Title
Towards a Systems Understanding in Biology: Data and Modeling

Author
Chang, Young Hwan

Publication Date
2013

Peer reviewed|Thesis/dissertation

Towards a Systems Understanding in Biology: Data and Modeling

by

Young Hwan Chang

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Claire J. Tomlin, Co-chair
Professor Masayoshi Tomizuka, Co-chair
Professor Karl Hedrick
Professor Shmuel Oren

Spring 2013

Towards a Systems Understanding in Biology: Data and Modeling

Abstract

Towards a Systems Understanding in Biology: Data and Modeling

by

Young Hwan Chang

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Claire J. Tomlin, Co-chair

Professor Masayoshi Tomizuka, Co-chair

Breast cancer signaling pathways and neural systems are composed of networks of mutually dependent and thus interconnected genes or neurons. Systems biology is an emerging interdisciplinary field of study that focuses on understanding these complex interactions using a systematic approach. The main goal is transforming biology into a fully quantitative, theory-rich science to understand complex behavior and produce effective predictions. However, since often only incomplete abstracted hypotheses exist to explain observed complex behavior and functions, new techniques for identifying breast cancer signaling pathways and neural systems need to be able to incorporate different types of experimental data and varying levels of prior knowledge.

This dissertation describes several efforts aimed towards an understanding of these systems by developing system identification tools and computational analysis techniques to facilitate these studies. In systems biology, models may focus on different features, different aspects and different objectives, so there can exist several model classes.

The first class of tools that we present includes the hybrid Boolean framework and optimization-based inference, and is developed in conjunction with existing mathematical tools. For example, the former is a combination of ordinary differential equations (ODEs) and Boolean networks (BNs), and the latter is that of graph models and linear time-varying systems. These tools take advantage of existing mathematical models and compensate for their limitations or disadvantages. The second class of tools that we present includes data-driven graph reconstruction using compressive sensing, discrete mode identification via sparse subspace clustering, and low-rank representation of neural activity. These adopt a data-driven approach using little prior knowledge which has its origin in the computer vision literature. Case studies of Human Epidermal growth factor Receptor2 overexpressed breast cancer and neural systems for a Brain Machine Interface are given.

Since the challenge in systems biology has become to show that the identified networks and corresponding mathematical models are enough to represent the underlying system, these tools are developed both for identifying models and also suggesting experimental directions

to better understand the systems. The ultimate goal of the work presented is to create a framework to broaden the spectrum of different modeling approaches since each model can be described through a different perspective and may highlight different aspects of the system.

To God, my family and my wife

Contents

List of Figures

Acknowledgments

The past five years of graduate studies at Berkeley has been a long journey for me but I have been blessed. I studied at great academic atmosphere and met my wife here in Berkeley. It has been an honor to join the research group of my advisor, Professor Claire Tomlin. Her research style and interest were good fit for me and she provided me with the entryway into systems biology research. Also, working under her patient guidance, support and encouragement has led to fruitful results. I have learned much from her and very much appreciated her understanding and support through my graduate studies.

I would also like to express my profound appreciation to Professor Masayoshi Tomizuka, Professor Karl Hedrick and Professor Shmuel Oren, who have been gracious in agreeing to be part of my dissertation committee, in taking their time to offer valuable comments and suggestions in spite of their busy schedules.

The work to be presented in my thesis is the result of collaboration with as well as support provided by many colleagues, collaborators and mentors. I appreciate and give my sincere thanks to our collaborators at Oregon Health and Science University (used to be at Lawrence Berkeley National Laboratory), including Nicholas J. Wang, James E. Korkola, Paul Spellman and Joe Gray for their valuable comments and sharing their experimental dataset, and also to members of Professor Carmena's laboratory, including Amy Orsborn, Suraj Gowda, Simon A. Overduin and Professor Jose Carmena for their important role in explaining to me the basics of neuroscience and brain machine interface, and feedback over the course of the project. I have been generously supported by a number of funding sources, Integrated Cancer Biology Program by the National Cancer Institute, Emerging Frontiers in Research and Innovations by National Science Foundation and STX foundation from Korea.

I would like to show my gratitude to the other individuals which have been important in my long journey at Berkeley. I am very much grateful to members of the hybrid systems laboratory, both past and present, for providing me with a comfortable working atmosphere. I would like to thank them for all the interesting discussion and research presentation during our group meeting, which have added to my knowledge with both depth and broadness and the numerous valuable comments, suggestions as well as new insights. I also would like to thank to many other students that provided the unique experiences in attending Berkeley.

In closing, I would like to thank my family, grandmother, parents, parents-in-law, sister and brother-in-law. My family provided emotional encouragement and always believed in me. Without them, I would have not been able to finish. Most of all, I would like to thank my wife, Jeesun for her dedicated support and always provided encouragement, happiness and sincere prayer.

Young Hwan Chang
Berkeley, California
May, 2013

Chapter 1

Introduction

"So far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality." (Albert Einstein)

1.1 Systems Biology

A system is composed of networks of mutually dependent and thus interconnected components that integrate the system into a unified whole. Every system exhibits emergent behavior, a property possessed not by the individual components on their own but by the whole system. It is apparent from simple investigations on the brain and cancer cell development that the structure of an entire system actually orchestrates and constrains the behavior of the components [151].

Systems biology is an emerging field and currently undergoing enormous expansion, from roots in molecular and genomic biology. It has generated revolutions in ecology, population biology, and evolutionary studies and is slowly making advances into biochemistry, developmental biology, genetics, and whole-plant biology. This has led biologists to recognize that biological systems can be studied not only in terms of their mechanistic, molecular-level components but also simultaneously in terms of complex interactions with biological functions [84]. This systems approach is integral to current research in molecular cell biology and stands in contrast to the reductionist paradigm of molecular biology, since reductionist mechanistic investigations would miss the vital element of orchestration [131][100].

The field of systems biology is an interdisciplinary field of study that focuses on complex interactions within biological systems. Even though the application of mathematical modeling to molecular cell biology is not a new endeavor and there is a long history of mathematical descriptions of biochemical and genetic networks, these mathematical works have not been considered central to most of molecular cell biology. However, this attitude is changing. Currently, system-level investigations are frequently accompanied by mathematical models, and improved by reincarnations of existing and new technologies such as treatment of can-

cer, mouse models, siRNA-based gene knocking down screenings, computational modeling of disease, and personalized cancer medicine [159].

The main focus of systems biology is to understand the regulation of complex cellular pathways and of intracellular communication in quantitative and predictable ways, so as to shed light on complex biological functions such as metabolism, cell signaling, cell cycle, apoptosis, and cell differentiation. For example, the genome has presented biologists with an opportunity to study genetic processes on a genomic scale, and to achieve quantitative understanding, not just of individual molecular mechanisms but also of their interactions and regulation at the systems level. Since the component identification market of modern biology is approaching saturation, understanding function is becoming increasingly important for molecular cell biology, function which is not at the level of single genes but rather at a higher level of abstraction involving pathways. Particularly from the end of the 20th century, the system-level investigation concept has been used widely in the biosciences in a variety of contexts, to understand biological systems and discover underlying properties and organisms functioning as a system whose theoretical description is only possible using techniques which fall under the remit of systems biology [2]. Many research papers provide definitions of systems biology [86][84][88][83][97][6]:

- *"Every object that biology studies is a system of systems." by Francois Jacob [88]*
- *"Systems Biology does not investigate individual genes or proteins one at a time, as has been the highly successful mode of biology for the past 30 years. Rather, it investigates the behavior and relationships of all the elements in a particular biological system while it is functioning." by Ideker et al.[83].*
- *"To understand biology at the system level, we must examine the structure and dynamics of cellular and organismic function, rather than the characteristics of isolated parts of a cell or organism." by Kitano [97].*
- *"Systems Biology seeks to predict the quantitative behavior of an in vivo biological process under realistic perturbation, where the quantitative treatment derives its power from explicit inclusion of the process components, their interactions, and realistic values for their concentrations, locations, and local states." by the National Institute of General Medical Science at NIH [6].*

Therefore, systems biology is concerned with the study of biological functions and mechanisms, underpinning inter-cellular and intracellular dynamical networks, by means of system orientated approaches. The transformation of biology into a fully quantitative, theory-rich science now seems inevitable, even though not yet quite within reach [161]. Also, these developments have produced great intellectual challenges to the many communities involved, such as mathematicians, physicist, chemists, biologists and even control engineers.

Systems biology can also be defined operationally, as by the MIT Computational and Systems Biology Initiative[1], in terms of four stages - measurement (from wet laboratory bio-

[1]http://csbi.mit.edu/

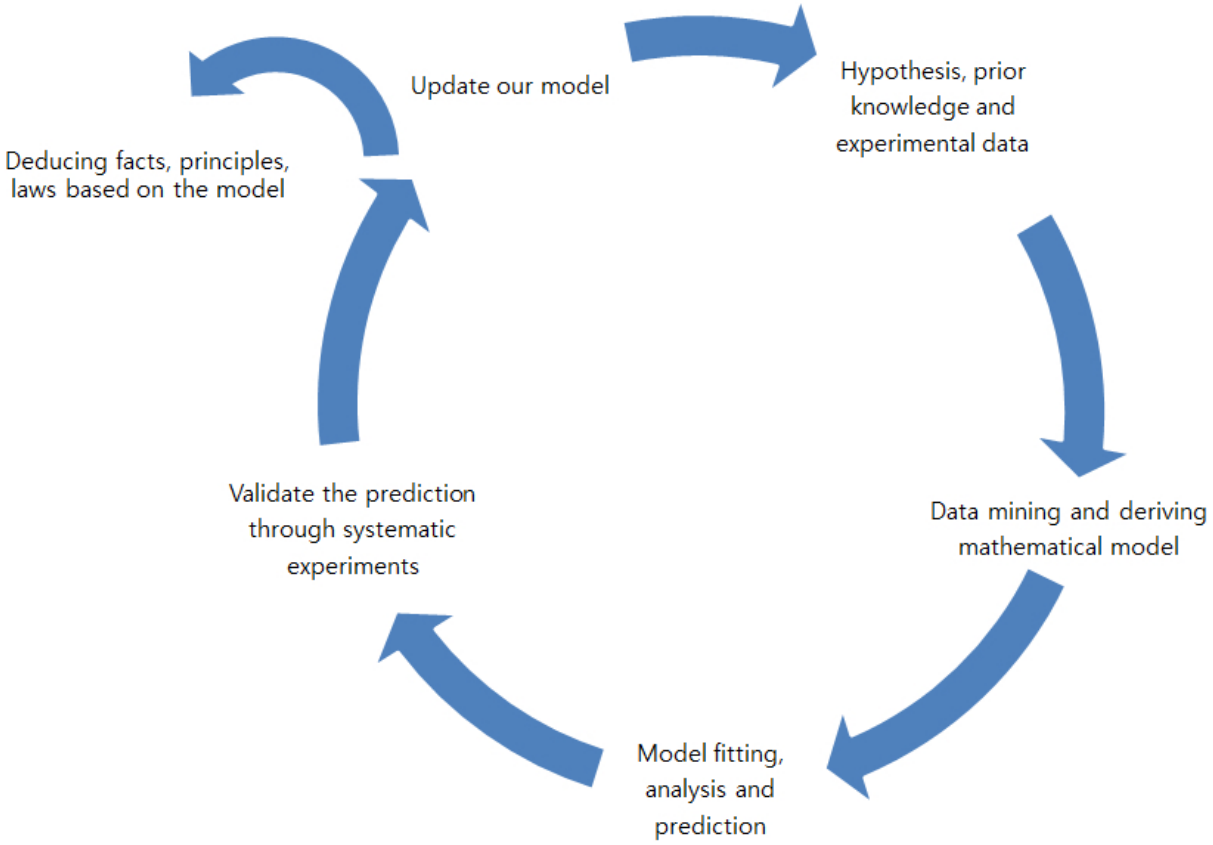


Figure 1.1: Operational definition of systems biology.

logical experiments), mining (bioinformatics, database and data mining), modeling (network models, mechanistic models, and biochemical models), and manipulation (systematic experiments such as molecular genetics, chemical genetics, drug treatment, cell engineering, etc.) [84]. Therefore, a systems biology approach means investigating the components of cellular networks and their interactions; applying experimental high-throughput and whole-genome techniques; and integrating computational and theoretical methods with experimental efforts. Thus, the overall goal of systems biology is to foster links among biology, mathematical model and engineering, and to create an interdisciplinary field in order to undertake the systematic analysis of complex biological phenomena. More detail descriptions of the overall operational stages are shown in Figure 1.1.

1.2 Biological Complexity

Complexity is a serious problem in biology, and it is likely that chemical and physical processes in living cells are extremely complex. The great strength of genomic biology is its

ability to unravel biological problems one gene (or protein) at a time. Even though this research paradigm has dominated life sciences for over 50 years, component-by-component analysis will not suffice in the study of signal transduction, oncogene transformation, neurobiology, and other processes in which many genes interact. The early stages of systems biology have been dominated by an emphasis on studying ever more genes in simple settings and using simple types of data. However, as the complexity of the system increases, biological systems can be characterized by a multi-dimensional landscape which defines distinct types of complexity. For instance, as the complexity of the system increases, the number of molecular species under investigation increases; the type of data changes from sequence to structure, to sub-cellular localization, and then to time-dependent changes in protein activities in cells; the complexity of the biology increases from cells to tissues, to organisms and finally to populations.

Recently, measurement can be undertaken in a high-throughput, multivariate manner using various kinds of array technologies. Because of these high dimensional datasets from which it is relatively difficult to draw hypothesis by intuition, many mathematical tools or algorithms are necessary for mining data and reducing dimensionality which generate hypotheses or possible interpretation of these datasets. There are many successes and significant contributions to basic understanding through the use of quantitative reasoning and theory, for example, the analysis of microarray data using hierarchical clustering and correlation coefficients between genes has stood the test of time, compared to more complex approaches [55]. This approach provides a good introduction to systems-level thinking about cellular dynamics. For similar reasons, computational modeling is required in order to develop new predictions, understand dynamic behaviors and design new experiments effectively. As biological systems represent complex, interconnected and even nonlinear molecular systems, unaided human intuition cannot produce effective predictions.

It should be noted that the complexity of biological systems does not reside solely in the number of components and interactions, nor in their associated structural and chemical properties, but also in the hierarchical connection across space and time scales from gene-level to cell-level to organism-level to population-level. Therefore, mathematical modeling should be composed of a multidisciplinary approach comprising dynamical and topological properties and aims at a systems understanding of underlying molecular and cellular mechanism of biological systems.

1.3 Mathematical Models of Biological Systems

One of the most exciting trends and important themes in systems biology involves the use of high-throughput genomic, proteomic, and metabolomic measurement data to construct models of complex systems and diseases. Used appropriately and effectively, mathematical models can represent pathways in a physically and biologically realistic manner, incorporate a wide variety of empirical observations, and generate novel and useful hypotheses. For example, Hodgkin presents a beautiful example of biophysical modeling that is the foundation

of neurobiology [82]. The story of Hodgkin and Huxley's study of giant nerve fibers runs step-by-step from the basic biology and physiology of axons and action potentials, through the crucial electrophysiology experiments, to the now famous *Hodgkin-Huxley equations*. These computational models or reconstructions of the biological system are necessarily dependent on both prior biological understanding of the system and the existence of well-annotated datasets describing and representing the components of these systems.

However, in almost any case, models are only rough representations of their biological counterparts because it is challenging to construct accurate models and establish rigorous links to experimental data. Nevertheless, mathematical models enable one to:

- elucidate network properties and formalize current knowledge in a non-ambiguous way including quantitative data;
- check the reliability of basic assumptions, and uncover lack of knowledge and requirements for clarification;
- tackle multiple sources of biological complexity at the same time in an effective and rigorous fashion;
- have significant predictive power and increase the effectiveness of experimental design and analysis.

Biological processes can be described in mathematical terms and each process can be described through different (mathematical) models. However, a biological object can be investigated by means of different experimental methods and the choice of a mathematical model or an algorithm depends on the problem, the purpose, and the intention of the investigator. Professor Lauffenburger provided a similar perspective at the *2011 AACR-NCI Systems Biology Joint Conference*:

"Choose the proper approach based on your aim and what you have."

Since a model has to reflect essential properties of the system, different models may highlight different aspects of the same instance. For a given object of study, many models may focus on different features of the object so there can exist many models of the given objects.

In general, modeling approaches can be either (1) based on prior biological understanding of the molecular mechanism involved or (2) constructed solely by analyzing the data itself, without having to make any assumptions about the underlying mechanism. It is also possible to have (3) a combination of these. In a physiochemical model [3], for example, elementary reactions such as physical association, biochemical transformation, and changes in cellular compartment are expressed as equations that are then linked into a larger system. Since the resulting mathematical model is based on literature whenever possible and identified with acquired experimental data, we call it a *"mechanistic model"* or *"theory-driven model"*. However, when we have little prior knowledge, the so-called *"data-driven model"* allows multivariate biological measurements to derive biological insights from the datasets.

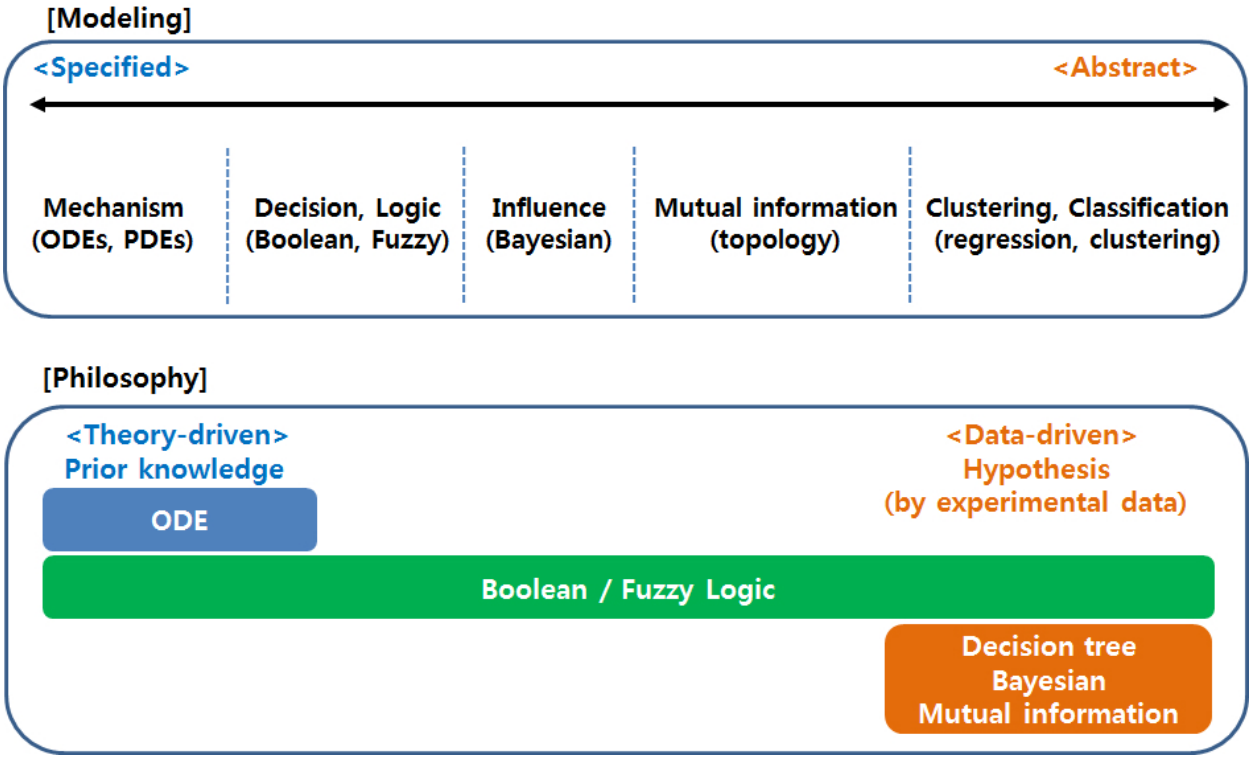


Figure 1.2: Spectrum of different modeling approaches in systems biology: *"Choose the proper approach based on your aim and what you have"* by Professor Lauffenburger, at the *2011 AACR-NCI Systems Biology Joint Conference.*

For example, data-driven models such as Bayesian inference, hierarchical clustering or classification often reveal new, surprising and unanticipated biological insight. These approaches can extract meaningful insight by reducing complexity in data itself. Figure 1.2 shows a spectrum of different modeling approaches in systems biology and underlying philosophy behind these modeling approaches.

