Taking logarithm to get

$$\begin{aligned} \log \mathrm{P}(\mathbf{X_{N+1}}, \mathbf{Y_N}) \propto (x_1 - \mu_1)^T P_1^{-1} (x_1 - \mu_1) + \log\det P_1 \\ + \ldots + (x_{1-\tau} - \mu_{1-\tau})^T P_{1-\tau}^{-1} (x_{1-\tau} - \mu_{1-\tau}) + \log\det P_{1-\tau} \\ + \sum_k (\xi_{\mathbf{k}} - \Gamma \mathbf{z_k})^T \Pi^{-1} (\xi_{\mathbf{k}} - \Gamma \mathbf{z_k}) + N \log\det \Pi. \end{aligned} \tag{B.21}$$

Defining more convenient notations,

$$\Sigma = \frac{1}{N} \sum_{k=1}^{N} \mathrm{E}[\mathbf{z_k} \mathbf{z_k}^T | \mathbf{Y_N}], \quad \Phi = \frac{1}{N} \sum_{k=1}^{N} \mathrm{E}[\xi_{\mathbf{k}} \xi_{\mathbf{k}}^T | \mathbf{Y_N}],$$

$$\Psi = \frac{1}{N} \sum_{k=1}^{N} \mathrm{E}[\xi_{\mathbf{k}} \mathbf{z_k} | \mathbf{Y_N}],$$

and the conditional expectation of equation (B.20) is equal to

$$\log\det P_1 + \mathrm{tr}[P_1^{-1}\,\mathrm{E}((\mathbf{x_1}-\mu_1)(\mathbf{x_1}-\mu_1)^T|\mathbf{Y_N})] + \ldots + \log\det P_{1-\tau}$$
$$+\,\mathrm{tr}[P_{1-\tau}^{-1}\,\mathrm{E}((\mathbf{x_{1}}_{-\tau}-\mu_{1-\tau})(\mathbf{x_{1}}_{-\tau}-\mu_{1-\tau})^T|\mathbf{Y_N})] + \mathrm{tr}[\Pi^{-1}(\Pi - \Psi\Gamma^T - \Gamma\Psi^T + \Gamma\Sigma\Gamma^T)]. \tag{B.22}$$

Taking derivative with respect to μ_1 and P_1 to get the estimator of them,

$$\mu_1 = \hat{x}_{1|N}, \quad P_1 = P_{1|N}.$$

The rest of the initial values are obtained in likewise manner, that is,

$$\mu_{1-i} = \hat{x}_{1-i}, \quad P_{1-i} = P_{1-i|N}, \quad 0 \le i \le \tau.$$

Similarly, for Γ and Π, we have

$$\Gamma = \Psi\Sigma^{-1}, \quad \Pi = \Phi - \Psi\Sigma^{-1}\Psi^T.$$

The formulas for Γ and Π can be proved by the following steps:

$$\mathrm{tr}[\Pi^{-1}(\Phi - \Psi\Gamma^T - \Gamma\Psi^T + \Gamma\Sigma\Gamma^T)] = \mathrm{tr}[\Pi^{-1}((\Gamma - \Psi\Sigma^{-1})\Sigma(\Gamma-\Psi\Sigma^{-1})^T + \Phi - \Psi\Sigma^{-1}\Psi],$$

from which the minimizer $\Gamma = \Psi\Sigma^{-1}$ follows, then taking derivative of the remaining terms involving Π,

$$\frac{\mathrm{d}}{\mathrm{d}\,\Pi}\log\det\Pi + \frac{\mathrm{d}}{\mathrm{d}\,\Pi}\mathrm{tr}[\Pi^{-1}(\Phi - \Psi\Sigma^{-1}\Psi)] = \Pi^{-1} - \Pi^{-1}(\Phi - \Psi\Sigma^{-1}\Psi)\Pi^{-1} = 0,$$

and $\Pi = \Phi - \Psi\Sigma^{-1}\Psi^T$ follows.

This set of updates look exactly the same to the M-step of regular EM algorithm for linear dynamical systems. The main difficulty is in state estimation, or the E-step. I presented one way to tackle this by augmenting states in chapter VI, an approach that can be used as the E-step here.

APPENDIX C

KALMAN FILTER AND SMOOTHER

The celebrated Kalman filter can take on many forms and disguises depending on the circumstances. I understand it as a recursive least square. Unfortunately, the complexity of formulas and involved derivation steps mask its simple nature, so I will first give a version of Kalman filter in the deterministic least square context and show the similarities between it and the more familiar version when used to estimate states in linear state-space systems. This will also motivate the orthogonality principle, which is rather abstract when we apply it to more general vector spaces. The orthogonality principle also is part of geometric approach to least square and Kalman filter, which is more intuitive and more visual and easier to understand and remember.