1.3.1 Mechanistic models

Mechanistic modeling seeks to describe biomolecular reactions in terms of equations derived from established physical and chemical theory. These theory-driven models use prior knowledge to represent a specific biological system and they work well for pathways in which components and connectivities are relatively well established. For instance, equations in biochemical reaction models refer to identifiable reactions such as activation or inhibition, and parameters have physical interpretation such as reaction rate or binding affinity.

A model can be viewed as translations of pathway maps into a mathematical form which should become easier and more transparent. The correct mathematical model for a biological

system depends on the properties of the system and the objectives of the modeling effort. Ordinary and partial differential equations (ODEs and PDEs) are most commonly used and both can be cast in either deterministic or stochastic form. For instance, ODE models represent the rates of production and consumption of individual biomolecular species in terms of mass action kinetics, an empirical law stating that rates of a reaction are proportional to the concentrations of the reacting species. Changes in localization are represented by compartmentalization; two fundamental assumptions of the compartmentalized ODE are that each species is well mixed within a compartment, and transport between compartments is slow and associates with an observable rate. If these assumptions are not satisfied, then it is necessary to change the model in which the concentration of each species is described explicitly with respect to space, typically using PDEs. Also, the models are likely to be hybridized by incorporating highly detailed representations of critical reactions and more granular and flexible views of the system as a whole.

The critical decisions in the design of mechanistic models are those about specifying the scope and level of detail. Obviously, reaction models can only encompass a small subset of all reactions taking place in cells. If the scope is too small, predictive power is lost; if the scope is too large, the uncertainty is overwhelming. The issue of model scope usually depends on both biological understanding of the system and existence of datasets representing the scope of the system.

1.3.2 Data-driven models

While mechanistic modeling approaches should be based on prior biological understanding of the molecular mechanisms involved, when prior knowledge is sparse, alternatively data-driven model is more appropriate. It can help us to analyze large data sets by simplifying measurements themselves, without having to make any assumptions about the underlying mechanism [89].

New measurement technology is permitting large-scale quantitative studies of signaling networks. Since it is routine to obtain measurements from the same cell or tissue across space, time and spectral image, data-driven model approaches have recently become far more powerful due to the new measurement technologies which are high-throughput, large scale and quantitative. Such data are hard to understand completely by inspection and intuition. However, data-driven modeling approaches, for example, Bayesian inference, classification, clustering and dimension reduction techniques such as principal component analysis can derive biological insights from large-scale experiment datasets.

In addition, *"more data"* actually means a broader sampling of the behavior and dynamics of the biological mechanisms [86]. Therefore, as systems biology matures, such multidimensional data is critical to develop models and draw hypotheses that describe the spatial and temporal dynamics of biological system. A significant challenge for developing models is to continue to handle these datasets where suitable representations do not exist yet, and to match the growing diversity and quantity of the dataset.

1.4 Challenges in Systems Biology: *it's not just "a" problem.*

1.4.1 The data problem

We often read that biologists are being overwhelmed with data and that the field is data rich [46]. However, we believe that biology is actually data poor relative to the complexity of the problems being tackled. In order to develop accurate systems-wide models of biological systems, we need a *"systematic data set"* which is self-consistent experimental data covering a substantial number of biochemical processes specific to the problem under study. For example, measurements should be collected under the same condition such as a particular cell line, the same space and time scale, except for the independent variables [94][146].

We refer to this as a systematic data set [46]. We believe that the creation of these data sets will be a substantial challenge. Even for simple types of data such as protein sequence, RNA expression levels etc., we have to make sure that all test procedures and collection data are under the same controlled conditions. Especially, for large scale and time consuming experiments, we should perform all experiments and collect data in the same fashion, (except for the independent variable).

Also, classifications of large data sets are quite important and should precede building a mathematical model. These classifications could improve the organization of the data sets in a way that reveals underlying dynamics. Therefore, the key to successful clustering is defining what is meant by *"similar"* and how *"close"* is defined. For example, since in general, biological data are not static but dynamic, the metric of distance might be *"geodesic"* distance. Figure 1.3 shows our classification result of a Reverse Phase Protein Array (RPPA) data set [81] based on Isometric feature mapping (Isomap) [11]. Isomap gives successful classification among different cell lines by cell line characteristics. For example, we can see that SKBR3 and AU565 cell lines are clustered closely because they are nearly identical at the genetic and transcriptomic levels as well as at the level of HER family proteins (A formal statement and explanation of Isomap are given in the Appendix). We also anticipate that the construction of systematic data sets will involve innovation [46]: (1) the development of new experimental methods to monitor key biological reactions in cells, tissues, and organisms; (2) mathematical modeling of experimental methods to uncover sources of variation and establish the "degree of belief" associated with individual measurements such as values and their probability density functions; and (3) novel informatics methods to gather and fuse measurements into reliable self-consistent data sets suitable for probabilistic analysis.

1.4.2 Model development

In order to build a mathematical model, we should identify the specific questions that shall be answered, along with the background, problem and hypotheses. For available knowledge, we can check and collect quantitative and structural knowledge such as the components of

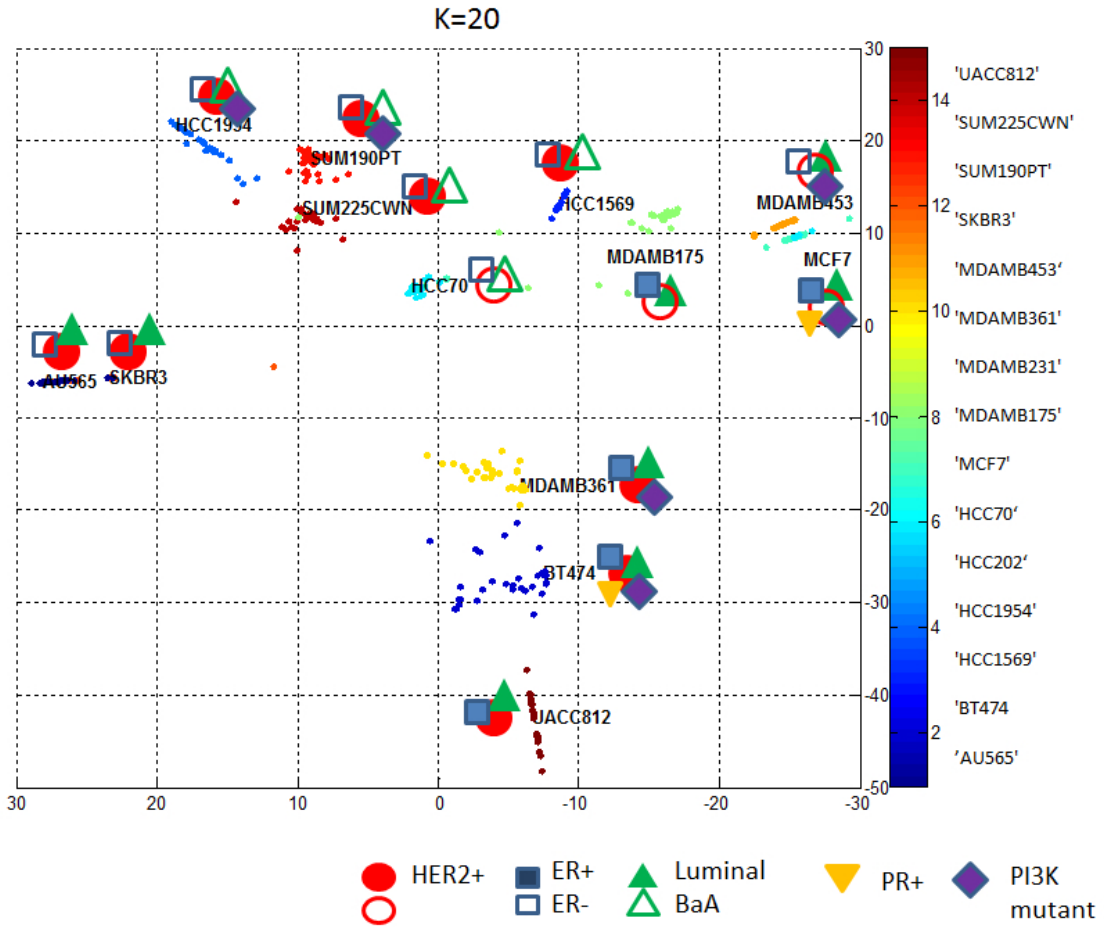


Figure 1.3: Successful classification of an RPPA data set [81] which includes 179 proteins measurement with different treatments (DMSO, LAP, AKTi, LAP+AKTi) at 8 time points (0.5hr, 1hr, 2hr, 4hr, 8hr, 24hr, 48hr, 72hr) for 15 cell lines (480×179). Shown are the results of Isomap and we put labels such as HER2+/−, ER+/−, Luminal, BaA, PR+ and PI3K mutant based on references. Note that the Isomap[11] result classifies RPPA data set in a low dimension feature space and clusters the dataset in a way consistent with cell line characteristics, for example, amplified, non-amplified, mutant type, and so on. 3-Dimensional representation of classification result is also presented in the Appendix.

the system, interaction map or network structure, kind of interactions and experimental results with respect to phenotype responses against different stimuli (e.g. gene knockout, RNAi, environmental conditions). The most important thing for selection of model classes is that we have to choose a proper model approach based on our aim and what we have. For example, we have to consider the level of description (atomistic, molecular, cellular, physiological), deterministic or stochastic model, discrete or continuous variables and static,

dynamical, spatio-temporal dynamical system and so on.

1.4.3 Challenges

Many mathematical modeling applications have traditionally sought to extrapolate from existing information and underlying principles to represent various systems. Based on these models, we could analyze, simulate, and predict behavior to further understand the system.

However, in studying biological systems, often only incomplete abstracted hypotheses exist to explain observed complex behavior and functions; the system dynamics is known to be sufficiently complex, and the dimension of state variables is large (relative to that of engineering applications) because of the complexity of the problem. Thus, the challenge has become to show that the identified networks and corresponding mathematical models are enough to represent the underlying system.

As an example, for developing a mechanistic model, since the parameter identification problem includes non-convex optimization in general, we cannot guarantee that our local optimal parameters represent the global optimal. For a similar reason, even though there are many data-driven approaches for inferring graph structure, there is no statistical guarantee on how well the inferred graph structure represents the underlying system when we consider time-varying and nonlinear systems. Also, a particularly challenging problem in data-driven reconstruction of networks is to identify whether or not important nodes in the graph structure are missing, how many are missing, and where these nodes are in the interconnection structure. Moreover, we should always consider measurement noise and unmodelled dynamics.

In spite of these challenges, as systems biology matures, in order to continue to affect the study of metabolic networks, protein interactions and genetic networks in systems biology, mathematical modeling must simultaneously characterize the complex and nonintuitive behavior of a network, while revealing deficiencies in the model and suggesting new experimental directions.

1.5 Organization

This dissertation covers several mathematical tools for modeling or identifying biological systems in systems biology. Parts of the material presented here has appeared previously in papers: Chang and Tomlin [33][37][35][36] and Chang *et al.* [34][31][32][38][115]. In the following, we provide an overview of the main themes from each of the subsequent chapters. Though the introduction contains few citations, the appropriate references to earlier work will be given in each chapter. Lastly, the contributions of this thesis are summarized below.

In Chapter 2, we cover a brief overview of both the Integrative Cancer Biology Program (ICBP) and Emerging Frontiers in Research and Innovation (EFRI) project. We consider two projects as applications for the proposed system identification tools. First, for the ICBP project, we consider the HER2 (Human Epidermal growth factor Receptor 2) overexpressed

breast cancer signal pathway. We have the basic skeletal framework of molecular events functioning parallel to and downstream of HER2/3 within the PI3K/Akt pathway which will be described in detail. Parts of the network are very well studied, yet there is a significant part of the network which is currently unknown. Our goal is to develop and experimentally validate an integrated computational model of responses of HER2-amplified cells to tyrosine kinase inhibitors (TKIs) targeting the catalytic kinase function of the HER family proteins.

Second, for the EFRI project, we consider the primate motor system in a setting that takes into consideration the strong coupling between brain, biomechanics and behavior. Our goal is to advance our understanding of fundamental principles in the neural control of movement in 3-dimensional space involving physical interactions with the world. The central question is whether motor programs help a monkey control a highly redundant multi-degree of freedom biomechanical plant or not. For this, we build a mathematical model and analysis tool in order to test the motor program. Through this study, we anticipate that our understanding of the primate motor system can help improve the current neuroprosthetic approaches.

In Chapter 3, we develop a Hybrid Boolean Model (ODE+Boolean) for HER2 overexpressed breast cancer signal pathways with postulated epigenomic feedback. The basic idea in this model is to combine continuous dynamical systems (an ODE model for already well-known parts of the network) with a discrete transition system (Boolean, for postulated but largely unknown components). This framework is easier to validate than a complete ODE model for large and complex signal pathways to find unknown pathways which match the response to experimental data. The advantage of using a Boolean model for the unknown parts of the network is that relatively few parameters are needed. Thus, the framework avoids over-fitting, covers a broad range of pathways, and easily represents various experimental conditions. The overall goal of the hybrid model is to predict the behavior of HER2 overexpressed breast cancer signaling pathway, thus helping to understand unknown parts of the pathway between experimental results and qualitative/quantitative results.

In Chapter 4, we propose a data-driven inference scheme to identify temporally evolving network representations of genetic networks. In the formulation of the optimization problem, we use an adjacency map as *a priori* information, and define a cost function which both drives the connectivity of the graph to match biological data as well as generates a sparse and robust network at corresponding time intervals. Through simulation studies on simple examples, it is shown that this optimization scheme can help to capture the topological change of a biological signaling pathway, and furthermore, might help to understand the structure and dynamics of biological genetic networks. We use this algorithm to study a breast cancer signaling pathway to understand short-term and long-term behaviors by capturing the dynamic evolution of the network and understanding HER2-amplified breast cancer signal pathway with a systems point of view. For example, for the short-term behavior after TKI is introduced, downstream components are regulated effectively because the biological network shows Positive Negative (PN) feedback. On the other hand, for long term behavior, the downstream components are activated because the biological network evolves to Positive Positive (PP) feedback which induces a slower but amplified signal response.

In Chapter 5, we consider the problem of reconstructing network structure from observed data, and in turn uncovering the underlying mechanisms responsible for the observed behaviors. First, we focus on a sparsely connected graph structure with all nodes accessible for measurement and no measurement noise, and then propose a method for reconstructing the graph structure without any *a priori* information of connectivity, based on time series gene expression data. Our method is based on compressive sensing (CS) and we show the importance of incoherence and demonstrate that incoherence in the sensing matrix can be used as a guideline for designing effective experiments.

Second, we consider a more general problem in which there might be hidden nodes which affect system dynamics, since a key challenge inherent in the network reconstruction problem for practical use comes from the necessity to deal with noisy and partial measurements. Then, we ask whether it is still possible to reconstruct the graph structure reliably when the dynamics of a certain node is corrupted by arbitrarily large errors and in addition, all the measurements are contaminated by measurement noise. We show that we can infer the graph structure reliably by solving a two-stage convex optimization problem and demonstrate our studies with a set of numerical example to illustrate its performance. Also, we show a simple biological example of HER2 overexpressed breast cancer using an RPPA dataset.

In Chapter 6, we present a new spatio-temporal perspective in analyzing neural activity dataset for the brain machine interface (BMI) system. We consider the use of a hybrid system approach in order to understand how the brain uses motor programs to control a highly redundant multi-degree of freedom arm. The monkey might decompose a simpler task and determine the sequence of the discrete modes of operation in which each discrete mode may represent a pose such as planning, execution or a certain pose or motor program. We consider a hybrid system modeling approach to understand the operation of brain and design a decoder map between neural activity and kinematic movements. This would be necessary for a big step towards the next level neuroprosthetic technology.

First, we use unsupervised segmentation of neural activity via Sparse Subspace Clustering (SSC) to identify the discrete mode. The idea is that once we interpret and extract the features of our system such as a switching condition, the number of modes and sequences of modes, then we could use conventional identification for a single Linear Time Invariant (LTI) system. We demonstrate that the monkey brain does in fact operate in various discrete modes when controlling arm movements which is a useful abstraction for engineering application.

Second, we apply Robust Principal Component Analysis (RPCA), a method well suited for extracting data matrices' low-rank component, for example dynamically meaningful structure, to neural spike datasets in order to extract neural signatures that signify the onset of submovements, a type of motor primitive. We aim to identify common event-related neural features and validate submovement-based motor primitives inferred from the hand velocity profiles. This method allows (1) removal of sparse corruption signal or task-irrelevant signal from data, (2) identification of task-related dynamic patterns, and (3) detection and prediction of submovements. We also explore using the Random Projection (RP) techniques and applying RP to data prior applying RPCA improved the submovement prediction performance by de-sparsifying neural data while preserving certain statistical characteristics of

aggregate neural activity.

In Chapter 7, we close with a summary of the main results presented in the dissertation, as well as some thoughts on direction on future work. The key novel ideas and main contributions in this thesis are:

- developing a hybrid Boolean framework which is easier to validate than a complete ODE model for large and complex signal pathways, avoids over-fitting, covers a broad range of pathways, and easily represents various experimental conditions;
- proposing an optimization-based inference scheme which is able to capture the topological change of a biological signaling pathway and might help to understand the structure and dynamics of biological genetic networks;
- developing a data-driven algorithm for identifying gene regulatory networks which is able to guarantee the exact reconstruction, suggest new experimental design and reveal deficiencies in the model without any prior information;
- providing new perspectives in analyzing neural activity datasets to understand and identify motor programs, and demonstrating that the monkey brain does in fact operate in various discrete modes and submovements are associated with neural activity when controlling arm movements.

Chapter 2

Project Overviews

We provide a general overview and background of two projects that are central to this dissertation: (1) "Modeling response to HER2 targeted therapies" in collaboration with OHSU/UC Berkeley/LBNL/MD Anderson/UCSF Center for Cancer Systems Biology[1] and (2) "A hybrid control systems approach to the Brain-Machine interfaces for exoskeleton control" (EFRI-M3C[2]) in collaboration with Brain-Machine Interface Systems Laboratory[3] and Mechanical Systems Control Laboratory[4]. Parts of the material described here have appeared in project proposals [72][29].

2.1 HER2 Overexpressed Breast Cancer

2.1.1 Background and significance

Approximately one quarter of sporadic breast cancers, representing about 45,000 new cases per year in the U.S., have amplification and overexpression of the HER2 gene [138]. An abundance of *in vitro* and *in vivo* experimental models have established that overexpression of HER2 is a potential cause confirming a link between HER2 amplification and breast cancer [104][78]. Furthermore, HER2-amplified breast cancer appears to represent a unique type of solid tumor cancer due to its exquisitely HER2-dependent biology [105]. This has been reproducibly demonstrated in engineered models of inducible HER2 overexpression and in knockdown models of human tumor cells [40][58][9][105][133].

The antibody trastuzumab was first introduced for treatment almost a decade ago and efforts to develop targeted therapies against HER2-amplified breast tumors began with the development of trastuzumab to inhibit HER2 signaling [14][139]. Trastuzumab induces reduced proliferation by downregulation of HER2/neu leading to disruption of factors and repression

[1] http://sysbio.banatao.berkeley.edu/
[2] http://www.nsf.gov/eng/efri/about.jsp
[3] http://www.eecs.berkeley.edu/∼carmena/
[4] http://www.msc.berkeley.edu

of proangiogenic factors, and inhibits the effect of overexpression of HER2. Trastuzumab showed significant clinical activity but molecular mechanisms underlying the clinical activity of trastuzumab remain unknown and may involve a significant component of immunological targeting of tumor cells [104][45][108]. More recently, tyrosine kinase inhibitors (TKIs) targeting the catalytic kinase function of the HER family proteins have been developed and represent the second generation of attempts to target HER2 signaling in HER2-amplified breast cancers. In contrast to trastuzumab, TKIs have a solid mechanistic basis and are well positioned to test the HER2-targeting treatment hypothesis. However, the single agent clinical activities of this class are modest and the most extensively studied TKI, lapatinib, which is introduced more recently, also shows efficacy but is not durable [75][22][67].

Clinical responses to trastuzumab and lapatinib are not uniform between patients and typically are not durable. Several mechanisms have been identified that may modulate response including (a) activating downstream mutations in the PI3K pathway, (b) cytokine and microenvironment mediated activation of interacting networks, (c) PI3K mediated changes in the HER3 phosphorylation-dephosphorylation equilibrium and (d) transcriptional feedback regulation from response related network elements. The central question that motivates this project is:

"Do the computational models of the diverse resistance and response mechanisms allow us to develop optimal receptor tyrosine kinase network targeted drug combinations?

2.1.2 Goals of the research

The goal of this project is to develop and experimentally validate an integrated computational model of responses of HER2-amplified cells to lapatinib (chosen because of its comparatively simple mechanism of action) that encompasses rapid phosphorylation (on the order of a few minutes) and second messenger based signaling, influences from the microenvironment and slower processes involving transcription mediated feedback regulation including epigenomic modulation (on the order of a few days). This model will enable comparative assessment of the relative importance of mechanisms of resistance, elucidation of mechanisms of transcription coupled feedback and optimization of treatment timing and multi-drug combinational strategies.

(a) Rapid phosphoprotein mediated signal transduction

HER2-amplification induced constitutive signaling is initiated by heterodimerization of HER2 with HER3 and other HER-family receptor tyrosine kinases (RTKs) to activate downstream signaling networks including PI3K and MAPK. These, in turn, initiate downstream transcriptional events including expression COX-2, CXCR4, ETS-family transcription factors and HIF-1α that drive increased proliferation, enhance survival, alter differentiation and increase migration. Initial signaling events are rapidly occurring on the time scale of minutes. The core components of this process are illustrated schematically in Figure 2.1. Various

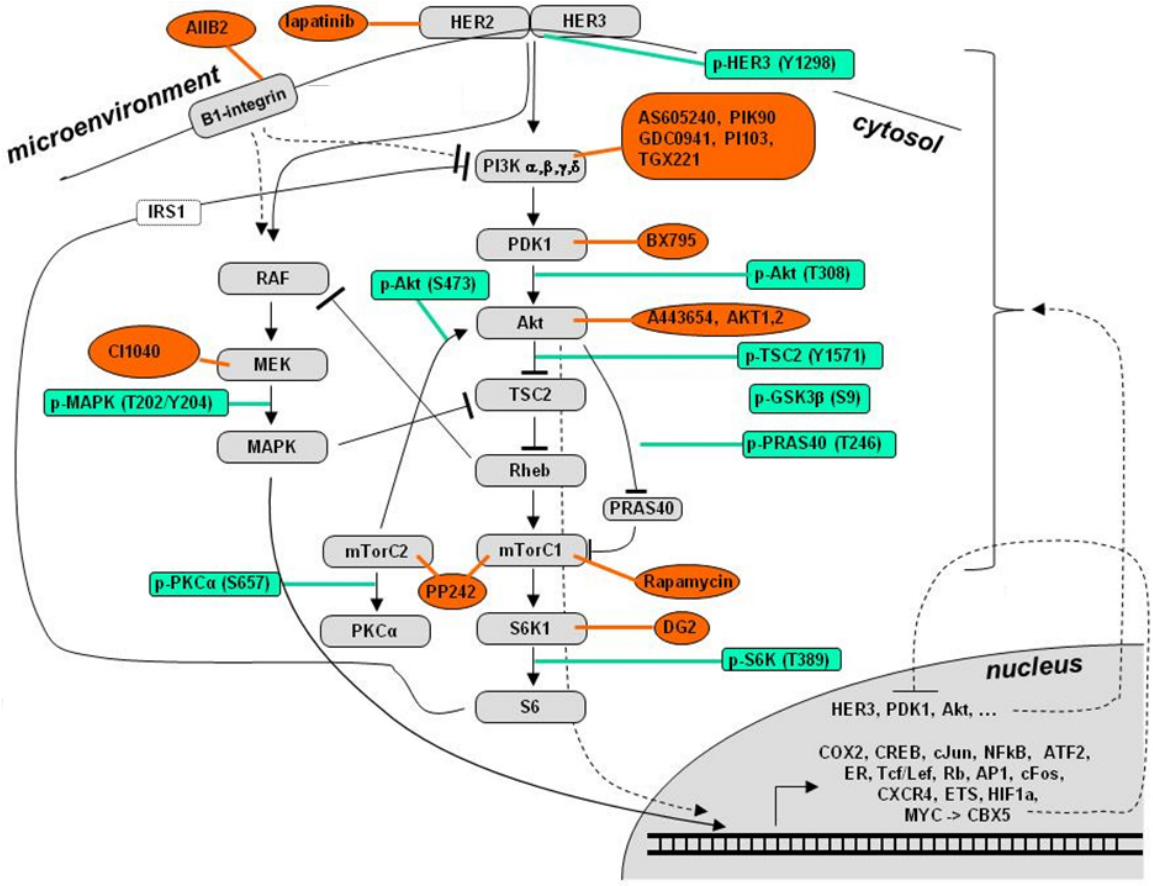


Figure 2.1: A simplified schematic showing activating (arrows) and inhibiting (bars) relationship within the basic skeletal framework of molecular events functioning parallel to and downstream of HER2/3 within the PI3K/Akt pathway. Cross-talk and relationship with the RAF/MEK/MAPK pathway are described. Signaling proteins are shown in gray. The drugs that will be used to perturb the circuit and interrogate the system are shown in orange adjacent to their targets. The protein readouts that will be used to assay signaling activity after drug treatments are shown in green at their proper locations on the circuit. Transcriptional feedback relations regulating signaling network components to be further elucidated are indicated as dashed lines (Figure 1 in [72]).

models have been developed to describe and help understand HER signaling. The most comprehensive model to date is the Immediate-early HER Reaction Model (IERMv1.0) developed in [39], which presents an ordinary differential equation (ODE) model of the mass action kinetics of Akt and ERK signaling, in response to stimulation of the four HER receptors. The model is developed based on the immediate-early response (0-120 minutes after receptor stimulation). To construct an accurate model, it is required to include the impact of mutations in genes encoding downstream signaling elements, especially PI3K mutations,

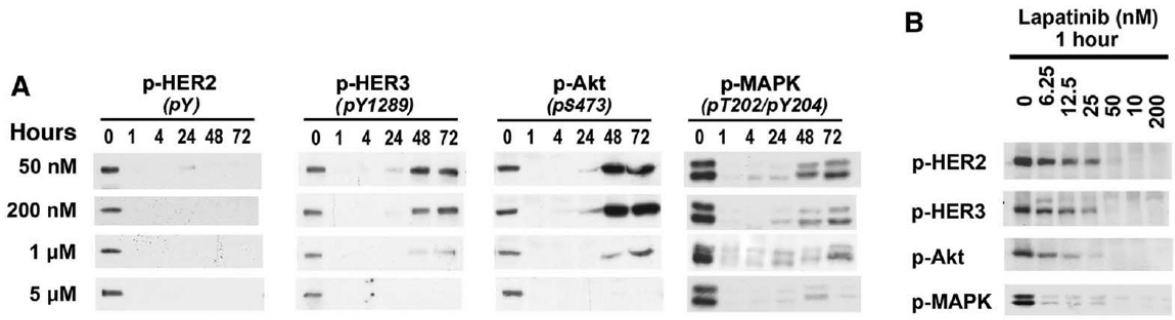


Figure 2.2: Inhibition of HER2/3 signaling by lapatinib. SKBR3 cells were treated for the times indicated and cell lysates were assessed using western blotting. These data show that HER2/3 signaling is inhibited at 1 hour by $50nM$ lapatinib but $5\mu M$ drug is required to durably inhibit HER2/3 signaling (Figure 1 A and B [5]).

since PI3K mutations dampen biological responses to lapatinib in preclinical models and in the clinic. Model features and parameters for PI3K network mutations may significantly alter both rate constants and the PI3K signaling network structure itself. Also, it is necessary to include a long term behavior which will be described below.

(b) Long term transcriptional feedback regulation

Despite continued treatment of tumor cells with TKIs, HER3 signaling and Akt activity resume after 12-24 hours as shown in Figure 2.2 [136][120]. This is not an artifact of monolayer cell culture model since it is reproducible in xenograft models *in vivo* [136]. If HER3 is not permitted to escape (by siRNA knockdown), TKI inhibition is highly pro-apoptotic. Thus, the failure to inhibit HER3/PI3K/Akt signaling significantly undermines the anti-tumor efficacies of TKIs against HER2-driven breast cancers. These findings are entirely consistent with the limited anti-tumor activities of these drugs seen in clinical trials and identify inhibition of the HER2-HER3 transactivation as a rate limiting barrier.

Our collaborative groups of Dr. Gray's Lab at OHSU and Dr. Moasser's Lab at USCF have extensively studied the mechanisms that allow HER3 to escape HER2-directed drug therapy. Gene expression profiling shows no induction of novel tyrosine kinases or increased expression of existing tyrosine kinases. There is no compensatory activation of other RTKs. Therefore, HER3 reactivation does not seem to be due to oncogene switching. Instead, the induction in HER3 expression is at least partially mediated through transcriptional upregulation of HER3 mRNA, a greater upregulation of total HER3 protein expression, an even greater upregulation of membrane HER3 expression, and a decreased rate of HER3 dephosphorylation [136]. Therefore, although HER TKIs inhibit HER2 catalytic kinase activity in a linearly dose-dependent manner, the HER2-HER3 complex signaling is endowed with a robust signal buffering capacity that sustains it against a nearly two-log inhibition of HER2

catalytic activity. Understanding and explaining this signal buffering mechanism including transcriptional feedback regulation to HER3 is a principal goal of our modeling effort in this project. Also, we have to validate the possibility that PI3K signaling may activate the epigenomic transcriptional repression mechanism and it will down-regulate elements of the HER2/3 signaling pathways, and model response of HER family inhibitors in tumors.

2.1.3 Approach and methodology

We will consider different aspects of treatment that influence treatment response: (a) rapid phosphoprotein mediated signaling and influences of PI3K mutations thereon and (b) long term transcriptional feedback regulation of HER3 and other proteins involved the rapid response. We will focus on the analysis of cancer cell as a complex biological system. For this, we develop and implement computational models of processes relevant to cancer prevention, diagnostics and therapeutics. The integration of experimental biology with mathematical modeling leads to new insights in cancer biology and innovative approaches to the management of cancer.

There is a growing body of dynamic models of HER signaling. All of these models are differential-equation based, either deterministic or stochastic ODEs. They provide insights into the operation of HER signaling, and are building blocks upon which other models can be based. We will build on existing models to develop models of the HER2/HER3 signaling network, to address the central questions outlined above: rapid phosphoprotein mediated signaling and influences of PI3K mutations thereon, influences of the microenvironment, and long term transcriptional feedback regulation of HER3. In order to do this, we will need to extend previous modeling efforts in several key directions, the details of which are outlined below:

- Late-response model: we will design models to help identify a largely unknown piece of the HER2/HER3 signaling networks involved in long term transcriptional feedback, and investigate the mechanisms behind the relatively slow dynamics of HER3 recovery in response to HER2 inhibition.

- Inference model: we will develop inference method to capture topological change of temporally evolving networks and understand short-term and long-term behavior of the HER2/HER3 signaling with a systems point of view.

- Reconstruction of gene regulatory network: we will develop new reconstruction method which can guarantee exact reconstruction of unknown piece of the HER2/HER3 signaling network or reveal deficiencies in the model.

The model will differ from existing work in important ways: it will exploit a mathematical separation of time scales for fast and slow dynamics, incorporate underlying genetic aberrations, and include parallel signaling. The inference method will be used to help understand the roles of cooperating genetic aberrations, transcriptional and translational regulation,

vesicle control and microenvironment in fast and slow dynamic processes. A combination of inference analysis, reconstruction method and dynamic modeling will be used to model the unexplored effects of modulation of transcription on HER2/3 signaling.

2.2 A Hybrid Control Systems Approach to Brain-Machine Interfaces for Exoskeleton Control

2.2.1 Background and significance

Our goal is to advance our understanding of fundamental principles in the neural control of movement in scenarios that involve physical interactions with the world. Furthermore, this work will transform neuroprosthetic systems to improve the quality of life for a large number of neurological patients. The central question that motivates this project is [29]:

> *"Does the brain use motor programs to help it control a highly redundant multi-degree of freedom (DOF) biomechanical plant such as the arm?"*

Here, motor programs are defined as short patterns of covariation between motor variables, where these patterns can recur in different behaviors at different amplitudes and times [92]. Sample motor variables are joint rotations [155][154] or the muscular electromyographic (EMG) activity acting on these joints. These programs, which are also referred to as time-varying postural or muscular synergies [50][117], are extracted from data using dimensionality reduction techniques such as non-negative matrix factorization [51][98]. A small number of programs have been reported for human reaching [53][52] and these programs also have kinematic analogues.

While motor programs are defined via statistical decomposition of movement data, Carmena's group has recently found evidence that they may also be encoded within the nervous system [113][114]. They measured forelimb EMG activity while monkeys made reaching and grasping movements, and simultaneously recorded cells in motor cortex. They found that there were large populations of cells sensitive to the recruitment parameters of motor programs, as well as the control of individual muscles. Most neurons were activated phasically, prior to motor program onset, suggesting that movements could be predicted by observing neural activity prior to program onset. This encoding framework is consistent with recent observations that motor cortical cells are well-tuned not only to instantaneous motor parameters such as movement direction, but also to entire movement trajectories [80][118][128] or the geometrically-simple curves into which they can be decomposed [144][145]. Moreover, intracortical microstimulation in rat [124], cat [156] and macaque [73][74] motor cortex can evoke complex multi-joint forelimb behaviors, via EMG patterns that Carmena's group has found can be decomposed into a small set of muscle synergies overlapping those seen in natural behavior [116].

The main goal of this project is to test the motor program framework in a setting that takes into consideration the strong coupling between brain, biomechanics and behavior. This

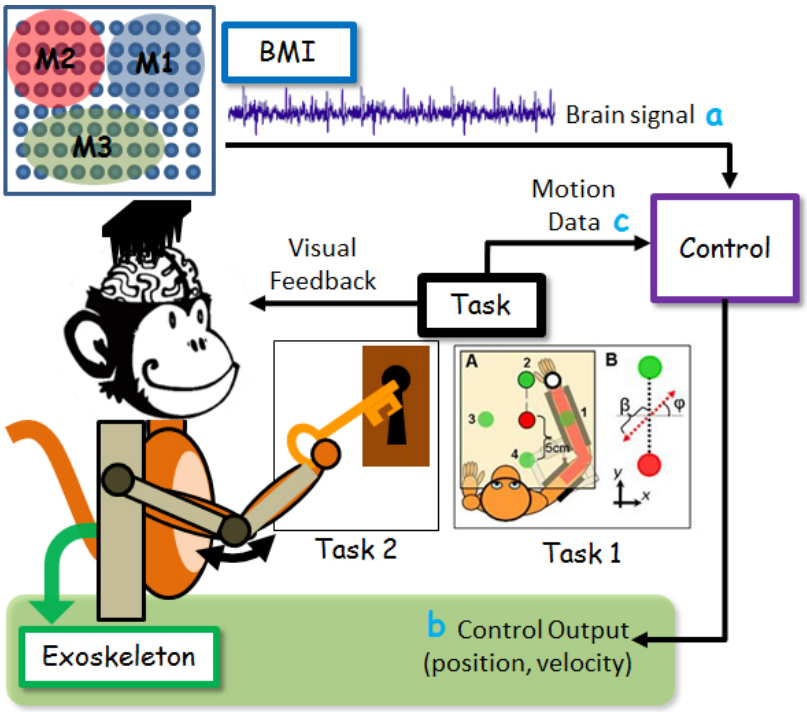


Figure 2.3: Block diagram of the different experimental and theoretical components of the system and cartoon of the behavioral tasks.

framework has generated several specific hypotheses, each of which we will test in a series of proposed experiments that require a combination of novel experimental, theoretical, and technical innovations. Together, we expect these will transform our understanding of the primate motor system, and disrupt current neuroprosthetic approaches to repairing it when damaged.

A hierarchical, distributed hybrid control system will enable system identification of discrete motor programs planned by the monkey, based on offline data analyses of neural recordings made while the animal makes reaching movements tracked by an exoskeleton. Combined with a musculoskeletal model [148][162], the hybrid control system will determine how the discrete motor programs are associated with the control of intrinsic joint rotations and forces, not just endpoint position. Used in an online control mode, the hybrid controller will serve not only to detect motor programs but to provide smooth, continuous prosthetic control. The ability to perform system identification of motor programs is basic to all components of this project, with incorporation of a musculoskeletal (MSK) model and an online control mode being additional capabilities to be programmed. Figure 2.3 represents a simplified schematic block diagram of the different experimental and theoretical components of the system.

2.2.2 Goals of the research

By the end of the project we expect to be able to synthesize all different innovations into a single new paradigm unifying brain, biomechanics, and behavior. Visually-cued motor plans in motor cortex will be: (a) read by a BMI, (b) interpreted by a hybrid controller and musculoskeletal model, and (c) translated into appropriate movements and stiffness in a multi-degree of freedom (DOF) upper-limb exoskeleton. The proposed research has the potential to dramatically impact the following fields:

- Neurophysiology of motor control: the outcome of this study will impact systems neuroscience by advancing our understanding of fundamental principles on the neural control of movement in scenarios that involve physical interactions in the world. Specifically we aim to elucidate principles of motor coordination that take into consideration the strong coupling between brain, biomechanics and behavior.

- Hybrid system identification and control: existing theoretical frameworks such as Kalman filtering with static decoder map dealt with dynamics, but have limitations. Here we propose a novel framework based on Hybrid System Identification and Control from which we can learn how the brain decomposes complex tasks into simpler tasks and discrete modes naturally. Also, we can provide a novel control scheme to perform a complex task and support this theory via experimental results. Moreover, the impact of this framework will drive the development of a method for identifying hybrid system models and design control schemes which provide an effective tool compromising between the richness and complexity.

- Neuroprosthetic systems: brain-machine interfaces that incorporate biomechanics in the loop will make a great impact on the quality of life for neurological patients by providing reliable performance while interacting with real objects and in real world scenarios. Specifically, this technology has the potential to impact millions of people suffering from spinal cord injuries, stroke and other neurological disorders. Moreover, the impact of this technology in the clinical realm will drive neural technology to the next level: human machine interaction, and augmentation of sensory, motor and cognitive capabilities in healthy subjects.

2.2.3 Approach and methodology

(a) Brain Machine Interface (BMI)

In a BMI system, the neural filter allows the experimenter to partially define the causality between brain activity and motor output, allowing movement intentions to be decoded without needing to account for the complex sensorimotor transformations embodied in the nervous system. Here, a novel form of filter accounting for cells' relationship to entire movement trajectories will be compared against conventional BMI decoders. As the neural filter determines the contribution of each recorded cell to movement kinematics, neuron dropping

can be used to investigate the effect of cells being lost within the sample available to the decoder, for example, due to small movements of the recording array. The influence of ongoing sensory feedback to the cells can be tested by tasks employing sensory gating of both visual and proprioceptive signals.

(b) Hybrid Control Scheme

The physical activity described in this study is dynamic limb motion within a BMI system. Our research seeks to understand and model a mapping between the brain and arm motion, to understand how the brain learns to control an exoskeleton. One of the key challenges is to develop a representation of the subject's intention including motion, speed of motion, and the involved forces and torques, enabling a clear control strategy, or set of strategies. Our technological focus in designing this representation will be a new kind of mathematical model integrating the continuous dynamics of motion with the modal dynamics of different poses or tasks.

Modeling techniques used today are largely based on continuous state dynamic models. As the number of state variables increase, though, to include variables representing the necessary information, such models quickly become intractable. Discrete state dynamic models have been used in the past to model large and complex systems; these models represent all configurations of the system as a finite set of behaviors, with dynamics modeled as simply switching from one behavior to another. While relatively simple, these models are often too abstract to capture important behavior. Hybrid systems provide an alternative, by combining both continuous and discrete state dynamics. These systems model changes in the system behavior as a continuous evolution interspersed with mode transition from one pose to another.

In this project, we consider a hybrid model with a large but finite number of discrete states. Each discrete state may represent a pose such as *"reaching to desired location"* or *"maintaining an endpoint condition"* or motor programs. A string of symbols with associated parameters can then be used to describe a sequence of motions, either repetitive stretching or dynamic motions between poses. The transitions between poses are constrained by the dynamic motion of the subject. In such a way, sequencing between modes forms a set of elementary motions, while the continuous states record the *"parameterization"* of each mode: the range of joint angles, the speed of motion, and time. Using this framework, we will develop a *"language"* of motions that may be used to describe the control of internal joint rotations and forces. This language requires relatively low bandwidth for representation and thus control, since a simple phrase describing each pose captures a large amount of information and describes motions that can be easily controlled using a series of simple linear controllers indexed to each mode.