In this appendix, I will give detailed derivation of Kalman filter and Kalman smoother using the innovation approach. Let observations y be a vector and state be another vector x, which we wish to estimate. Although y should be linearly related to x by a design matrix A, which is known. But for whatever the reason, we have

$$y \approx Ax.$$

In linear algebra terms, this says y is not in the range of A, or equivalently that $y = Ax$ has no solution. So we want to find $\hat{y}$ so that the square of error $e = y - \hat{y}$ is minimized. To put that in familiar terms, we have the following optimizing problem:

$$\min_{\text{for all feasible } x} (y - Ax)^T (y - Ax),$$

where $\hat{x}$ is an estimate of x and $\hat{x}$ is over some linear vector space, most likely an

Euclidean space. This is the classical least-square problem. The solution can be obtained through taking derivatives against $\hat{x}$ and setting it to zero.

But a more direct method that will prove powerful is the projection theorem, also called orthogonality principle. It states that the best estimate of a vector in a subspace must have its error, the difference between the estimate and observation, be orthogonal to the subspace. Here the subspace is the range of A and the vector is y. Applying the orthogonality principle we get

$$A^T(y - A\hat{x}) = 0.$$

Assuming A^TA is invertible and we have the familiar solution to deterministic least-square problem:

$$\hat{x} = (A^TA)^{-1}A^Ty.$$

An easy extension of the above solution is a weighted least-square, where the objective function becomes

$$(y - A\hat{x})^TS(y - A\hat{x}),$$

where S is a symmetric positive definite matrix. Rather than going through tedious derivation using derivatives, we can just take advantage of the generality of the orthogonality principle and define a new norm so that

$$(y - A\hat{x})^TS(y - A\hat{x}) = [S^{T/2}(y - A\hat{x})]^T[S^{T/2}(y - A\hat{x})], \tag{C.1}$$

where $S = S^{1/2}S^{T/2}$. The right hand side of equation (C.1) is in the form of regular Euclidean norm. The solution to weighted least-square is then

$$\hat{x} = (A^TSA)^{-1}A^TSy. \tag{C.2}$$

If S happens to be block diagonal then we can partition A and y so that

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_k \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}, S = \begin{bmatrix} S_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & S_k \end{bmatrix}. \tag{C.3}$$

Rewriting equation (C.2) as

$$\hat{x}_k = \left(\sum_{i=1}^{k} A_k^T S_i A_i \right)^{-1} \left(\sum_{i=1}^{k} A_i^T S_i y_i \right), \tag{C.4}$$

where $\hat{x}_k$ is the estimate of x given k observations. If there are $k+1$ observations, then

$$\begin{aligned} \hat{x}_{k+1} &= \left(\sum_{i=1}^{k+1} A_k^T S_i A_i \right)^{-1} \left(\sum_{i=1}^{k+1} A_i^T S_i y_i \right) \\ &= Q_{k+1}^{-1} \left(\sum_{i=1}^{k} A_i^T S_i y_i + A_{k+1}^T S_{k+1} y_{k+1} \right) \\ &= \hat{x}_k - Q_{k+1}^{-1} (A_{k+1}^T S_{k+1} A_{k+1}) \hat{x}_k + Q_{k+1}^{-1} A_{k+1}^T S_{k+1} \\ &= \hat{x}_k + \underbrace{Q_{k+1}^{-1} A_{k+1}^T S_{k+1}}_{\text{Kalman gain}} \underbrace{(y_{k+1} - A_{k+1} \hat{x}_k)}_{\text{innovation}}. \end{aligned} \tag{C.5}$$

This is the form we will see for the Kalman filter: recursive update of estimate by taking weighted sum of previous estimate and innovation. To continue to use the geometrical formulation of projection and orthogonality with random variables, the concepts of vector spaces must be expanded beyond the regular Euclidean spaces, to cover vector-valued random variables. Here, we will only cover the basic definitions and not be concerned with mathematical details and mathematical rigor. In particular, we will not cover anything related to completeness of vector spaces or convergence of vectors. These materials are covered by any functional analysis textbook.

But rather, we will concentrate on one unique feature of this vector space: the inner product is a square matrix.

Before proceeding, I will note here that all numbers are real in this dissertation, not complex numbers like the treatment in [90]. The main complexity of complex numbers for this work is in differentiation and complex-valued random variables, neither of which is simple, nor is the complex number strictly necessary for this work, so we take the practical approach and the simpler treatment is chosen. Even Kailath et al. [90], a book renown for its complete coverage, only gives an outline of differentiation of complex numbers and a short introduction to complex-valued random variables, leaving more detailed treatments to other books.

In this section we follow Kailath et al. [90], especially the appendix of Chapter 4. Their choice of an unusual vector spaces whose inner products result in matrices is very convenient, but, as they noted, uncommon in literature.

Suppose there is a ring of scalars, $\mathcal{S}$, over which a vector space, $\mathcal{V}$ is defined. The vector must be closed under vector addition and closed under scalar multiplication. Furthermore, the addition and multiplication must obey the following rules:

$$\forall x, y \in \mathcal{V}, \forall \alpha, \beta \in \mathcal{S},$$

$$\begin{aligned} x + y &= y + x & (\alpha + \beta)x &= \alpha x + \beta y \\ (x + y) + z &= x + (y + z) & (\alpha\beta)x &= \alpha(\beta x) \\ \alpha(x + y) &= \alpha x + \alpha y & \underline{0} \cdot x &= 0; \underline{1} \cdot x = x, \end{aligned}$$

where $\underline{0}$ and $\underline{1}$ designate the zero vector and identity vector, respectively.

The ring of scalars we will use is the $n \times n$ matrices of real numbers, $n > 1$. The reason for this is so that we can define an inner product on vector space $\mathcal{V}$ that yields elements of $\mathcal{S}$. The elements of vector space $\mathcal{V}$ are n-dimensional vector-valued random variables.

The inner product then is the expectation of two random variable vectors:

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{V}, < \mathbf{x}, \mathbf{y} >= \mathrm{E}\, \mathbf{x}\mathbf{y}^T \tag{C.6}$$

Every inner product must obey three rules:

Linearity:

$$< \alpha_1 x_1 + \alpha_2 x_2, y >= \alpha_1 < x_1, y > +\alpha_2 < x_2, y >$$

Reflexivity:

$$< y, x >=< x, y >^T$$

Nondegeneracy:

$$\|x\|^2 \overset{\triangle}{=} < x, x >= 0 \text{ if and only if } x = 0,$$

which also serves the definition of inner products.

Of course, we could have defined the vector space as scalar random variables and real numbers as the ring, which would make the resulting vector space more familiar, but more difficult to work with in deriving Gramians.

The Gramian is usually defined for a collection of vectors $v_1, v_2, \ldots, v_n$ as

$$[G]_{ij} =< v_i, v_j > .$$

This is tedious, when we have to concatenate a series of inner products to obtain a matrix. But instead of this piece-meal fashion, we could see that if $V = [v_1, v_2, \ldots, v_n]^T$, then

$$G = \mathrm{E}\, V V^T =< V, V >,$$

a single inner product resulting in the Gramian.

We summarize the situation here because it can be confusing:

- The elements of the vector space are multiple-dimensional random variables in column form.
- The scalar result of an inner product of two vectors is a square matrix.
- The vector-valued random variables can also be multiplied by regular matrices and vectors and real numbers to result in a new random variable.
- In particular, a random variable that is the product of a matrix and another random variable can be part of an inner product that produces a new matrix.

Some simple properties concerning random variables (also vectors in the vector space defined in this appendix) are very useful and used thorough out this dissertation:

- If random variables $\mathbf{x}$ and $\mathbf{y}$ are both zero mean, i.e., $\mathrm{E}\,\mathbf{x} = 0$ and $\mathrm{E}\,\mathbf{y} = 0$, then their variances and covariances are inner products: $\mathrm{var}\,\mathbf{x} =< \mathbf{x}, \mathbf{x} >$, $\mathrm{cov}(\mathbf{x}, \mathbf{y}) =< \mathbf{x}, \mathbf{y} >$.
- Two independent random variables $\mathbf{x}$ and $\mathbf{y}$ whose means are both zero are orthogonal: $< \mathbf{x}, \mathbf{y} >= \mathrm{E}\,\mathbf{x}\mathbf{y}^T = (\mathrm{E}\,\mathbf{x})(\mathrm{E}\,\mathbf{y}^T) = 0$. Strictly speaking, we only need one random variable's mean to be zero for this to be true.
- From the orthogonality principle, the optimal vector in subspace $\mathcal{Y}$ that approximates vector $x \notin \mathcal{Y}$ is $\hat{x} = P_{\mathcal{Y}}(x)$, the vector x projected onto $\mathcal{Y}$, where $P_{\mathcal{Y}}(\cdot)$ is the projection operator for subspace $\mathcal{Y}$.

It is already noted that Kalman filter in the deterministic case can be seen as a recursive least square; now we will show that it can be seen as a recursive Stochastic least square in the linear state estimation problem. The first step is to find the answer to Stochastic least square, then the recursion; they are covered in the next subsection. Both of these are relative simple, given the vector space defined above and the orthogonality principle, so the main materials of this section is the second subsection where taking advantage of the state-space form to result in the actual formulas is involved.