The idea of using hybrid system models to represent dynamic motion has its origins in the computer vision literature [153]. The key novel ideas in this project are: 1) the development of a method for system identification of hybrid system models from primate data in the Carmena Lab and the use of such data to test the quality of our models; 2) the use of these

models to design control law robust to experimental contingencies experienced during online control of neuroprosthetic devices; and 3) the integration of the hybrid controller with an MSK model to enable skillful control of an exoskeleton.

We represent this process as two maps: the first is the mapping of information about the uncertain, internal and external environment of the agent into a top-level control decision. The high level controller will not only detect motor program but will coordinate actions and provide smooth transitions between modes. The second is the process that maps the top-level control decision to the sequence of control and coordination actions that enables a subject to complete a task or a sequence of tasks. Identifying models for this process requires research into a paradigm that provides for the seamless integration of probabilistic, discrete techniques with control of continuous systems.

Chapter 3

Modeling Response of Biological Signal Pathways using a Hybrid Boolean Framework

Mathematical models in systems biology are often constructed by either Ordinary Differential Equation (ODE) modeling or logical (Boolean) modeling. We develop a Hybrid Boolean Model (ODE+Boolean) for biological signal pathways with postulated epigenomic feedback which represents long term transcriptional feedback regulation of HER3. The basic idea in this model is to combine continuous dynamical systems (an ODE model for already well-known parts of the network) with a discrete transition system (Boolean for postulated but largely unknown components). We use the existing or well-known ODE model to "trigger" signal pathways represented by a Boolean model. This framework is easier to validate than a complete ODE model for large and complex signal pathways, for example to find unknown pathways which match the response to experimental data. The advantage of using a Boolean model for the unknown parts of the network is that relatively few parameters are needed. Thus, the framework avoids over-fitting, covers a broad range of pathways, and easily represents various experimental conditions. The overall goal of the hybrid model is to predict the behavior of biological signal pathways, thus helping to understand unknown parts of the pathway between experimental results and qualitative/quantitative results. Extensions are discussed, and numerical examples in biological systems as well as engineering example are provided. The material in this chapter is based on the work in [37].

3.1 Introduction

Mathematical models in systems biology are often constructed by either ordinary differential equation (ODE) modeling or logical (Boolean) modeling [18]. The ODE model can capture detailed "low-level" phenomena such as protein concentration and mass action kinetics [39][87]. However, the model is estimated by determining free parameters or unknown re-

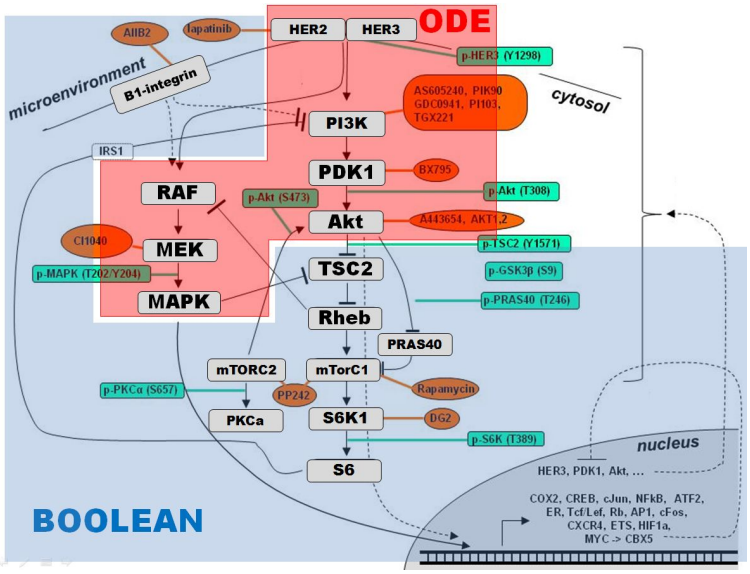
action rate constants using typically limited experimental data, even though large amounts of experimental data are required in order to avoid over-fitting. Moreover, ODE models typically require a good understanding of underlying dynamics. On the other hand, logical models such as Boolean Networks (BNs) seek qualitative rather than quantitative models of biological systems [132][106]. BNs can succeed in capturing "high-level" phenomena such as activation or deactivation with fewer parameters, so they can be used to evaluate model structure. However, since they cannot capture "low-level" dynamics for nonlinear systems such as biological signal pathways, such logical models cannot guarantee accuracy. In spite of this limitation, most research using this BN approach includes analysis of qualitative pathways or gene regulation networks because they are easy to simulate or evaluate [132][106][129]. In this chapter, we will introduce a Hybrid Boolean Model (ODE+Boolean) to combine the advantages of each model.

Previous work has focused on a hybrid model which uses simple continuous dynamics, lumps the complexity into the discrete inputs and gives us the capability to analyze the model mathematically [68][91][76]. Also, the use of hybrid (especially piecewise affine) models for the analysis and identification of biochemical regulatory networks are proposed [163][122][90][15]. Motivated by previous work, we develop a way to combine continuous dynamical systems (ODE, already well-known parts) with discrete transition systems (BN, unknown parts) to cover a broad range of signal pathways. We use distinct modes with Boolean variables which abstract the detailed nonlinear behavior of a system into piecewise continuous dynamics with discrete transitions from one mode to another.

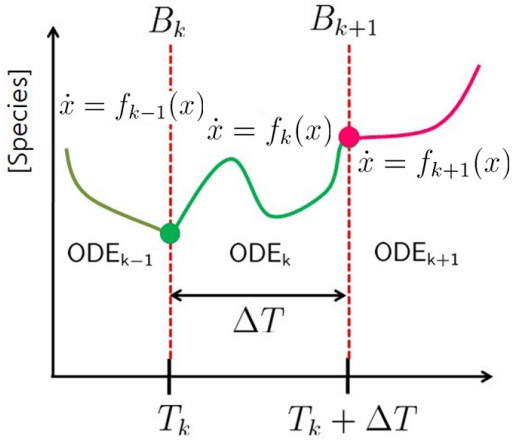
This framework is easy to validate and it is easy to include unknown pathways to match the response to the experimental data with only a few parameters. The strengths of this framework are that it:

- Avoids over-fitting using limited data (relatively few new parameters introduced);
- Covers a broad range of pathways and it is easy to simulate large networks with different stimulations and inhibitions (close to what biologists do in their experiments);
- Can help to understand the dynamics that underlie complex interwoven networks;
- Admits analyses such as phase portraits with Boolean variables which can help to understand qualitative dynamic behavior without simulation.

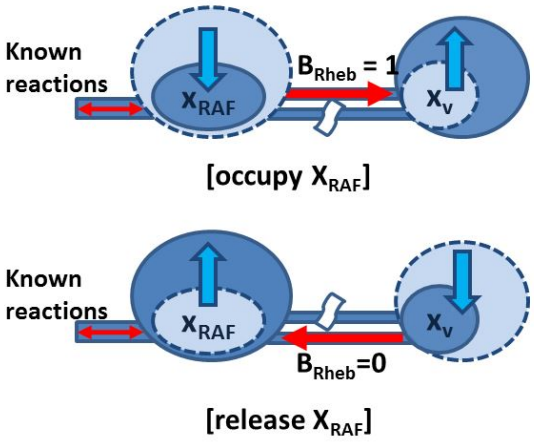
In this chapter, we study how we combine ODE models with Boolean networks in order to cover a broad range of signal pathways. For example, in the HER2 positive breast cancer network [72] that we study in Figure 3.1(a), we have a well-known ODE model for the upstream part of the gene regulatory network (red part) which triggers the unknown downstream part of the network (blue part, Boolean network) where only a few states can be measured. Also, these Boolean networks affect the known ODE model by interaction or feedback. We could build a full ODE model covering a broad range of signal pathways but it requires many measurements to estimate all parameters and avoid over-fitting. Here, we



(a)



(b)



(c)

Figure 3.1: (a) Conceptual diagram of the hybrid Boolean framework for the HER2/3 network in breast cancer (ODE: well-known model + Boolean network: unknown parts; arrow end: activation, dash end: inhibition) [72][137] (b) conceptual representation of trajectories in the hybrid boolean model (c) conceptual representation of "virtual state".

propose the framework which covers a broad range of pathways but which uses limited measurement data. Therefore, our proposed model might be a rather abstract representation compared with the full ODE model. Key contributions of this work include:

- Developing a way to construct a hybrid Boolean framework for biological signaling pathways and connect ODE to BN and BN to ODE;

- Demonstrating how we can estimate new parameters by introducing "virtual states";
- Providing a qualitative-quantitative modeling method and analysis of biological systems.

The rest of the chapter is organized as follows: we introduce the Hybrid Boolean Model in Section 3.2, and present parameter estimation for the proposed framework based on the Pontryagin Maximum Principle in Section 3.3. Also, we analyze our model and discuss possible applications with numerical examples in Section 3.4.

3.2 Hybrid Boolean Models and Simulation

A hybrid Boolean model represents a dynamical system with interacting continuous dynamics and discrete-event dynamics. Continuous dynamics arise as an ODE specifying how the mass action kinetics evolves over time. Discrete components encoded as a BN can be triggered by discrete switches encoding protein concentrations reaching unknown thresholds. Also, the BN can affect the dynamics of the known ODE model if the ODEs have interactions with the BN. For example, Figure 3.1(a) shows that [$\mathsf{Akt(ODE)}$] activates [$\mathsf{Rheb(BN)}$] which inhibits [$\mathsf{RAF(ODE)}$] (i.e., $\mathsf{Akt_{ODE}} \dashv (\mathsf{TSC2} \dashv \mathsf{Rheb})_{\mathsf{BN}} \dashv \mathsf{RAF_{ODE}}$). Therefore, there are two classes or events which connect the ODE to the BN and vice versa in our framework and these transitions between different dynamic systems need to be treated carefully.

3.2.1 Notation

Let $\{x(t) \in \mathbb{R}^{N_{\mathsf{ODE}}}\}_{T_k}^{T_{k+1}}$ denote the trajectory of the system state where N_{ODE} is the number of state variables for the ODE; suppose that it evolves according to the governing equation $\dot{x} = f_k(x,t)$, where T_k and T_{k+1} represent discrete time points for the BN; the dynamic model $f_k : \mathbb{R}^n \times [T_k, T_{k+1}] \to \mathbb{R}^n$ represents the system dynamics from T_k to T_{k+1}. In other words, at each discrete time T_k, a system transition could occur, and the system dynamics f_k as shown in Figure 3.1(b) could switch to another system f_{k+1}. The transitions are related to *virtual state* and *threshold* which are explained in the following section. Also, we denote the Boolean state by $B_p \in \{0,1\}$ where $p \in \{1,...,N_{\mathsf{BN}}\}$ and the ODE state by $x_q,\ q \in \{1,...,N_{\mathsf{ODE}}\}$ where N_{BN} and N_{ODE} represent the number of state variables for the BN and ODE model respectively. Here, the number of Boolean variables (N_{BN}) is not necessarily the same as the number of proteins represented by the BN, as our framework needs only "event related" Boolean variables, as we discuss below.

3.2.2 Virtual State Connecting BN to ODE

As mentioned above, there are two classes of variables in our framework: a *threshold* connecting ODE to BN and a *virtual state* connecting BN to ODE.

Definition (*Threshold*) A threshold Θ_{ij} determines the activation of the BN or simply the value of the Boolean state B_j which can affect the ODE based on the x_i.

$$B_j = \begin{cases} 1 & \text{if } x_i \geq \Theta_{ij} \\ 0 & \text{otherwise} \end{cases}$$

Definition (*Virtual state*) A virtual state x_{virtual} is introduced as a new state variable which is used to buffer values of another continuous state variable at the boundary of the ODE/BN, as described below.

Intuitively, the virtual state may be described as follows: imagine two reservoirs which are connected to each other with control valves in Figure 3.1(c). One represents a state variable in the ODE which interacts with the BN (i.e., x_{RAF} in Figure 3.1(a), here x denotes the state variable in the ODE) and the other is a virtual state (x_{virtual}). For example, if the value of x_{Akt} is greater than the given threshold (i.e., $x_{\mathsf{Akt}} \geq \Theta_{\mathsf{Akt,Rheb}}$), the BN, $x_{\mathsf{Akt}} \dashv B_{\mathsf{TSC2}} \dashv B_{\mathsf{Rheb}} \dashv x_{\mathsf{RAF}}$ in Figure 3.1(a) is activated. Then, [Rheb(BN)] inhibits [RAF(ODE)] so a certain amount of x_{RAF} is occupied by x_{virtual} and *blocked* in the known reaction as shown in Figure 3.1(c)(top). The amount of inhibited (*"pre-occupied"*) x_{RAF} which is stored in x_{virtual} can be *released* to x_{RAF} once B_{Rheb} is deactivated ($B_{\mathsf{Rheb}} = 0$) in Figure 3.1(c)(bottom). This *occupying* and *releasing* are explained with equations in the following section.

(a) Inhibition Reaction

In terms of biology and biochemistry, inhibitors are molecules that prevent or block the chemical reaction or function. Therefore, the inhibition process prevents the activation of a certain biochemical pathway by accumulating the key product of the pathway. In terms of modeling, we can introduce a new virtual state variable which occupies a certain state variable when the inhibition is triggered, and releases a preoccupied amount when the

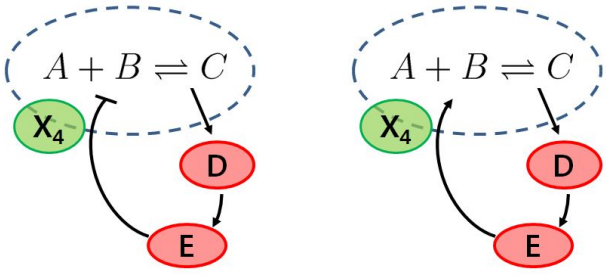


Figure 3.2: Simple examples (a) inhibition (dash end from E to B) and (b) activation (arrow end from E to B).

inhibition is deactivated (reversed) in equation (3.1). In other words, this virtual state can be used as a *reservoir* or *memory* to encode the transition information.

Consider the simple example in Figure 3.2 (a) where $x_{\mathsf{A}}, x_{\mathsf{B}}$ and x_{C} represent state variables in a simple ODE model (assuming known reaction). Also, B_{D} and B_{E} represent states in the BN, so when the concentration of x_{C} is above a certain threshold ($x_{\mathsf{C}} \geq \Theta_{\mathsf{C,E}}$, where $\Theta_{\mathsf{C,E}}$ is the threshold which determines the activity of the BN), the BN, $[\mathsf{C}] \to [\mathsf{D}] \to [\mathsf{E}] \dashv$ is activated, and it starts inhibition of x_{B} in the ODE model. On the other hand, when x_{C} is below the threshold ($x_{\mathsf{C}} < \Theta_{\mathsf{C,E}}$), the BN is deactivated, then a pre-occupied amount of x_{B} ($= x_{\mathsf{virtual}}$) is released through the reverse reaction because most of the biological reactions are reversible. Therefore, the threshold is the amount of x_{C} which can affect the ODE model by triggering BN feedback (i.e., $B_{\mathsf{E}} = 0$ or 1). Here we formulate the inhibition reaction from BN by introducing a virtual state, x_{virtual}, as shown below:

$$\begin{bmatrix} \mathsf{If}\ x_{\mathsf{C}} \geq \Theta_{\mathsf{C,E}} & \to B_{\mathsf{E}} = 1\ (\mathsf{occupy}\ x_{\mathsf{B}}) \\ \mathsf{If}\ x_{\mathsf{C}} < \Theta_{\mathsf{C,E}} & \to B_{\mathsf{E}} = 0\ (\mathsf{release}\ x_{\mathsf{B}}) \end{bmatrix} \Longleftrightarrow \begin{bmatrix} x_{\mathsf{B}} \xrightarrow{k_1} x_{\mathsf{virtual}} \\ x_{\mathsf{B}} \xleftarrow[k_2]{} x_{\mathsf{virtual}} \end{bmatrix}$$

$$\begin{aligned} \dot{x}_{\mathsf{B}} &= -k_1 x_{\mathsf{B}} B_{\mathsf{E}} + k_2 x_{\mathsf{virtual}} \overline{B}_{\mathsf{E}} \\ \dot{x}_{\mathsf{virtual}} &= k_1 x_{\mathsf{B}} B_{\mathsf{E}} - k_2 x_{\mathsf{virtual}} \overline{B}_{\mathsf{E}} \end{aligned} \tag{3.1}$$

where B_{E} represents the Boolean state of $[\mathsf{E}]$ and x_{B}, x_{virtual} represent states of $[\mathsf{B}]$ and $[\mathsf{virtual\ state}]$ in the ODE model respectively. Without loss of generality, the inhibition reaction between virtual state and known state (i.e., x_{B}) could be any nonlinear dynamics but here, we assume a simple linear model for the inhibition reaction. Also, we can guarantee mass conservation because $\dot{x}_{\mathsf{B}} + \dot{x}_{\mathsf{virtual}} = 0$ in equation (3.1).

(b) Activation Reaction

An activation in biology or biochemical sciences generally refers to the process whereby something is prepared or excited for a subsequent reaction. In other words, an activation is the process which helps in developing a certain biochemical pathway or reaction. For example, in cellular signaling pathways, a reversible phosphorylation of proteins is an important regulatory mechanism. Enzymes called kinases (phosphorylation) and phosphatases (dephosphorylation) are involved in this process. Many enzymes and receptors are switched "on" or "off" by phosphorylation and dephosphorylation. A reversible phosphorylation results in a conformational change in the structure in many enzymes and receptors, causing them to become activated or deactivated.

Modeling for activation is parallel to modeling for inhibition but we have to take the initial condition into consideration for the activation process. For inhibition cases, the system dynamics is the same as without the BN when the signal pathway is not triggered (i.e., $B_{\mathsf{E}} = 0$ if $x_{\mathsf{c}} < \Theta_{\mathsf{C,E}}$ in equation (3.1)) since the BN could affect the ODE model after triggered. For the activation processes, we modify the chain as two consecutive inhibition processes by

using Boolean algebra. For example, consider the simple example case in Figure 3.2 (b). The only difference between the inhibition in Figure 3.2 (a) and this is that B_{E} activates x_{B}. When x_{C} is above a certain threshold, the BN is activated and it activates or increases x_{B} in the ODE model. On the contrary, when x_{C} is below this threshold, the BN is deactivated and the reverse reaction occurs. We introduce $B_{\bar{\mathsf{E}}}$ between x_{B} and B_{E} as "$B_{\mathsf{E}} \to x_{\mathsf{B}}$"$\Longleftrightarrow$ "$B_{\mathsf{E}} \dashv B_{\bar{\mathsf{E}}} \dashv x_{\mathsf{B}}$":

$$\begin{bmatrix} \text{If } \mathsf{x_C} \geq \Theta_{\mathsf{C,E}} \to \mathsf{B_E} = 1,\ \mathsf{B_{\bar{E}}} = 0 \text{ (increase } \mathsf{x_B}) \\ \text{If } \mathsf{x_C} < \Theta_{\mathsf{C,E}} \to \mathsf{B_E} = 0,\ \mathsf{B_{\bar{E}}} = 1 \text{ (decrease } \mathsf{x_B}) \end{bmatrix} \Longleftrightarrow \begin{bmatrix} \mathsf{x_B} \xleftarrow{k_1} \mathsf{x_{virtual}} \\ \mathsf{x_B} \xrightarrow[k_2]{} \mathsf{x_{virtual}} \end{bmatrix}$$

$$\begin{aligned} \dot{x}_{\mathsf{B}} &= -k_1 x_{\mathsf{B}} B_{\bar{\mathsf{E}}} + k_2 x_{\mathsf{virtual}} \overline{B}_{\bar{\mathsf{E}}} \\ \dot{x}_{\mathsf{virtual}} &= k_1 x_{\mathsf{B}} B_{\bar{\mathsf{E}}} - k_2 x_{\mathsf{virtual}} \overline{B}_{\bar{\mathsf{E}}} \end{aligned} \tag{3.2}$$

Therefore, the main difference between inhibition processes (3.1) and activation processes (3.2) is the "triggering behavior". For the activation case in equation (3.2), the virtual state can be increased (prepared) before triggering (i.e., $B_{\bar{\mathsf{E}}} = 1$ if $x_{\mathsf{C}} < \Theta_{\mathsf{C,E}}$) which is exactly consistent with the biological definition, for example, an activation reaction in biology refers to the process whereby something is prepared.