For two related (nothing to estimate if independent), zero-mean, random variables, $\mathbf{x}$ and $\mathbf{y}$, of which $\mathbf{y}$ is observation, $\mathbf{x}$ the hidden variable to be estimated, the best linear estimator is $\hat{\mathbf{x}} = K\mathbf{y}$, for some matrix K. The orthogonality principle says K must obey

$$< \mathbf{x} - \hat{\mathbf{x}}, \mathbf{y} >= 0,$$

$$< \mathbf{x} - K\mathbf{y}, \mathbf{y} >= 0,$$

$$< \mathbf{x}, \mathbf{y} >=< K\mathbf{y}, \mathbf{y} >,$$

$$K =< \mathbf{x}, \mathbf{y} >< \mathbf{y}, \mathbf{y} >^{-1},$$

assuming $R_{\mathbf{y}} =< \mathbf{y}, \mathbf{y} >$ is invertible. The optimal estimator then is

$$\hat{\mathbf{x}} =< \mathbf{x}, \mathbf{y} > R_{\mathbf{y}}^{-1} y.$$

If there are many independent observations $\mathbf{y}_1, \ldots, \mathbf{y}_N$, then

$$\hat{\mathbf{x}} = \sum_{j=1}^{N} < \mathbf{x}, \mathbf{y}_j > R_{\mathbf{y}_j}^{-1} y_j.$$

In the real world observations are rarely independent, and for linear state-space models, they are pointedly not. There is a way to make a set of vectors orthogonal in linear algebra, which is known as the Gram-Schmidt process. So now, thanks to the vector space defined in this appendix, we can apply the same process to random variables.

Suppose we have orthogonalized $\mathbf{y}_1, \ldots, \mathbf{y}_{i-1}$ into $\mathbf{e}_1, \ldots, \mathbf{e}_{i-1}$ orthogonal vectors called innovations, with property

$$< \mathbf{e}_j, \mathbf{e}_k >= 0, \qquad j \neq k$$

then the next innovation

$$\mathbf{e}_i = \mathbf{y}_i - \sum_{j=1}^{i-1} < \mathbf{y}_i, \mathbf{e}_j > \|\mathbf{e}_j\|^{-2} \mathbf{e}_j. \tag{C.7}$$

The initial value of this recursion is $\mathbf{e}_1 = \mathbf{y}_1$. The second part of equation (C.7) is $\mathbf{y}_i$ projected onto the subspace $\mathcal{L}\{\mathbf{e}_1, \ldots, \mathbf{e}_{k-1}\} = \mathcal{L}\{\mathbf{y}_1, \ldots, \mathbf{y}_{i-1}\}$. Kalman filter of course does not want $\mathbf{y}$ projected but $\mathbf{x}$ projected onto the subspace span by all the observations, but the form is still the same. And that is the power of innovation approach, its general applicability. Kalman et al. [194] used this approach in their seminal paper, but somehow failed to notice the same approach can be used to derive the smoother, with the result that their paper left smoothing as a future work.

The state-space model we will assume is

$$\mathbf{x}_{i+1} = F_i \mathbf{x}_i + G_i \mathbf{w}_i \tag{C.8}$$

$$\mathbf{y}_i = H_i \mathbf{x} + \mathbf{v}_i, \tag{C.9}$$

for $i = 1, \ldots, N$, and where $\mathbf{x}_i$ is the state and $\mathbf{y}_i$ is the observation, where F_i, G_i, H_i are known parameters, and noises $\mathbf{w}_i$ and $\mathbf{v}_i$ are uncorrelated white Gaussian random variables with zero means and variances $Q_i > 0$ and $R_i > 0$, respectively. The matrix G_i's main purpose is to deal with the case that not all elements of state $\mathbf{x}_i$ are Stochastic, as when part of $\mathbf{x}_{i+1}$ linearly depends on part of $\mathbf{x}_i$, which we will see an instance of when we deal with models with time delays. There is no input in equations (C.8) and (C.9) but it is very easy to obtain estimates of states with inputs from those without inputs, and inputs do not impact on variance estimates.

Not obvious but very important is the property that

$$< \mathbf{x}_i, \mathbf{w}_j >= 0, \quad < \mathbf{x}_i, \mathbf{v}_j >= 0, \quad j \le i,$$

the state is always independent of current and future noises, and

$$< \mathbf{y}_i, \mathbf{w}_j >= 0, \quad < \mathbf{y}_i, \mathbf{v}_j >= 0 \quad j > i,$$

the observation is always independent of future noises. We could also denote the independence by its vector space equivalent, orthogonality, so $< \mathbf{x}_i, \mathbf{w}_j >= 0$ is same as $\mathbf{x}_i \perp \mathbf{w}_j$.

From equation (C.7), we have the innovation as

$$\mathbf{e}_i = \mathbf{y}_i - \hat{\mathbf{y}}_{i|i-1},$$

recall that the second term of equation (C.7) is $\mathbf{y}_i$ projected onto $\mathcal{L}\{\mathbf{e}_1, \ldots, \mathbf{e}_{k-1}\}$, which is same as the estimator of $\mathbf{y}_i$ given all observations before time i. By same token, the estimator of state $\mathbf{x}_i$ given all observations before time i is

$$\hat{\mathbf{x}}_{i+1|i} = \sum_{j=1}^{i} < \mathbf{x}_{i+1}, \mathbf{e}_j > \|\mathbf{e}_j\|^{-2} \mathbf{e}_j.$$

Define $R_{e,j} = \|\mathbf{e}_j\|^2$, the major assumption so far is that $R_{e,j} > 0$, which can be derived from $R_j > 0$. As it turns out, computing $< \mathbf{x}_{i+1}, \mathbf{e}_j >$ when $j < i$ is involved, but if we split the term when $j = i$ out from the rest, we get

$$\begin{aligned} \hat{\mathbf{x}}_{i+1|i} &= \sum_{j=1}^{i-1} < \mathbf{x}_{i+1}, \mathbf{e}_j > R_{e,j}^{-1} \mathbf{e}_j + < \mathbf{x}_{i+1}, \mathbf{e}_i > R_{e,i}^{-1} \mathbf{e}_i \\ &= \hat{\mathbf{x}}_{i+1|i-1} + < \mathbf{x}_{i+1}, \mathbf{e}_i > R_{e,i}^{-1} \mathbf{e}_i. \end{aligned} \tag{C.10}$$

Now we must turn to state transition equation.

$$\begin{aligned}
< \mathbf{x}_{i+1}, \mathbf{e}_i > &= F_i < \mathbf{x}_i, \mathbf{e}_i > + G_i < \mathbf{w}_i, \mathbf{e}_i > \\
&= F_i(< \mathbf{x}_i, \tilde{\mathbf{x}}_{i|i-1} > H_i^T + < \mathbf{x}_i, \mathbf{v}_i >) + < \mathbf{w}_i, \tilde{\mathbf{x}}_i > H_i^T + < \mathbf{w}_i, \mathbf{v}_i > \\
&= F_i(< \tilde{\mathbf{x}}_{i|i-1}, \tilde{\mathbf{x}}_{i|i-1} > H_i^T + 0) + 0 + 0 \\
&= F_i P_{i|i-1} H_i,
\end{aligned} \tag{C.11}$$

where P_i is the variance of state estimator $\hat{\mathbf{x}}_i$ and whose recursive estimation will be discussed shortly. In addition,

$$\begin{aligned}
R_{e,j} &=< \mathbf{e}_j, \mathbf{e}_j > \\
&=< H_j \tilde{\mathbf{x}}_{j|j-1} + \mathbf{v}_j, H_j \tilde{\mathbf{x}}_{j|j-1} + \mathbf{v}_j > \\
&= H_j P_{j|j-1} H_j^T.
\end{aligned} \tag{C.12}$$

Therefore, the state estimator is

$$\hat{\mathbf{x}}_{i+1|i} = F_i \hat{\mathbf{x}}_{i|i-1} + F_i P_i H_i (H_i P_{i|i-1} H_i^T)^{-1} \mathbf{e}_i.$$

If measurements go missing for one or more time steps, this scheme does not work, so S.F. Schmidt at NASA came up with a decomposition of the state estimator into a time update and a measurement update, so that if a measurement is missing, the measurement update can be skipped for that particular measurement but the filter can continue. The time update is simple:

$$\hat{\mathbf{x}}_{i+1|i} = F_i \hat{\mathbf{x}}_{i|i}.$$

This simply projects equation (C.8) onto all the observations up to time k. The measurement update for state estimator is

$$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + P_i H_i (H_i P_{i|i-1} H_i^T)^{-1} \mathbf{e}_i.$$

This separate updates of state estimator is the most popular form of Kalman filter we see today.

Kalman and Bucy [194] defined the variance of state estimator as

$$P_{i|i-1} =< \tilde{\mathbf{x}}_{i|i-1}, \tilde{\mathbf{x}}_{i|i-1} > .$$

Because of the popularity of separate updates for Kalman filter, here we will use that form of variance update as well. The time update is again simple:

$$P_{i+1|i} = F_i P_{i|i} F_i^T + G_i Q_i G_i^T.$$

The equation above comes from taking variance of equation (C.8). The measurement update is more involved.

$$\begin{aligned} P_{i|i} &=< \tilde{\mathbf{x}}_{i|i}, \tilde{\mathbf{x}}_{i|i} > \\ &=< \tilde{\mathbf{x}}_{i|i}, \mathbf{x}_i > -K_{f,i} < \mathbf{e}_i, \mathbf{x}_i > \\ &= P_{i|i-1} - K_{f,i}(H_i < \tilde{\mathbf{x}}_{i|i}, \mathbf{x}_i > +0) \\ &= P_{i|i-1} - K_{f,i} H_i P_{i|i-1} \\ &= P_{i|i-1} - P_{i|i-1} H_i^T R_{e,i}^{-1} H_i P_{i|i-1}. \end{aligned} \tag{C.13}$$