(c) Multiple Pathways

In this section, we present how to handle multiple pathways by showing a simple example (Figure 3.1(a)); there are multiple pathways (i.e., $\mathsf{Akt} \dashv \mathsf{TSC2} \dashv \mathsf{Rheb} \to \mathsf{mTorC1} \to \mathsf{S6K1} \to \mathsf{S6} \dashv \mathsf{PI3K}$ and $\mathsf{Akt} \dashv \mathsf{TSC2} \dashv \mathsf{Rheb} \dashv \mathsf{RAF}$). Here, we only need two Boolean variables (B_{S6}, B_{Rheb}) and two virtual states (for $x_{\mathsf{PI3K}}, x_{\mathsf{RAF}}$ respectively). They include abstract information about the pathways which are denoted by BNs. Therefore, the only thing we have to do is finding a Boolean network which affects ODE by using Boolean algebra. Moreover, we typically need only a few virtual states at the point connecting from BN to ODE or "event related" Boolean variables which is the end point of Boolean network and affects ODE (i.e., $B_{\mathsf{S6}}, B_{\mathsf{Rheb}}$).

3.3 Parameter Estimation

The Hybrid Boolean model introduces new parameters such as kinetic constants $k_1,\ k_2$ in equation (3.1) and (3.2) and threshold ($\Theta_{\mathsf{C,E}}$) which are unknown and only weakly constrained by experimental knowledge. For example, a threshold should be between 0 and maximum of concentration. A crucial problem for this framework is that these parameters are difficult to estimate because they are correlated with each other in the system under consideration. In this case, there might be many local optima and the parameters might be poorly determined because of lack of data, or there might be ambiguities brought by

redundancy in the system. However, we can reformulate the hybrid Boolean framework as a nonlinear optimal control problem and then identify parameters with virtual state variables and Boolean variables.

3.3.1 Reformulation and Nonlinear Switched Dynamical System

We can reformulate the hybrid Boolean model in a simple but abstract way. Consider the example in Figure 3.2(a) again and suppose that we know the dynamics of [A] [B] and [C] with known reaction rates k and k_r as follows:

$$\begin{aligned} \dot{x}_{\mathsf{A}} &= -kx_{\mathsf{A}}x_{\mathsf{B}} + k_r x_{\mathsf{C}} \\ \dot{x}_{\mathsf{B}} &= -kx_{\mathsf{A}}x_{\mathsf{B}} + k_r x_{\mathsf{C}} \\ \dot{x}_{\mathsf{C}} &= kx_{\mathsf{A}}x_{\mathsf{B}} - k_r x_{\mathsf{C}} \end{aligned} \tag{3.3}$$

We introduced virtual states and new reactions in equation (3.1) with unknown reaction rates k_1 and k_2. Define $\mathbf{x} = [x_{\mathsf{A}}\ x_{\mathsf{B}}\ x_{\mathsf{C}}\ x_{\mathsf{virtual}}]^T = [x_1\ x_2\ x_3\ x_4]^T$ where x_4 is a virtual state. Then, we can combine BN (3.1) and known ODE (3.3) by formulating a hybrid Boolean model as shown below:

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} -kx_1x_2 + k_r x_3 \\ -kx_1x_2 + k_r x_3 \\ -k_1x_2B_{\mathsf{E}} + k_2x_4\overline{B}_{\mathsf{E}} + kx_1x_2 - k_r x_3 \\ k_1x_2B_{\mathsf{E}} - k_2x_4\overline{B}_{\mathsf{E}} \end{bmatrix} = \begin{bmatrix} -kx_1x_2 + k_r x_3 \\ -kx_1x_2 + k_r x_3 \\ kx_1x_2 - k_r x_3 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -x_2 & x_4 \\ 0 & 0 \\ x_2 & -x_4 \end{bmatrix} \begin{bmatrix} k_1B_{\mathsf{E}} \\ k_2\overline{B}_{\mathsf{E}} \end{bmatrix} \\ &\triangleq f(\mathbf{x}) + \begin{bmatrix} g_1(\mathbf{x}) & g_2(\mathbf{x}) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} \end{aligned} \tag{3.4}$$

where each u_i is a piecewise continuous function (i.e. $u_1 = k_1B_{\mathsf{E}} \in \{0, k_1\}$, $u_2 = k_2\overline{B}_{\mathsf{E}} \in \{0, k_2\}$). Here, B_{E} and $\overline{B}_{\mathsf{E}}$ are Boolean states of [E] which are either 0 or 1 exclusively. Also, we can nicely decouple the hybrid Boolean model ([ODE] + [BN]) as $f(\mathbf{x})$ (known dynamics)

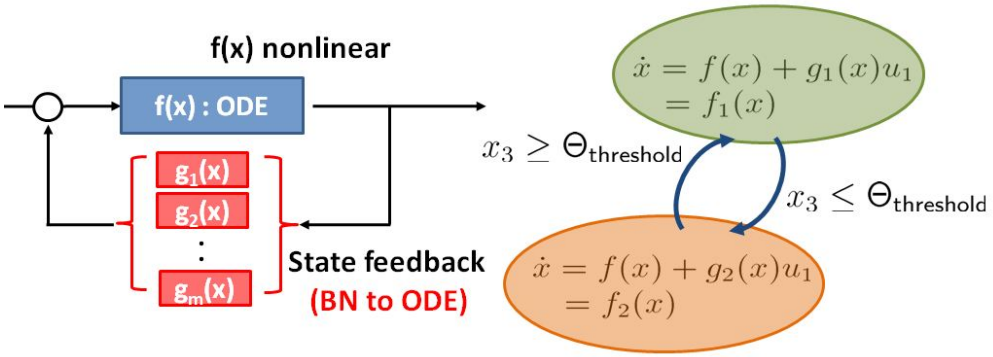


Figure 3.3: Nonlinear switched dynamical system with state feedback.

and given state feedback $g(\mathbf{x})(\mathsf{BN})$ as shown in Figure 3.3 and then, the parameter estimation problem reduces to an optimal control problem in constrained nonlinear switched dynamical systems. Moreover, each optimal input u_i includes new parameters such as thresholds and reaction rates in an abstract way. Once we find the optimal control input, we could use this to determine such information.

3.3.2 The Pontryagin Maximum Principle [57]

By reformulation, we can now formulate the parameter estimation as a control problem. For parameter estimation problems, the objective is to minimize the squared error between experimental data and simulated data from our model. Our attention will be focused on the following optimal control problem: minimize with respect to all piecewise continuous functions of time $u(\cdot)$, the final time T and the performance criterion is as follows:

$$\min_{u(\cdot)} J(\mathbf{u}, T) = \min_{u(\cdot)} \phi(\mathbf{x}(T), T) + \int_{t_0}^{T} L(\mathbf{x}, \mathbf{u}, t)dt \tag{3.5}$$

which is subject to the nonlinear dynamic system as follows:

$$\dot{\mathbf{x}} = h(\mathbf{x}, \mathbf{u}, t) = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}(t), t \in [0, T] \tag{3.6}$$

where $f(\mathbf{x})$, $g(\mathbf{x})$ are continuously differentiable in $\mathbf{x}$ and continuous in t, and that the same is true for the terminal cost ϕ. Also, L is denoted as follows:

$$L(\mathbf{x}, \mathbf{u}, t) = \sum_{i=1}^{N_d} q(i)\|x_i(t) - x_{i,d}(t)\|^2 + \mathbf{u}(t)^T R\mathbf{u}(t) \tag{3.7}$$

where N_d is the number of observable species from experimental data and the other constraints are:

$$u_i(t) \in \{0, k_j\}, \quad t \in [0, T] \tag{3.8}$$
$$x_j(t) \geq 0, \quad t \in [0, T] \tag{3.9}$$

where k_j are new parameters which we have to estimate and $x_j(t)$ represents the concentration of species which is non-negative. We can drop the last constraint (3.9) because we use a mass action kinetics in which mass conservation holds. Also, we can apply the linear programming relaxation of a 0-1 integer program here. In mathematics, the linear programming relaxation of a 0-1 integer program is the problem that arises by replacing the constraint that each variable must be 0 or 1 by a weaker constraint, that each variable belong to the interval $[0, 1]$. This relaxation technique transforms an NP-hard optimization problem into a related problem that is solvable in polynomial time. Therefore, we can apply the same relaxation technique:

$$0 \leq u_i(t) \leq k_j, \quad t \in [0, T] \tag{3.10}$$

Since we do not know the upper limit k_j, we apply a penalty for each input $u_i(t)$ with the resulting relaxation as shown below:

$$0 \leq u_i(t), \quad t \in [0, T] \tag{3.11}$$

Also, we can penalize for exclusiveness of each input in (3.4) because $u_1 = k_1 B_{\mathsf{E}}$ and $u_2 = k_2 \bar{B}_{\mathsf{E}}$ should be exclusive by our definition in equation (3.4). For example, if $B_{\mathsf{E}} = 1$, then $u_1 = k_1$ but $u_2 = 0$. Here, we use a relaxation technique which allows us to solve the optimization problem efficiently but results in our solution not being exactly the same as that of the original integer programming formulation in equation (3.8). For example, our solution might not be fully consistent with the threshold-crossing feedback or sharp transition, for example, "on/off" behavior. Thus, we might not provide clear estimation of thresholds, switching time and parameters because of loss of information by simplifying these processes but our model could still be used to approximate continuous phenomena by concatenating different models using discrete variables.

We are now ready to formulate *the Maximum Principle* [57]. First, we can define Lagrange multipliers to adjoin the constraints to the performance index. Since the constraints are determined by the differential equation (3.6), an associated multiplier or co-state $\mathbf{p}(t)$ is a function of time and equal to the gradient of the optimal value function evaluated along the optimal trajectory. Thus, the augmented performance index is given by:

$$\phi(\mathbf{x}(T), T) + \int_0^T \{L(\mathbf{x}, \mathbf{u}, t) + \mathbf{p}^T(t)(f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t) - \dot{\mathbf{x}}(t))\}dt \tag{3.12}$$

The Maximum Principle can be written in a more compact and symmetric form with the help of the Hamiltonian function:

$$H(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) = L(\mathbf{x}, \mathbf{u}, t) + \mathbf{p}^T(t)(f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t)) \tag{3.13}$$

and we can integrate (3.12) by parts and consider the equation under the calculus of variations with respect to $\mathbf{x}(t)$ and $\mathbf{u}(t)$:

$$\phi(\mathbf{x}(T), T) - \mathbf{p}^T(T)\mathbf{x}(T) + \mathbf{p}^T(0)\mathbf{x}(0) + \int_0^T [H(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) + \dot{\mathbf{p}}^T(t)\mathbf{x}(t)]dt \tag{3.14}$$

We consider the above equation in the calculus of variations with respect to $\mathbf{x}(t)$ and $\mathbf{u}(t)$ as follows:

$$(\frac{\partial \phi}{\partial \mathbf{x}} - \mathbf{p}^T)\delta\mathbf{x} \mid_{t=T} + \mathbf{p}^T \delta\mathbf{x} \mid_{t=0} + \int_0^T \{(\frac{\partial H}{\partial \mathbf{x}} + \dot{\mathbf{p}}^T)\delta\mathbf{x} + \frac{\partial H}{\partial \mathbf{u}}\delta\mathbf{u}\}dt \tag{3.15}$$

For a stationary point, the following conditions should be satisfied for all allowable variations:

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \mathbf{p}} = h(\mathbf{x}, \mathbf{u}) \tag{3.16}$$

$$-\dot{\mathbf{p}} = \frac{\partial H}{\partial \mathbf{x}} = \frac{\partial h}{\partial \mathbf{x}}^T p + \frac{\partial L}{\partial \mathbf{x}} \tag{3.17}$$

$$\mathbf{u}^*(t) = \arg\min_{\mathbf{u}} H(\mathbf{x}^*, \mathbf{u}, \mathbf{p}^*, t) \tag{3.18}$$

and the boundary condition is as follows:

$$\mathbf{p}(T) = (\frac{\partial \phi}{\partial \mathbf{x}})^T \mid_{t=T} \tag{3.19}$$

The Lagrange multiplier $\mathbf{p}(t)$ is a dynamic variable and it evolves backward in time with the final condition $\mathbf{p}(T)$. In contrast to *the Hamilton Jacobi Bellman (HJB)* solution to the infinite horizon optimal control problem, the Euler-Lagrange solution is explicitly solved for as a function of time $\mathbf{u}(t)$, not as a feedback law. Also, the resulting optimal trajectory is only valid for the specified initial condition $\mathbf{x}(t_o)$. Since the Euler-Lagrange equations specify the conditions for the existence of a stationary point, they represent necessary conditions for an optimal trajectory.

3.3.3 Numerical Examples

(a) Example 1

Consider the simple example in the previous section (equation (3.3) where $k = 0.01$, $k_r = 0.005$). First, we simulate our model in equation (3.3) under the following constraints given for the BN. We then use the synthesized data as "experimental data" to estimate parameters using the proposed method:

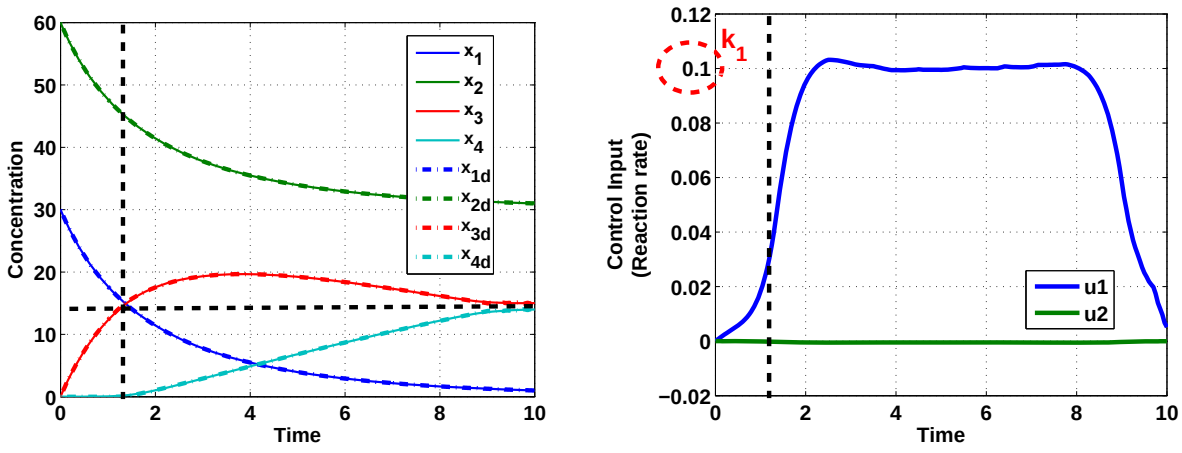
Boolean Network:	
if $x_{\mathsf{C}} \geq T_{\mathsf{C}} = 15$: $k_1 = 0.1$ $k_2 = 0.0$	[Inhibition]
if [BN triggered] and $x_{\mathsf{C}} < T_{\mathsf{C}} = 15$	
$k_1 = 0.0$ $k_2 = 0.1$	[Release]
otherwise	
$k_1 = 0.0$ $k_2 = 0.0$	

Figure 3.4(a) shows simulated (desired, dotted line) data and we use these as our experimental data. Figure 3.4(b) shows the estimated input u^* (i.e., $u_1 = k_1 B_{\mathsf{E}}, u_2 = k_1 \bar{B}_{\mathsf{E}}$) from which we can extract useful information such as $k_1 (= 0.1)$ and triggering point (threshold, $T_{\mathsf{C}} \approx 15$). Obviously, since we use relaxation in equation (3.10) and (3.11), our solution might be a local minimum.

(b) Example 2

Consider another example with different constraints for the BN as shown below:

Boolean Network (with hysteresis):	
if $x_{\mathsf{C}} \geq 20 = T_1$	
$k_1 = 0.2$ $k_2 = 0.0$	[Inhibition]
if [BN triggered] and $x_{\mathsf{C}} < 15 = T_2$	
$k_1 = 0.0$ $k_2 = 0.1$	[Release]
otherwise	
$k_1 = 0.0$ $k_2 = 0.0$	



(a) (b)

Figure 3.4: Simulated and estimated result for example 1: (a) input data ($\mathbf{x}_d$), estimated data ($\mathbf{x}$) and estimated threshold (T_c) (b) estimated parameter k_1.

Figure 3.5(a) and (c) show simulated data for the ODE with BNs using different initial conditions. If we use the data in Figure 3.5(a), we find that u^* is close to zero (within the numerical error) in Figure 3.5(b), obviously meaning that there is no triggering signal to the downstream pathway. Then, we can extract only one piece of information (i.e., $T_C \geq 20$). If we had another data set (Figure 3.5(c)), we could get more useful information as shown in Figure 3.5(d) because the data includes dynamic behavior from downstream of BN. We can infer the Boolean state based on the virtual state x_4 which tells us not only new parameters k_1 and k_2 but also when the transition occurs (i.e., threshold). Since we use relaxation in equation (3.10) and (3.11), estimated parameters are not the same as true parameters but they are close to the values as shown in Figure 3.5(d).

In general, we have several datasets because of different conditions or control of experiments such as different initial conditions or different drug combinations. First, we can find the optimal control input (i.e., feasible parameter sets P_1, P_2, P_3, ...) for each data set (D_1, D_2, D_3,...). Then, we can specify the feasible set (i.e., $P^* = P_1 \cap P_2 \cap P_3 \cap ...$) based on finding the intersection of all feasible parameter sets from different datasets. Therefore, as the number of experimental datasets increases, we would expect our parameter set to approach the true parameter set.

(c) Application in engineering (Parallel Parking)

A parameter estimation for the hybrid Boolean framework can be used for finding a locally optimal sequence of modes and corresponding switching times between modes in hybrid systems. For example, parallel parking requires turning the steering wheel at certain points and releasing it at others, like transitions in a hybrid system (Figure 3.6(a)(b)). We formulate