Thus completes the celebrated Kalman filter. Although there are different formulations of the smoothing problem, here I refer to the so-called fixed-interval smoothing problem, where given a sequence of observations $y_1, \ldots, y_N$, we want to find out the best estimator $\mathbf{x}_{i|N}$ for $i \in 1, \ldots, N$. As it turns out, the innovation approach gives us a solution in a straightforward manner, using only the results from Kalman filter.

First let us define a new notation,

$$P_{i,j} =< \hat{\mathbf{x}}_{i|i-1}, \hat{\mathbf{x}}_{j|j-1} > .$$

Obviously, $P_{i,i} = P_{i|i-1}$. Again, by the orthogonality principle we have

$$\hat{\mathbf{x}}_{i|N} = \sum_{j=1}^{N} < \mathbf{x}_i, \mathbf{e}_j > R_{e,j}^{-1} \mathbf{e}_j, \tag{C.14}$$

which can be pulled into two parts as

$$\hat{\mathbf{x}}_{i|i-1} + \sum_{j=1}^{N} < \mathbf{x}_i, \mathbf{e}_j > R_{e,j}^{-1} \mathbf{e}_j.$$

Of course, we could have used filtered estimator $\hat{\mathbf{x}}_{i|i}$ instead of the predicted estimator $\hat{\mathbf{x}}_{i|i-1}$ with corresponding change in the second term, and the result would be valid as well. $R_{e,j}$ remains unchanged from equation (C.12). The main change comes in $< \mathbf{x}_i, \mathbf{e}_j >$ where $j \geq i$. Now let us look a closer look:

$$\begin{aligned} < \mathbf{x}_i, \mathbf{e}_j > &=< \mathbf{x}_i, H_j \tilde{\mathbf{x}}_{j|j-1} > + < \mathbf{x}_i, \mathbf{v}_j > \\ &=< \tilde{\mathbf{x}}_{i|i-1}, \tilde{\mathbf{x}}_{j|j-1} > H_J^T + < \hat{\mathbf{x}}_{i|i-1}, \tilde{\mathbf{x}}_{j|j-1} > + < \mathbf{x}_i, \mathbf{v}_j > \\ &= P_{i,j} H_j^T + 0 + 0, \end{aligned} \tag{C.15}$$

where the first zero is become $\hat{\mathbf{x}}_{i|i-1}$ is in the vector space span by the first $i-1$ observations which is a subset of the vector space span by the first $j-1$ observations, but $\tilde{\mathbf{x}}_{j|j-1}$ is orthogonal to the vector space span by the first $j-1$ observations, and the second zero is by assumption that state noises are uncorrelated with observation noises. So now we need to compute $P_{i,j}$.

The key is a recursive relation of the error,

$$\tilde{\mathbf{x}}_{i+1|i} = F_{p,i} \tilde{\mathbf{x}}_{i|i-1} + G_i \mathbf{w}_i - K_{p,i} \mathbf{v}_i, \tag{C.16}$$

where

$$F_{p,i} = F_i - K_{p,i} H_i \quad \text{and} \quad K_{p,i} = F_i P_i H_i (H_i P_{i|i-1} H_i^T)^{-1}.$$

By recursively expanding $\tilde{\mathbf{x}}_{j|j-1}$ and therefore decreasing j to i using the equation above, we arrive at a simple formula for $P_{i,j}$ as

$$P_{i,j} =< \hat{\mathbf{x}}_{i|i-1}, \hat{\mathbf{x}}_{j|j-1} >= P_{i|i-1} F_{p,i}^T F_{p,i+1}^T \cdots F_{p,j-1}^T, \quad j > i. \tag{C.17}$$

Defining a convenience variable $\lambda_{i|N}$ as

$$\lambda_{i|N} = P_{i|i-1} \sum_{j=i+1}^{N} F_{p,i}^T F_{p,i+1}^T \cdots F_{p,j-1}^T H_j^T R_{e,j}^{-1} \mathbf{e}_j,$$

and the smoother is

$$\hat{\mathbf{x}}_{i|N} = \hat{\mathbf{x}}_{i|i-1} + P_{i|i-1} \lambda_{i|N}, \tag{C.18}$$

and we also have the corresponding variance matrix as

$$P_{i|N} = P_{i|i-1} + P_{i|i-1} \Lambda_{i|N} P_{i|i-1}, \tag{C.19}$$

where

$$\Lambda_{i|N} =< \lambda_{i|N}, \lambda_{i|N} >= F_{p,i}^T \Lambda_{i+1|N} F_{p,i} + H_i^T R_{e,}^{-1} H_i, \qquad \lambda_{N+1|N} = 0. \tag{C.20}$$

Here $\Lambda_{i|N}$ is in recursive form and we can do the same for $\lambda_{i|N}$:

$$\lambda_{i|N} = F_{p,i}^T \lambda_{i+1|N} + H_i^T R_{e,}^{-1} \mathbf{e}_i, \qquad \lambda_{N+1|N} = 0. \tag{C.21}$$

Although the smoother presented here is in recursive form, it is not recursive in $\hat{\mathbf{x}}_{i|N}$ and $P_{i|N}$ as in [91]. To do so, one simply uses the filter version of equation (C.18) and (C.19) (they are the predicted version) to solve for $\lambda_{i|N}$ and $\Lambda_{i|N}$ and plug them back into equation (C.18) and (C.19). I will do some of that in the next section on lag-one covariance.

Lag-one covariance is not part of regular Kalman smoother but is required for parameter estimation. We include it here for coherence.

Define lag-one covariance as

$$M_{i,i-1|N} =< \tilde{\mathbf{x}}_{i|N}, \tilde{\mathbf{x}}_{i-1|N} > .$$

For the same of brevity we will refer to $M_{i,i-1|N}$ as $M_{t|N}$.

The derivation of it is similar to Kalman smoother and in fact depends on some of the results in the derivation of Kalman smoother. The first clue we use is that instead of starting with lag-one smoother definition we start with $P_{i,i-1}$,

$$\begin{aligned} < \tilde{\mathbf{x}}_{i|i-1}, \tilde{\mathbf{x}}_{i-1|i-2} > &=< \tilde{\mathbf{x}}_{i|N} + P_{i|i-1}\lambda_{i|N}, \tilde{\mathbf{x}}_{i-1|N} + P_{i-1|i-2}\lambda_{i-1|N} > \\ &=< \tilde{\mathbf{x}}_{i|N}, \tilde{\mathbf{x}}_{i-1|N} > + < \tilde{\mathbf{x}}_{i|N}, P_{i-1|i-2}\lambda_{i-1|N} > \\ &\quad + < P_{i|i-1}\lambda_{i|N}, \tilde{\mathbf{x}}_{i-1|N} > + < P_{i|i-1}\lambda_{i|N}, P_{i-1|i-2}\lambda_{i-1|N} > \\ &= M_{i|N} + 0 + 0 + P_{i|i-1} < \lambda_{i|N}, \lambda_{i-1|N} > P_{i-1|i-2} \\ &= M_{i|N} + P_{i|i-1} < \lambda_{i|N}, F_{p,i-1}^T\lambda_{i|N} + H_{i-1}^T R_{e,i}^{-1} e_{i-1} > P_{i-1|i-2} \\ &= M_{i|N} + P_{i|i-1}(\Lambda_{i|N}F_{p,i-1} + < \lambda_{i|N}, H_{i-1}^T R_{e,i}^{-1} e_{i-1} >)P_{i-1|i-2} \\ &= M_{i|N} + P_{i|i-1}(\Lambda_{i|N}F_{p,i-1} + 0)P_{i-1|i-2}, \end{aligned} \tag{C.22}$$

therefore,

$$M_{i|N} = P_{i-1,i}^T - P_{i|i-1}\Lambda_{i|N}F_{p,i-1}P_{i-1|i-2}. \tag{C.23}$$

Equation (C.23) is enough for computation with all necessary values available from Kalman filter and smoother, and since lag-one covariance does not have to be symmetric positive definite, there is no square-root implementation either.

However, equation (C.23) is not in a recursive form and is not the way it usually appears, for instance in [88]. To obtain the usual recursive form, we need a convenient notation. Define

$$J_i = P_{i|i} F_i^T P_{i+1|i}^{-1}.$$

Also important is a equality

$$P_{i|i-1} F_{p,i}^T = P_{i|i} F_i^T$$

when observation and process noise are uncorrelated, which gives

$$J_i = P_{i|i-1} F_{p,i}^T P_{i+1|i}^{-1}.$$

From equation (C.23) we have

$$M_{i+1|N} = P_{i,i+1}^T - P_{i+1|i} \Lambda_{i+1|N} F_{p,i} P_{i|i-1}. \tag{C.24}$$