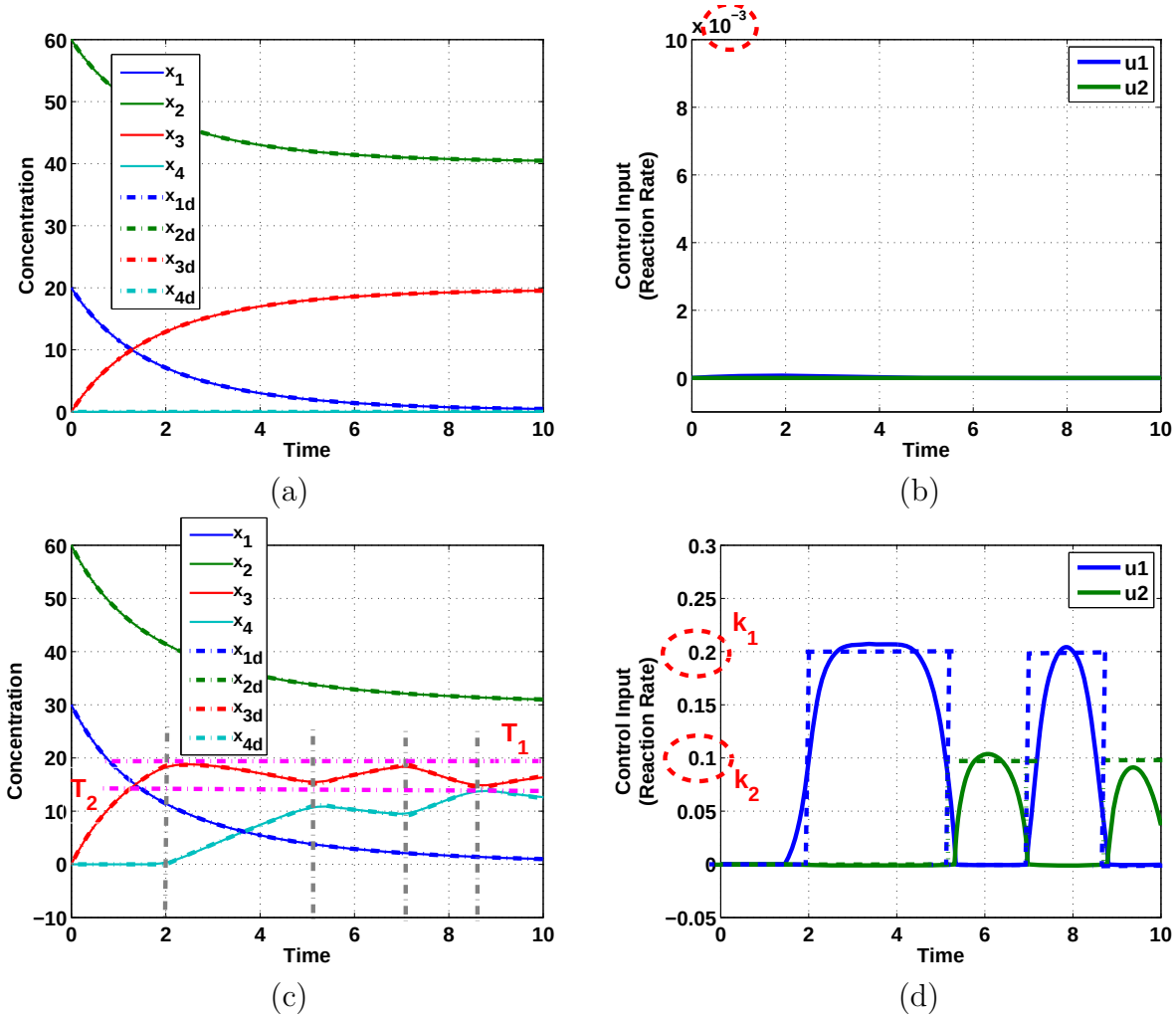
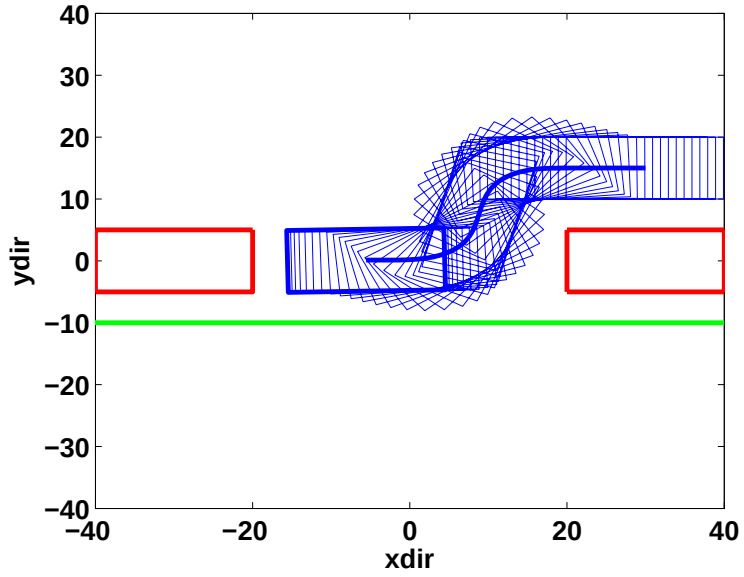
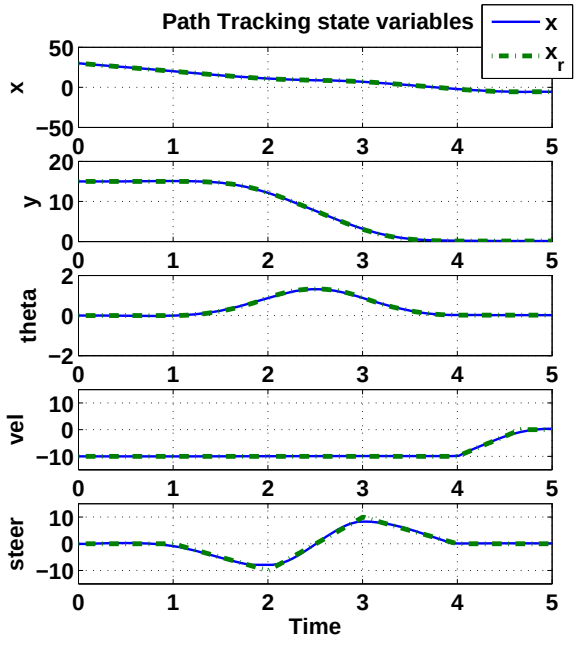


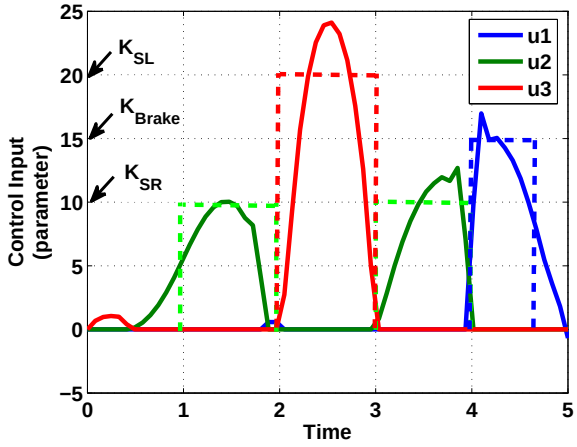
Figure 3.5: Simulated and estimated result for example 2: (a)(b) BN is not triggered (c)(d) BN is triggered at a different initial condition. (a) Input data and estimated data (not triggered) (b) estimated parameter (c) input data and estimated data where T_1 and T_2 represent thresholds respectively (d) estimated parameter k_1 and k_2.



(a)



(b)



(c)

Figure 3.6: Parallel parking example: (a)(b) (dotted line) x, y, θ are desired trajectories and (solid line) trajectories are generated by the input u^* (c) optimal mode sequence and switching time based on input u^*.

this problem based on our framework and here, we assume that we can use brake and steering wheel only in order to make the problem simple. Suppose that we have a desired trajectory (x, y, θ in Figure 3.6(b)) for parallel parking but we do not know the control input/sequence ($vel, steer$ in Figure 3.6(b)) as we don't know the amount of virtual state. A simple unicycle model is expressed as shown below and a cost function penalizes error $\|x - x_d\|$, $\|y - y_d\|$ and $\|\theta - \theta_d\|$:

$$X = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \\ v(t) \\ \phi(t) \end{bmatrix}, f(X) = \begin{bmatrix} v(t)\cos\theta(t) \\ v(t)\sin\theta(t) \\ \frac{1}{L}v(t)\tan\phi(t) \\ 0 \\ 0 \end{bmatrix}, g(X) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ K_{brake} & 0 & 0 \\ 0 & K_{SR} & K_{SL} \end{bmatrix} \quad (3.20)$$

where K_{brake} is 15, K_{SR} is 10 and K_{SL} is 20 in our example. By applying our parameter estimation method, we can get a locally optimal mode sequence and switching time sequence with estimated parameters as shown in Figure 3.6(c), close to true parameter values.

3.4 Analysis of Hybrid Boolean model

In this section, we will compare and discuss the hybrid Boolean model and ODE model.

3.4.1 Comparison Between Hybrid Boolean Model and ODE Model

So far, we have developed a hybrid Boolean model and represented how to estimate parameters. It is reasonable to compare this framework with a pure ODE model, in terms of model size and ease of estimating parameters. The hybrid Boolean model uses distinct modes with Boolean variables which abstract the detailed nonlinear behavior of a system with discrete transitions from one mode to another. Therefore, the model could provide a useful abstraction in understanding physical behavior. Also, it could provide a certain foundation for modeling and understanding biological systems at desired levels of approximation or abstraction. For example, large and multi-scale dynamical systems can be simplified by replacing certain state variables for relatively slow dynamics as piecewise constant approximations. In general, the assumption of mass action kinetics is that there is a large number of molecules which are homogeneously mixed. However, these assumptions may fail inside a cell because there are only a few molecules governing the reaction. These dynamics are best modeled using discrete transitions. The other thing we have to consider is steady state. We introduce new states and reactions which are interconnected with existing states so equilibrium points for the new model are affected by these interactions as follows:

[No interaction (ODE only)]

$$\dot{x}_i = f_i(\mathbf{x})$$

[Inhibition(or Activation)(BN) : $\mathsf{x_i} \dashv \mathsf{x_j}$ (or $\mathsf{x_i} \rightarrow \mathsf{x_j}$)]

$$\begin{aligned} \dot{x}_i &= f_i(x) - \eta(x_i, x_j) + \psi(x_{\mathsf{virtual}}) \\ \dot{x}_{\mathsf{virtual}} &= \eta(x_i, x_j) - \psi(x_{\mathsf{virtual}}) \end{aligned}$$

where each $f_i(\mathbf{x})$ represents the known ODE model. Therefore, if there is no interaction, which means $\eta(\cdot) = \psi(\cdot) = 0$, an equilibrium point satisfies $f_i(\mathbf{x}_{eq}) = 0,\ \forall i$. On the other hand, for the hybrid Boolean model, an equilibrium point should satisfy $f_i(\mathbf{x}_{eq}) = 0,\ \forall i$ with constraints such as $\eta(x_i, x_j) - \psi(x_{\mathsf{virtual}}) = 0$. Also we can formulate biological reactions in a general form as shown below:

$$\dot{x} = f(x, y, z, w, ...) + \sum_{i=1}^{m} k_{ir} z_i - \sum_{i=1}^{m} k_i x y_i + \sum_{j=1}^{n} \gamma_j w_j - \sum_{l=1}^{o} \eta_l x$$

where $f(x, y, z, w, ...)$ represents known dynamics or existing ODE and z_i represents the concentration of the complex XY_i so the second and the third terms in the RHS represent the reaction in which X participates, the fourth term represents inflow into X and the last term represents outflow from X. Here, the inflow and outflow represent a cascade type signal pathway. Therefore, an equilibrium point satisfies $f(x, y, z, w, ...) = 0$ with constraints such as $\sum_{i=1}^{m} k_{ir} z_i - \sum_{i=1}^{m} k_i x y_i + \sum_{j=1}^{n} \gamma_j w_j - \sum_{l=1}^{o} \eta_l x = 0$, so the interactions from BN to ODE affect the behavior of system dynamics.

3.4.2 Switching Surface

Hybrid Boolean models deal with nonlinear effects by invoking discrete mode switching when certain states reach or exceed threshold values. These thresholds can be represented as switching surfaces in the phase plane which invoke discrete mode switching. Therefore, a phase plane of a hybrid Boolean model shows a multi-dimensional space with different system dynamics. Consider a simple hybrid model as follows:

$$\begin{aligned} \dot{x}_1 &= -k_1 x_1 B_1 + k_{1r} x_1 \bar{B}_1 \\ \dot{x}_2 &= -k_2 x_2 B_2 + k_{2r} x_2 \bar{B}_2 \end{aligned} \tag{3.21}$$

where $B_1 = 1$ if $x_1 \geq T_1 > 0$ and $B_2 = 1$ if $x_2 \geq T_2 > 0$ and each $x_i \geq 0$ because the states in biochemical reactions represent concentrations which are nonnegative. Also, here x_1 and x_2 are decoupled with each other but in general, these could be coupled in a complex way. Figure 3.7 shows a phase plane with Boolean variables. In this example, we have two Boolean variables so there are four cases which are (BN : 00, 01, 10, 11); one stable system (11), two saddle points (01,10) and one unstable system (00). Consider point $(x_1, x_2) = (T_1, T_2)$. Each threshold (solid line) can be used as a sliding mode surface which attracts trajectory flow

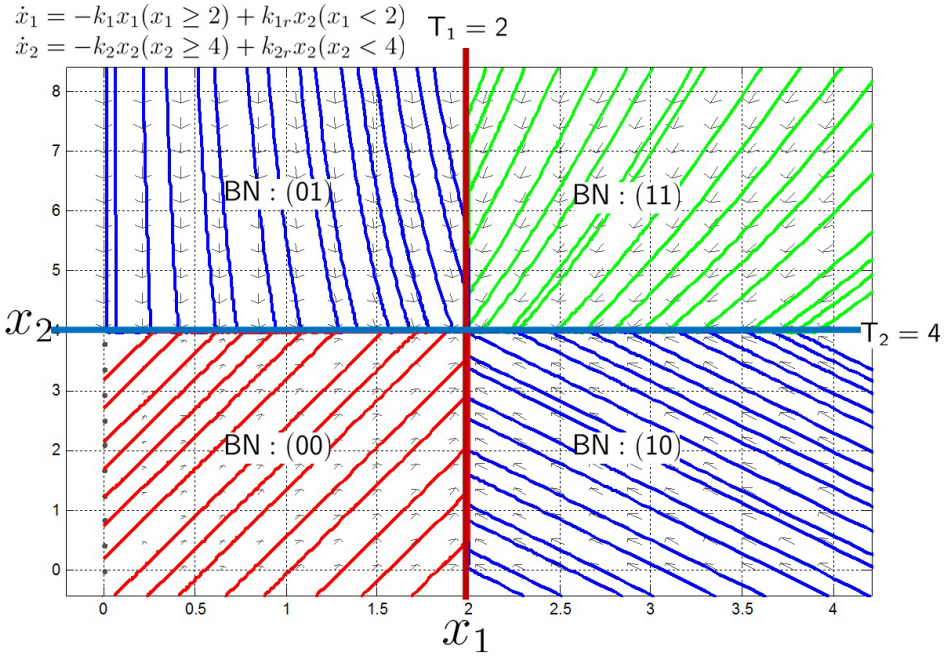


Figure 3.7: Phase portrait of a simple hybrid model.

$\phi_t(x)$ and makes all trajectories flow toward (T_1, T_2). Define $S_1 = x_1 - T_1$ which is a sliding surface if it satisfies the condition $S_1 \dot{S}_1 < 0$:

$$\begin{aligned} S_1\dot{S}_1 &= (x_1 - T_1)\dot{x}_1 = (x_1 - T_1)(-k_1 x_1 B_1 + k_{1r} x_1 \bar{B}_1) \\ \text{if } x_1 &> T_1,\ S_1\dot{S}_1 = (x_1 - T_1)(-k_1\ x_1) < 0 \\ \text{if } x_1 &< T_1,\ S_1\dot{S}_1 = (x_1 - T_1)(+k_{1r} x_1) < 0 \end{aligned}$$

Similarly, we can prove $S_2 = x_2 - T_2$ is also a sliding mode surface. Therefore, we have two equilibrium points of the system which are $(0, 0)$ and (T_1, T_2) in equation (3.21). Moreover, we can consider a "drug model" with this simple model. In biological experiments, small molecule inhibitors may be used as drugs, usually these are replenished every day in order to make the drug concentration constant $x_{drug} \approx C$. Therefore, we can combine the drug model which inhibits x_1 as $\dot{x}_1 = -k_{drug} x_1 x_{drug} \approx -k_{drug} C x_1 \approx -k'_{drug} x_1$ and consider the phase plane with Boolean variable for different concentration of drug (different k'_{drug}) as shown in Figure 3.8. The graphical representation shows that the sliding surface cannot attract the flow of dynamics if enough drug is used. However, if we put a small amount drug, flows are still captured by the sliding surface. Therefore, intuitively, we can consider this Boolean variable as a barrier and we can calculate the amount of drug which can overcome this barrier or change the domain of attraction. This graphical representation could be extended to multiple dimensions and could be useful to understand or gain intuition about complex biological networks.

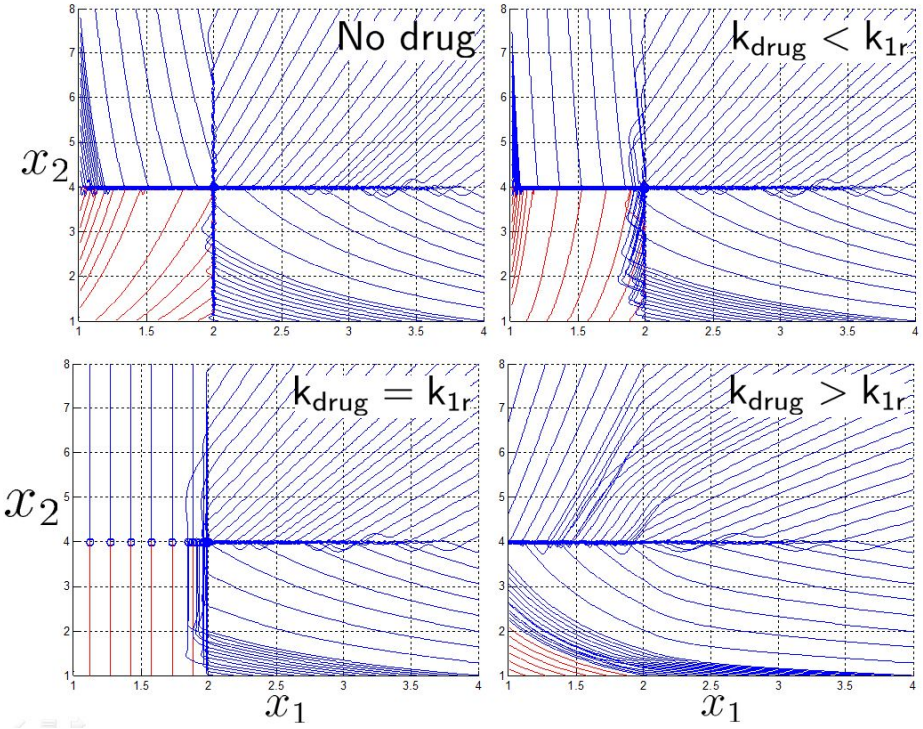


Figure 3.8: Phase analysis for a simple example with drug model.

3.5 Conclusions

In this chapter, we have shown how to combine an ODE model, representing a known network, with an unknown part represented by a BN. The BN could represent a broad range of biological systems. We proposed the concept of a "virtual state", a variable containing transition information, and we presented the parameter estimation method for our framework. Numerical examples in biology and engineering were presented. We also briefly showed that a graphical representation could help to understand biological systems and be useful in the control context. We are currently applying our framework to HER2 over-expressed breast cancer, where short term phosphoprotein mediated signaling is very well studied, yet there is a significant part of the network such as long term transcriptional feedback regulation of HER3 and influences of PI3K mutations which are currently unknown. Also, we are extending our framework to consider measurement noise.

Chapter 4

Optimization-based Inference for Temporally Evolving Networks

The problem of identifying dynamics of biological networks is of critical importance in order to understand biological systems. In this chapter, we propose a data-driven inference scheme to identify temporally evolving Boolean network representations of genetic networks. In the formulation of the optimization problem, we use an adjacency map as *a priori* information, and define a cost function which both drives the connectivity of the graph to match biological data as well as generates a sparse and robust network at corresponding time intervals. Also, we extend this algorithm for the continuous time case, and formulate the problem as a Linear Quadratic (LQ) Optimal Control problem by using a given graph structure as *a priori* information. Through simulation studies on simple examples, it is shown that this optimization scheme can help to capture the topological change of a biological signaling pathway, and, furthermore, might help to understand the structure and dynamics of biological genetic networks with a systems point of view. As an example, we apply this algorithm to study a breast cancer signaling pathway [72] to understand short-term and long-term behaviors by capturing the dynamic evolution of the network. The material in this chapter is based on the work in [36][31][32].

4.1 Introduction

Modeling of biological genetic networks has received much recent research attention. Many current data-driven inference algorithms such as Bayesian network models of genetic networks formed by coding *a priori* knowledge on the regulatory relationships into probabilistic models [127][61][166], are limited in their ability to represent temporally evolving dynamics. On the other hand, there are many studies of identification of regulatory networks using deterministic models such as ordinary differential equations (ODEs) or linear models based on least squares identification [134][12][135]. However, such assumptions about the model structure could be problematic because prejudices are automatically imposed which then

restrict the representation and understanding of biological data. For example, a key assumption of a mass action kinetics model is that there is a large number of molecules which are homogeneously mixed, an assumption which may fail inside a cell when there are only a few molecules governing the reaction. Therefore, such "theory-driven" modeling requires a good understanding of the dynamics of the signaling pathway. However, often models of biological systems are too complex to understand because of the large number of components involved and the nonlinearity of the reaction or interaction. As a result, the behavior of these systems in general cannot be completely understood from a systems point of view.

On the other hand, logical models like Boolean Networks (BNs) seek completely qualitative rather than quantitative models of biological systems. BNs can succeed in capturing high-level phenomena such as activation or deactivation with fewer parameters than their ODE counterpart, and can be used to evaluate model structure. However, they cannot capture transient response, only steady state. In spite of this, there are many applications of Boolean networks to modeling and analyzing biological systems because they are easy to simulate or evaluate, as well as an increase in research activities to address questions arising from biological applications [142][171].

Since a graph is a natural way to represent a biological network, if a system can be abstracted into a graph, it might help to understand the biological network. A graph is a set of vertices which represents states, and a set of edges which depicts the relationship or connection between two or more states. A given connectivity or adjacency map is a signed, directed graph $GR = (V, E, S)$ where V is a set of vertices, E is a set of directed edges, and $S : E \rightarrow \{-1, 0, +1\}$. For example, $e_{ij} = 1$ represents the case in which input node j activates output node i. If input node j inhibits output node i, then $e_{ij} = -1$. If input node j does not affect output node i, then $e_{ij} = 0$. Also, graphs are well-suited for situations in which there is little prior or explicit knowledge of the dynamics. Moreover, if we can build a graph model to represent biological data, we could escape imposed prejudices from the model structure. There are several graph mining approaches to biological networks [4][77]. These approaches represent biological networks as graphs, where nodes represent genes and edges represent relationships between each gene, and discover frequent patterns or motifs [4] in these graphs. They focus on structural features of networks and can effectively uncover the functional interaction structure of a biological network. Also, these approaches consider time-invariant networks and local or modular behavior of large networks. Recent studies [164][96] have proposed a concept of a temporal sequence of network motifs where the motifs change according to the dynamic nature of the biological system and can describe pivotal developmental events which cannot be captured by the static network approach; the former [164] develops algorithms for graph-rewriting rules based on machine learning techniques, which brings complexity issues with analyzing very large graphs [164]. On the other hand, the latter [96] applies a temporal sequence of network motifs analysis by reconstructing the active sub-networks (3-node sub-graphs).