Taking advantage of recursive relation of $\Lambda_{i+1|N}$ and $\Lambda_{i|N}$, we obtain

$$\begin{aligned}
M_{i|N} &= P_{i-1,i}^T - P_{i|i-1}(F_{p,i}^T \Lambda_{i+1|N} F_{p,i} + H_i^T R_{e,i}^{-1} H_i) F_{p,i-1} P_{i-1|i-2} \\
&= P_{i-1,i}^T - P_{i|i-1} F_{p,i}^T P_{i+1|i}^{-1} (P_{i+1|i} \Lambda_{i+1|N} F_{p,i} P_{i|i-1}) P_{i|i-1}^{-1} F_{p,i-1} P_{i-1|i-2} \\
&\quad - P_{i|i-1} H_i^T R_{e,i}^{-1} H_i F_{p,i-1} P_{i-1|i-2} \\
&= (P_{i-1|i-2} F_{p,i-1}^T)^T - P_{i|i-1} H_i^T R_{e,i}^{-1} H_i F_{p,i-1} P_{i-1|i-2} \\
&\quad - P_{i|i-1} F_{p,i}^T P_{i+1|i}^{-1} (P_{i,i+1}^T - M_{i+1|N}) P_{i|i-1}^{-1} F_{p,i-1} P_{i-1|i-2} \\
&= P_{i|i-1} J_{i-1}^T - P_{i|i-1} H_i^T R_{e,i}^{-1} H_i P_{i|i-1} J_{i-1}^T + J_i (M_{i+1|N} - F_i P_{i|i}) J_{i-1}^T \\
&= P_{i|i} J_{i-1}^T + J_i (M_{i+1|N} - F_i P_{i|i}) J_{i-1}^T;
\end{aligned}$$

the last equality uses the measurement update formula of Kalman filter. The initial condition of this recursive formula is less elaborate:

$$\begin{aligned}< \tilde{\mathbf{x}}_{N|N}, \tilde{\mathbf{x}}_{N-1|N} > &=< \tilde{\mathbf{x}}_{N|N}, \mathbf{x}_{N-1} + \hat{\mathbf{x}}_{N-1|N} > \\ &=< \tilde{\mathbf{x}}_{N|N}, \mathbf{x}_{N-1} > +0 \\ &=< \tilde{\mathbf{x}}_{N|N}, \tilde{\mathbf{x}}_{N-1|N-2} + \hat{\mathbf{x}}_{N-1|N-2} > \\ &=< \tilde{\mathbf{x}}_{N|N}, \tilde{\mathbf{x}}_{N-1|N-2} > +0 \\ &=< \mathbf{x}_N - \hat{\mathbf{x}}_{N|N}, \tilde{\mathbf{x}}_{N-1|N-2} >,\end{aligned}$$

using the measurement update formula on $\hat{\mathbf{x}}_{N|N}$ to get

$$\begin{aligned}&< \mathbf{x}_N - k_{f,N}\mathbf{e}_N + \hat{\mathbf{x}}_{N|N-1}, \tilde{\mathbf{x}}_{N-1|N-2} > \\ &=< \tilde{\mathbf{x}}_{N|N-1}, \tilde{\mathbf{x}}_{N-1|N-2} > - < k_{f,N}\mathbf{e}_N, \tilde{\mathbf{x}}_{N-1|N-2} > \\ &= F_{N-1}P_{N-1|N-1} - k_{f,N} < H_N\tilde{\mathbf{x}}_{N|N-1} + \mathbf{v}_N, \tilde{\mathbf{x}}_{N-1|N-2} > \\ &= F_{N-1}P_{N-1|N-1} - k_{f,N}H_N < \tilde{\mathbf{x}}_{N|N-1}, \tilde{\mathbf{x}}_{N-1|N-2} > \\ &= F_{N-1}P_{N-1|N-1} - k_{f,N}H_NF_{N-1}P_{N-1|N-1} \\ &= (I - k_{f,N}H_N)F_{N-1}P_{N-1|N-1}.\end{aligned} \tag{C.25}$$

Kalman filter and related topics fill volumes and in this short appendix I only presented how some of the formulas used in this dissertation came about. Again, Kailath et al. [102] is an excellent resource for its thoroughness and readability. In there, one can find topics I have not included here, in particular, the square-root implementation of Kalman filter and smoother. Because of numerical inaccuracy, various variance matrices could become non-positive definite in the process of computation, which is a major source of numerical instability in Kalman filter. One solution is to propagate the square-root of the variance matrices, thus ensuring the positive definiteness, but at the cost of more computational effort. Another way is to turn all the update formulas into scalar updates. This is also discussed in Durbin and Koopman's book [181].

VITA

Hao Xiong received his B.S. degree from the University of Houston in August 2001 and his Ph.D degree from Texas A&M University in December 2008.

His research interests include dynamic modeling of genetic regulatory networks, signal transduction networks, and metabolic networks. He is interested in the functional role of these biological networks and how their dynamics and organizing principles cause them to robustly perform their functions. He can be reached at

Department of Computer Science

Texas A&M University

TAMU 3112

College Station, TX 77843-3112

haoxiong@tamu.edu.