In this chapter, we focus our attention on identifying time-varying linear models of sparse biological networks represented as graphs. We develop time-varying linear models, where the model remains constant for a time step or a series of time steps. This can capture dynamics

which change over time and can allow the graph at each time step to be sparse. With this approach in mind, our question becomes how to infer the graph structure from a set of data and how to find the most reasonable model among many possible configurations, since our problem formulation has fewer constraints than theory-driven modeling. We assume *a priori* information is given as a connectivity map; however this is not necessary, as the map may include all possible connections in the case of no *a priori* information. In the HER2 positive breast cancer signal pathway that we study, parts of the network are very well studied, yet there is a significant part of the network which is currently unknown. Therefore, we can include the well known signaling pathway as *a priori* information and may include all possible connections for the unknown parts.

In general, including known information helps to find a more biologically reasonable model. For example, without any *a priori* information, our algorithm can find the most reasonable model in the sense of minimizing a cost function, but if we include known information as a connectivity map, we find the most reasonable model satisfying the connectivity map. Despite uncertainties about details for a given biological system, we often have reasonable qualitative knowledge about interactions of each gene, so we can use this information as *a priori* information. In this setting, the model behavior is solely based on this qualitative information which guarantees biologically reasonable behavior: a sparse and smooth evolving network. We formulate our cost function based on those assumptions. Then, using convex optimization techniques, we find the sparsest time-varying graph consistent with experimental observations [31]. Also, the reconstructed graph shows the signal propagation through the sparse network that drives the placement of links and nodes. It might help to uncover the underlying dynamics and how the system dynamics evolves over time.

The rest of this section is organized as follows: Section 4.2 presents the proposed method related to modeling of biological networks and an optimization problem formulation with simple examples. Section 4.3 presents an example of the biological network of HER2 overexpressed breast cancer which has motivated our work. In Section 4.4, we consider a continuous time case, present a problem formulation with a graphical model and reformulate it as an LQ optimal control problem. In Section 4.5, we apply the proposed algorithm to simple example through simulation studies. Finally, conclusions are given in Section 4.6.

4.2 Method

We define a state vector $x(t) = [x_1(t), ..., x_n(t)]^T$, the components of which represent concentrations of proteins or states in a biological network and n represents the number of components or states. The evolution of state $x(t)$ can be modeled using an ordinary differential equation (ODE):

$$\dot{x}(t) = f(x(t), p) \tag{4.1}$$

where p is a parameter set. The nonlinear dynamic system (4.1) can be approximated by a linear system based on forming the Jacobian around steady states as shown below:

$$\delta\dot{x}(t) = \frac{\partial f}{\partial x}\delta x(t) + \frac{\partial f}{\partial p}\delta p = A\delta x(t) + B\delta p \tag{4.2}$$

A system in the form of (4.2) can be considered as a weighted directed graph. In this, A represents connectivity and B represents the sensitivity to parameter variation. If A_{ij} is zero, node j has no direct effect on node i. Also, if $A_{ij} > 0$, node j activates node i. Similarly, if $A_{ij} < 0$, node j inhibits node i. In [141][77], a convex optimization is constructed as follows:

$$\begin{aligned} &\min_{A,\bar{B}} ||(\dot{\tilde{X}} - \bar{B} - A\tilde{X})W||_F \\ &\text{subject to } card(A) \leq k, A_{i,j} > 0, A_{r,s} < 0 \end{aligned} \tag{4.3}$$

where $\tilde{X}(= [X_1\ X_2\ ...\ X_L])$ represents time course data set with different stimulations and/or inhibitions and each X_i represents the matrix form of n different components at M different time points $X_i = \begin{bmatrix} x^i_{1,1} & x^i_{1,2} & ... & x^i_{1,M} \\ x^i_{2,1} & x^i_{2,2} & ... & x^i_{1,M} \\ ... & ... & ... & ... \\ x^i_{n,1} & x^i_{n,2} & ... & x^i_{n,M} \end{bmatrix}$. Also, $\bar{B}(= [B_1\ B_2\ ...\ B_L])$ represents the set of sensitivities of parameter variation with $B_i = \overbrace{[b_i\ ...\ b_i]}^{M}$, and W represents a weighting matrix for specific experiments. Also, k is a given positive constant which represents maximum connectivity, all $A_{i,j} > 0$ represent activation edges (node j activates node i) and all $A_{r,s} < 0$ represent inhibition edges (node s inhibits node r). Therefore, this approach gives us the optimal static graph map consistent with various experimental data sets.

We extend this idea to a dynamic graph model. First, we define $\mathcal{X} = [X_N^T, X_{N-1}^T, ..., X_1^T]^T \in \mathbb{R}^{n\cdot N\times 1}$ where $X_k \in \mathbb{R}^{n\times 1}$ represents experimental data or known values (normalized or Booleanized biological data) at time k for $1 \leq k \leq N$, the components of which represent concentrations or activities in a biological network, n is the number of states of X_k, and N is the number of discrete time steps. We define an augmented matrix $\mathcal{G} = f(G_1, G_2, ..., G_N)$ which is a function of the dynamic graph G_k where each $G_k \in \mathbb{R}^{n\times n}$ is a connectivity map at time k for $1 \leq k \leq N$ which is based on *a priori* information, or a connectivity map denoted by GR. The augmented matrix, $\mathcal{G}$, satisfies an evolution of the state X_k as $X_k = G_k X_{k-1}$. In contrast to previous methodologies for dynamic graph analysis [164][96], we formulate a convex optimization-based inference method, where we embed the dynamics of a linear time-varying representation, and enforce sparsity and smooth evolution at corresponding time intervals.

4.2.1 Dynamic Graph (Linear Time Varying System)

The state X_k evolves along with time and constitutes the following linear time-varying system:

$$X_k = G_k X_{k-1} \tag{4.4}$$

where $G_k = g(GR, X_k|X_{k-1})$ is a function of both the connectivity map and time series data. Note that G_k describes how the edge activities evolve over time. For example, for given connectivity map GR, we allow the change of strength of connection to drive our dynamic model consistent with biological system or experimental data. At each time step, only a few of edges may evolve based on the relationship between X_k and X_{k-1}. If all the interactions between each component are properly identified, we can reconstruct the map G_k in terms of the connectivity and strength. For instance, $G_k(i, j) = 0.5$ represents that node j activates node i with strength 0.5. The strength might be related to the reaction rate and the concentration of other species, demonstrated by the Jacobian of a mass action kinetics model.

The goal of system identification of biological systems is to infer each G_k for $1 \leq k \leq N$ consistent with both a biological data set $\mathcal{X}$ and *a priori* information GR. In general, a gene regulation network (GRN) has the following characteristics [10]:

- *Directionality: regulatory control is directed from regulators to regulated genes.*
- *Sparsity: each single gene is controlled by a limited number of other genes, which is small compared to the total gene content (and also to the total number of transcription factors) of an organism.*
- *Combinatorial control: the expression of a gene may depend on the joint activity of various regulatory proteins.*

Since GRNs have a sparse structure with combinatorial control, we should reconstruct the sparsest graph consistent with experimental observations. We can construct an optimization problem as follows:

$$\min_{G_k} ||X_k - G_k X_{k-1}|| + \gamma ||A_k|| \quad \text{subject to } G_k = g(GR, X_k|X_{k-1}) \tag{4.5}$$

where the second term in the cost function penalizes the cost of adding edges in order to avoid heavy combinatoric computation, A_k is defined as follows:

$$A_k = G_k \cap (G_k \cap G_{k-1})^c = G_k - G_{k-1} \tag{4.6}$$

and γ is a positive constant. Therefore, A_k enforces the network to be sparse and thus the cost function represents a trade-off between reconstruction error and sparsity. Here we define the function g as shown below:

$$G_k = g(GR, X_k|X_{k-1}) = GR \oplus MAP = \mathsf{Proj}_{MAP} GR \tag{4.7}$$

where $\oplus$ is defined as a projection operator onto $MAP \in \mathbb{R}^{n\times n}$ whose i-th column is a column vector, the components of which are all one if $X_{k-1}(i)$ is active, which means the state of the i-th element is over the threshold. On the other hand, if $X_{k-1}(i)$ is non-active, then the i-th column of MAP is a zero column vector. Therefore, this projection gives us all possible candidate edges based on both X_{k-1} and GR. For example, if x_i at the $(k-1)$th step is active, then the i-th column of GR contains the candidate edges. On the other hand, if x_i at $(k-1)$th step is not active, we cannot use the i-th column of GR as candidate edges. By using (4.7), our method generates a sparse network representation without any *Lasso*-type regressions in (4.5).

If we implement the optimization problem for every single step as shown in equation (4.5), the penalty term for sparsity does not play the role of generating a sparse network, but instead uses all possible edges. In other words, distributing signal to all possible nodes (dense network) gives us a lower cost than distributing signal to only a few nodes (sparse network) in our formulation (4.5). We can think about this by considering *dynamic programming* in optimization. The main idea behind dynamic programming is that, to solve a given problem, we need to solve different parts of the problem (subproblems), then combine the solutions of the subproblem to reach an overall solution in a recursive manner. Similarly, in order to find the sparsest smoothly evolving graph, we need to have a certain connection between every subproblem. For example, when we consider the overall time horizon in problem (4.8), the penalty term for sparsity can play a key role in generating a sparse graph structure by connecting the discrete time dynamics at each time step with those at different time steps. Then, we can construct a convex optimization problem for the proposed identification problem as shown below:

$$\begin{aligned} \min_{G_1,\ldots,G_N} \quad & \sum_{k=1}^{N} ||X_k - G_k X_{k-1}||_2 + \gamma\{\sum_{k=2}^{N} ||G_k - G_{k-1}||_F + ||G_1||_F + ||G_N||_F\} \\ & \text{subject to } G_k = g(GR, X_k|X_{k-1}) \end{aligned} \tag{4.8}$$

Note that the first term of problem (4.8), the summation of $||X_k - G_k X_{k-1}||$, forces the minimization of the reconstruction error for a given dynamical network at time k for $1 \leq k \leq N$. Also, the second term, the summation of $||G_k - G_{k-1}||_F$ plays the role of realizing a smooth evolution and minimizes the change in network evolution. Finally, with the penalty term $||G_1||_F + ||G_N||_F$ which acts as a boundary constraint, we can find the sparsest dynamic graph. We can also arrange and reformulate equation (4.8) as follows:

$$\begin{aligned} \min_{\mathcal{G}} \quad & ||\mathcal{X} - \mathcal{G}\mathcal{X}||_2 + \gamma||(\mathcal{G}^T - \mathcal{G}) \times W||_F \\ & \text{subject to given } \mathcal{X}, W \\ & G_k^{act} \quad \geq 0, G_k^{inhib} \leq 0, G_k^{others} = 0 \end{aligned} \tag{4.9}$$

where $\mathcal{G} = \begin{bmatrix} O_n & G_N & O_n & ... & O_n \\ O_n & O_n & G_{N-1} & ... & O_n \\ ... & ... & ... & ... & ... \\ O_n & O_n & ... & O_n & G_1 \\ O_n & O_n & ... & O_n & I_n \end{bmatrix}$, $\mathcal{X} = \begin{bmatrix} X_N \\ X_{N-1} \\ ... \\ X_1 \\ X_0 \end{bmatrix}$ and $W = \begin{bmatrix} I_n & O_n \\ O_n & I_n \\ I_n & O_n \\ O_n & I_n \\ ... & ... \\ ... & ... \end{bmatrix}$ where $\mathcal{X} \in \mathbb{R}^{n\cdot(N+1)\times 1}$, $\mathcal{G} \in \mathbb{R}^{n\cdot(N+1)\times n\cdot(N+1)}$ and $W \in \mathbb{R}^{n\cdot(N+1)\times 2\cdot n}$ for $1 \leq k \leq N$. Note that the first term of the cost function in equation (4.9) is a reconstruction error cost and the second term plays the role of connecting each discrete system with another and realizing a smooth evolution of the network by selecting effective edges with inequality constraints.

4.2.2 Static Graph (Linear Time Invariant System)

If we assume that the graph model does not evolve with time (static graph $G_k = G$) such as with a linear time invariant system [77], we can modify the structure of $\mathcal{G}$ and constraints as shown below for a fixed pattern graph:

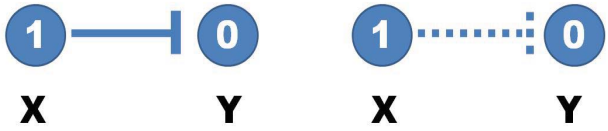
$$\mathcal{G} = \begin{bmatrix} O_n & G & O_n & ... & O_n \\ O_n & O_n & G & ... & O_n \\ ... & ... & ... & ... & ... \\ O_n & O_n & O_n & O_n & G \\ O_n & O_n & O_n & O_n & I_n \end{bmatrix}$$
$$G^{act} \geq 0, G^{inhib} \leq 0, G^{others} = 0 \tag{4.10}$$

where $G = \bar{g}(GR)$ does not depend on time (compared with $G_k = g(GR, X_k|X_{k-1})$ for a linear time-varying system). Note that for a fixed graph structure, the optimal solution represents the average connectivity map [77].

4.2.3 Dynamic Graph vs. Static Graph

We can compare the dynamic graph and static graph method: the main difference in cost function from dynamic and static graph is the penalty for sparsity as follows:

$$||(\mathcal{G}^T - \mathcal{G})W)||_{F,dynamic} = \begin{bmatrix} O_n & -G_N \\ \Delta G_{N-1} & O_n \\ O_n & \Delta G_{N-2} \\ ... & ... \\ O_n & \Delta G_1 \\ G_1 & O_n \end{bmatrix} \tag{4.11}$$



(a) (b)

Figure 4.1: Possible cases for inhibition edge (dash end) where X and Y represent different genes or states; 1 represents activated state and 0 represents deactivated state: (a) inhibition reaction is triggered and (b) inhibition reaction is not occurred.

$$||(\mathcal{G}^T - \mathcal{G})W)||_{F,static} = \begin{bmatrix} O_n & -G \\ O_n & O_n \\ O_n & O_n \\ ... & ... \\ O_n & O_n \\ G & O_n \end{bmatrix} \tag{4.12}$$

where $\Delta G_k = G_{k+1} - G_k$. Also, if we modify the constraint for a dynamic graph similar to the static graph approach, the dynamic graph approach gives us a lower cost than the static graph approach because the structural constraint restricts the degrees of freedom in choosing edges:

$$\min_{\mathcal{G}} ||\mathcal{X} - \mathcal{G}\mathcal{X}||_2 + \gamma||(\mathcal{G}^T - \mathcal{G}) \times W||_F \leq \min_{\bar{\mathcal{G}}} ||\mathcal{X} - \bar{\mathcal{G}}\mathcal{X}||_2 + \gamma||(\bar{\mathcal{G}}^T - \bar{\mathcal{G}}) \times W||_F \tag{4.13}$$

where $\mathcal{G}$ represents the optimal solution of the dynamic graph approach and $\bar{\mathcal{G}}$ represents the optimal solution of the static graph approach.

4.2.4 Inhibition Edges

Based on our formulation of the optimization problem, we can find the optimal solution which satisfies a trade-off between representation of data (dynamics), sparsity and smooth evolution. However, the optimal solution does not include any inhibition edges ($\dashv$) because it is not necessary according to our optimization problem as shown in Figure 4.1. For example, if X is active (1) and Y is not active (0), then there might be two possible cases: X inhibits Y (X $\dashv$ Y connected, Figure 4.1 (a)) or no connection between X and Y (Figure 4.1 (b)). Having no connection would give the lower cost. However, we can handle inhibition edges using Boolean logic as an algebraic constraint as shown below:

$$\mathsf{Y} = \mathsf{not}\ \bar{\mathsf{Y}}\ (=\sim \bar{\mathsf{Y}}) \tag{4.14}$$

Also, we extend this algebraic constraint to a normalized state as shown below:

$$\mathsf{Y} + \bar{\mathsf{Y}} = 1 \tag{4.15}$$

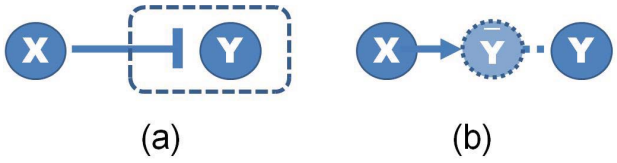


Figure 4.2: (a) Inhibition edge (X inhibits Y) (b) modified edge (X activates $\bar{\mathsf{Y}}$, and the relation between $\bar{\mathsf{Y}}$ and Y is defined by Boolean logic or algebraic constraint)

Consider the simple case shown in Figure 4.2, in which state X inhibits state Y. Using an algebraic constraint (4.14), we can change the inhibition edge to an activation edge with the new state $\bar{\mathsf{Y}}$ as shown below:

$$\mathsf{X} \dashv \mathsf{Y} \Leftrightarrow \mathsf{X} \dashv (\sim \bar{\mathsf{Y}}) \Leftrightarrow \mathsf{X} \rightarrow \bar{\mathsf{Y}} - \mathsf{Y} \tag{4.16}$$

Hence, we extend states if there are inhibition edges and introduce a diagonal weighting matrix $\mathcal{M}$, which makes all species have the same penalty as shown below:

$$\min_{\mathcal{G}} ||\mathcal{M} \times (\bar{\mathcal{X}} - \mathcal{G}\bar{\mathcal{X}})||_2 + \gamma||(\mathcal{G}^T - \mathcal{G}) \times W||_F \tag{4.17}$$

where $\bar{\mathcal{X}}$ represents extended states and $\mathcal{M}_{ii} = \{1, \frac{1}{\sqrt{2}}\}$. If there exist x and $\bar{x}$ for a specific state, $\mathcal{M}_{ii} = \frac{1}{\sqrt{2}}$, and otherwise, $\mathcal{M}_{ii} = 1$.

4.2.5 Numerical Examples

In this section, we consider simple examples to illustrate the proposed inference scheme.

(a) Simple Gene Regulatory Network

We first consider a simple example composed of four genes. The *a priori* information and the snapshot of gene expression are shown in Figure 4.3. Here we do not consider state extension for inhibition edges which means the optimal solution does not include any inhibition edges. By varying the parameter γ, we can sweep out the optimal trade-off curve between the reconstruction error and the sparsity of a solution as shown in Figure 4.4. We can choose the optimal parameter γ^* by the graphical representation: the extreme point γ^* on the trade-off between the sparsity and the reconstruction error. Once we fix the parameter γ^*, we solve the constrained convex optimization problem (4.28) using CVX [19]. Figure 4.5 shows the dynamics of the connectivity graph. We can capture the temporal graph not only in terms of connection but also strength of the edge. From the optimal graph representations, we could extract how the signaling pathway evolves over time with a systems point of view. Also we can compare the two approaches: dynamic and static graph approach with average of dynamic graph.

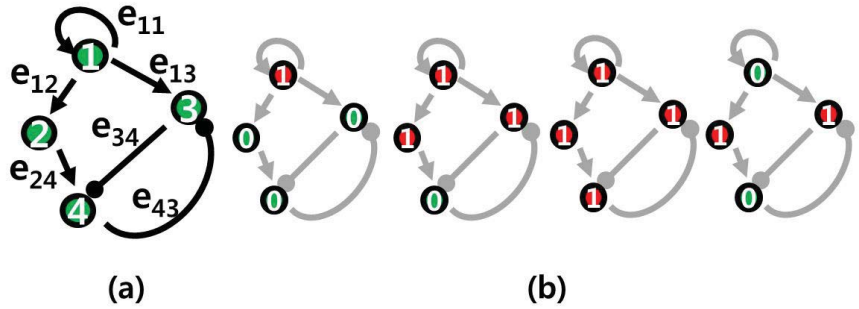


Figure 4.3: (a) *a priori* connectivity map where the arrows indicate activation and blunted lines denote inhibition. (b) snapshots of gene expression from time step k=1 to k=4 (red or 1: activated states, green or 0: deactivated states)

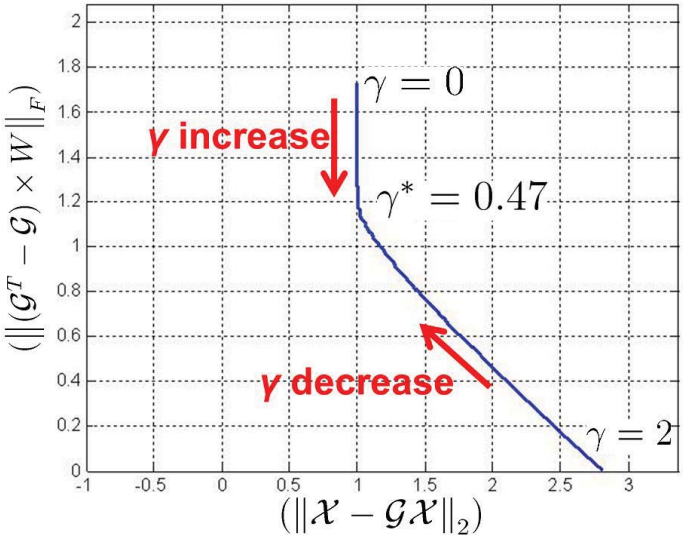


Figure 4.4: Trade-off curve between the model fitting ($||\mathcal{X} - \mathcal{G}\mathcal{X}||_2$) and the sparsity ($|||(\mathcal{G}^T - \mathcal{G}) \times W||_F$) with tuning parameter γ.

(b) Simple Gene Regulatory Network with different structure

Here, we add 1 edge which connects node 3 to node 2 as shown in Figure 4.6 and solve the optimization problem again. We can see the difference of the strength of edge 12 (e_{12}) compared with above example. Basically, for the previous example, the optimal graph shows the robust pathway distributing power evenly (e_{12} and e_{13} in Figure 4.5) because both pathways are effective equally. However, for this example, an additional pathway (e_{32}) changes the topology of the graph which makes the optimal graph choose the more effective or economical path ($e_{13} - e_{32}$) shown in Figure 4.7. In other words, the optimal solution shows that the strength of e_{12} decreases because there exists a more effective pathway ($e_{13} - e_{32}$).

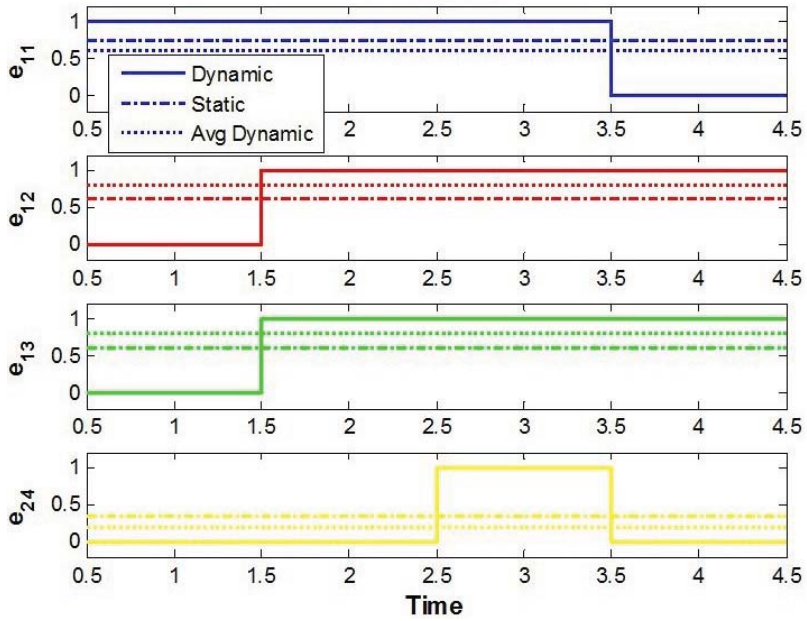


Figure 4.5: The optimal solution for example (a): the magnitude of each edge represents the strength of connection. Also, we compare the result from the dynamic and static graph approaches with the average of the dynamic graph.

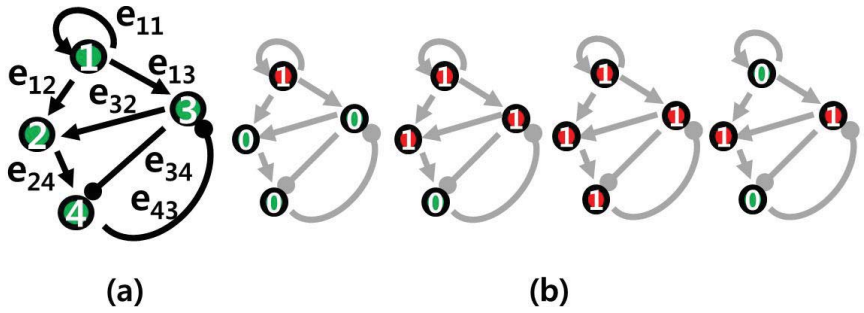


Figure 4.6: *a priori* connectivity map with an additional edge (e_{32}) which connects from node 3 to node 2.

4.3 Biological Signal Pathway Examples

4.3.1 p53 Signal Pathway

Aswani *et al.* [8] proposed a graph-theoretic topological control applied to the p53 signaling pathway. We apply our approach to understand how the controller affects the biological pathway and capture the evolution of signaling pathway. We define $X = [\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3, \mathsf{x}_4]^\mathsf{T} =$

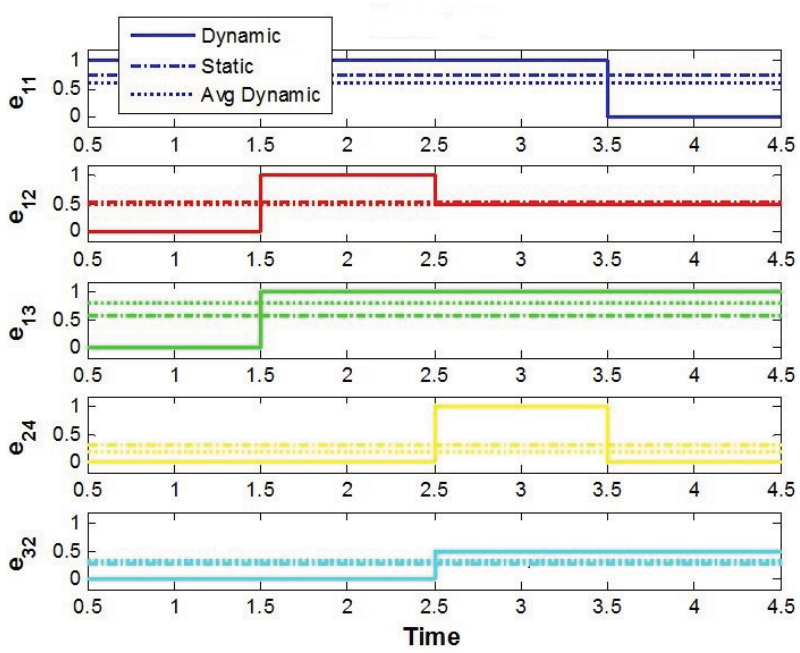
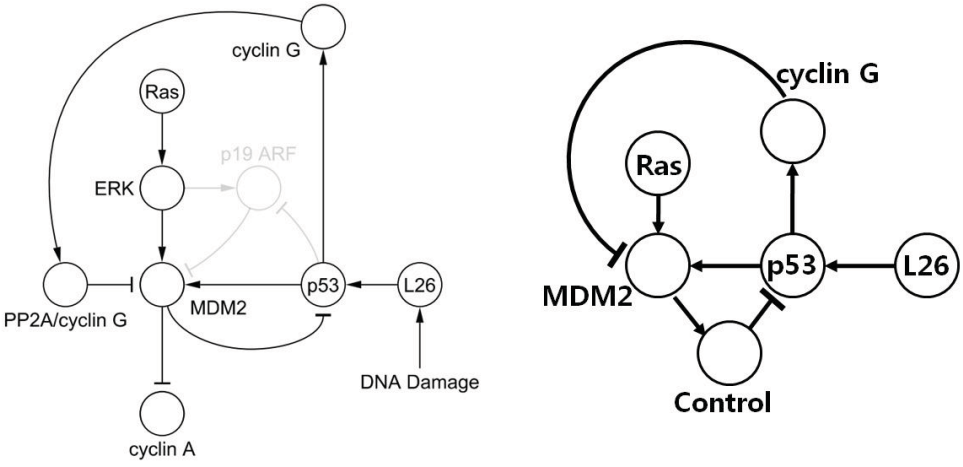


Figure 4.7: An optimal solution for example (b): an additional pathway(e_{32}) changes the topology of graph which makes the optimal graph to choose more effective path ($e_{13} - e_{32}$) rather than e_{12} (for example (a), the optimal graph shows distributing power evenly, $e_{12} = e_{13}$)



(a) (b)

Figure 4.8: (a) an abnormal p53 pathway in Figure 3(c) [8] (b) the abstract model which includes the effect of controller.

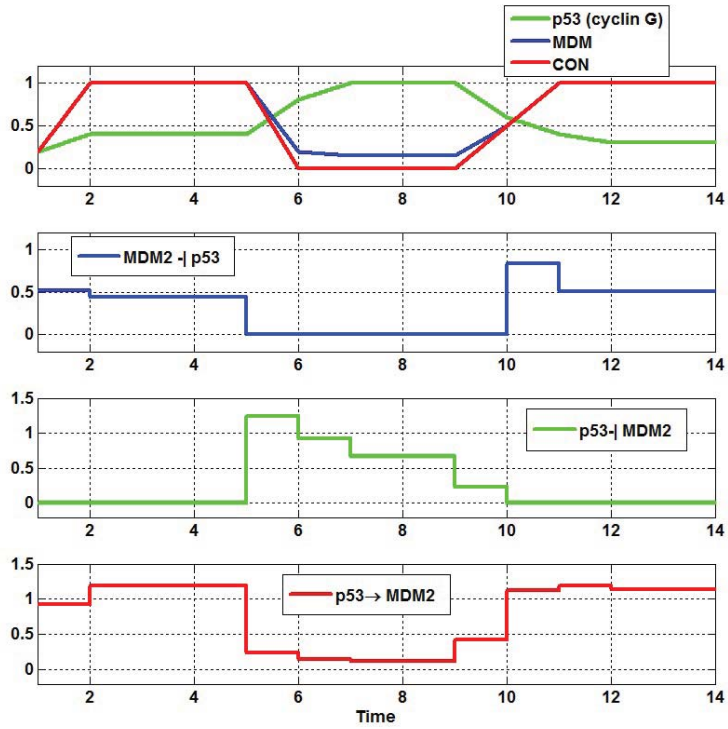


Figure 4.9: (top) normalized time course plots for the abnormal p53 pathway with controller in Figure 4 (c) [8] (lower) dynamic evolution of each edge of abnormal p53 pathway with the controller: the p53 regulates MDM2 similar to the normal p53 pathway by increasing strength of inhibition edge ([p53]-[cyclinG]-[MDM2]).

$[x_{MDM}, x_{p53}, x_{cyclinG}, x_c]$ where x_c is an imaginary state which represents the proposed control scheme (actually removing the edge in [8]):

$$\begin{aligned} x_c &= x_{MDM} \text{ if controller Off} \\ x_c &= 0 \quad\quad \text{if controller On} \end{aligned} \tag{4.18}$$

Hence, by introducing this imaginary state, we have an abstract model of abnormal p53 signaling pathway with controller in Figure 4.8(b). Also, we can define GR as follows based on Figure 4.8(b) including the state extension due to incorporating the inhibition edges:

$$GR = \overbrace{\begin{bmatrix} 0 & 0 & e_{21} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e_{3\bar{1}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e_{4\bar{2}} \\ 0 & 0 & e_{23} & 0 & 0 & 0 \\ e_{14} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}^{x_1 \quad \bar{x}_1 \quad x_2 \quad \bar{x}_2 \quad x_3 \quad x_4} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.19}$$

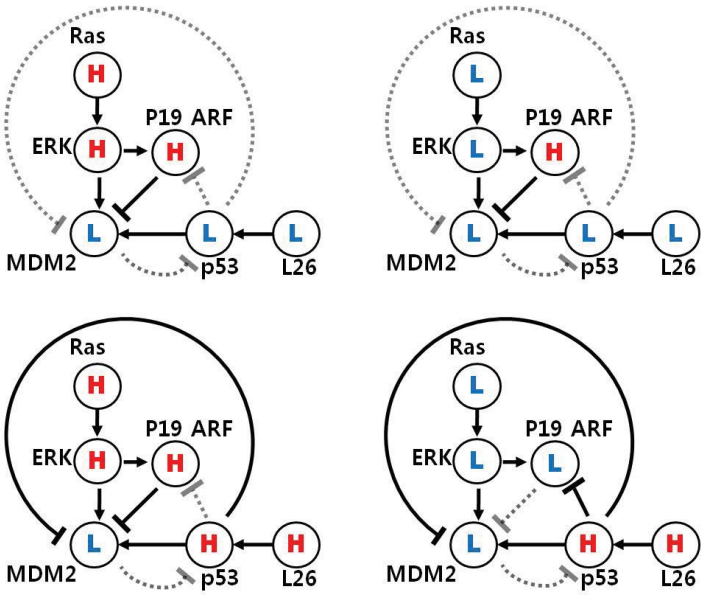


Figure 4.10: Possible cases for a normal p53 signaling pathway with different combinations of both [Ras] and [L26] where H represents an active state and L represents a non-active state [8]: the p19 ARF mainly regulates MDM2 and it cannot affect MDM2 from p53 through p19 ARF.

Here, we normalize the data and then apply our algorithm. We can capture the dynamic evolution of the graph in Figure 4.9. The controller causes the p53 concentrations to increase to higher levels by regulation edge from MDM2 to p53 and causes increased strength of inhibition edge ([p53]-[cyclin G]-[MDM2]). In other words, p53 regulates MDM2 similar to the normal p53 pathway [8]. If the controller is not applied again, the strength of edge ([p53]-[cyclin G]-[MDM2]) decreases and the strength of activation edge [p53]-[MDM2] increases. This causes MDM concentrations to increase to higher levels which cause regulation p53 by inhibition edge ([MDM2]-[Controller]-[p53]) similar to the abnormal p53 pathway [8].

We can also apply our algorithm for the normal p53 pathway in order to compare with the abnormal p53 pathway with controller. In a normal p53 pathway, we can consider all possible combinations of both Ras and L26 as two input signals. Here, the basic assumption is that the inhibition reaction is stronger than the activation reaction. Then, we find that the p19 ARF mainly regulates MDM2 and it cannot affect MDM2 from p53 through p19 ARF as shown in Figure 4.10. Hence, we can use the same abstract model in Figure 4.8(b) for a normal p53 signaling pathway. The optimal solution gives us that the normal p53 cell uses mainly inhibition edges from p53 to MDM2 through cyclin G which means p53 regulates MDM2, as shown in Figure 4.11. Therefore, the controller drives the abnormal p53 cell to the normal p53 cell by removing the inhibition edge from MDM2 to p53 as Aswani *et al.* [8] proposed.

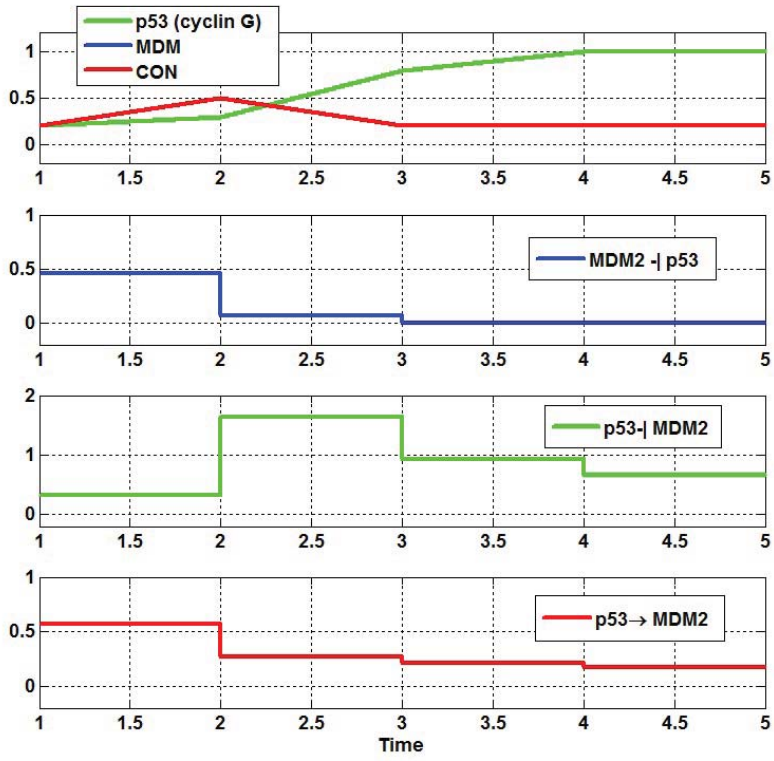


Figure 4.11: (top) normalized time course plots for the normal p53 pathway shown in Figure 4(a)[8] (lower) the dynamic evolution of each edge of the normal p53 pathway: the edge activity shows that the normal p53 cell uses mainly inhibition edges form p53 to MDM2 through cyclin G.

4.3.2 HER2 Overexpressed Breast Cancer

The experimental studies were done for investigating the effects of Tyrosine Kinase inhibitors (TKIs) on the BT474 and SKBR3 cell lines [137]. In this work, short term effects and long term effects of applying Gefitinib (a TKI) to those cell lines were studied and important effects of how the cancer cells overcome or escape from the inhibitory effects of TKIs were discovered. The authors in [137] showed that HER3 is recruited from the cytoplasm to the cell membrane by vesicular trafficking to increase the triggering signal in order to escape from HER2 inhibition. Also, they tested the effects of vesicular trafficking: when vesicular trafficking was stopped, phospho-HER3 and phospho-Akt did not survive the inhibition of HER2.

We suspect there might be short-term and long-term topological changes because the TKI can inhibit and regulate downstream effectively in the short-term but it cannot regulate for the long-term. Therefore, we hypothesize that during the short-term, there might be Positive

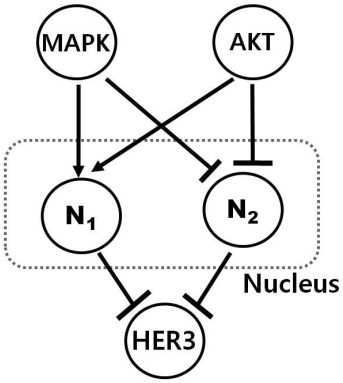


Figure 4.12: Bifan motif of nucleus, which is two-layered graphs with edges from nodes in top- to bottom-layer: there is a fail-safe mechanism, HER2-HER3 signaling which is buffered is protected against an inhibition of HER2 catalytic activity, and a compensatory mechanism by cross-talk between MAPK and Akt which results in robust activation of buffering.

Negative (PN) Feedback [95] so the TKI inhibits HER3 effectively. However, for long-term behavior, even a small triggering signal could amplify the phospho-Akt signal, because of Positive Positive (PP) Feedback which is similar to vesicular trafficking. On the other hand, if the topology does not change, TKI should be able to regulate downstream over the long term even though HER3 is recruited by vesicular trafficking.

We define the *a priori* map from biological information [137][5][87] where we include a nucleus model to capture this possible topology change. The behaviors of the nucleus are not yet understood, however we abstract it with the switch as shown in Figure 4.12. Basically, there is a fail-safe mechanism, HER2-HER3 signaling which is buffered so that it is protected against an inhibition of HER2 catalytic activity and it is driven by the negative regulation of HER3 by Akt [5]. Also, there is a compensatory mechanism by cross-talk between MAPK and Akt which results in robust activation of this buffering. However, the compensatory buffering prevents apoptotic tumor cell death from occurring as a result of the combined loss of MAPK and Akt signaling [5]. For example, once a signal is triggered and either MAPK or Akt is high, then the nucleus stays active so MAPK and/or Akt are trying to keep the compensatory buffering. However, once both MAPK and Akt are down regulated, the nucleus is deactivated for all time.

We apply the proposed optimization technique and the result is shown in Figure 4.13 and 4.14. Here, we use the generated data based on biological experimental data (western blot [137][5]). By applying the proposed method, we find that there are three main phases: before TKI is introduced (triggering network), right after TKI is introduced (short-term) and long term behavior after TKI is introduced. We can capture the topology change of the biological network: for the initial stage (Figure 4.14 (a)), the signal is triggered and propagated along activation edges. After TKI is introduced, downstream components such as phospho-HER3, PI3K, Akt and MAPK are regulated because TKI inhibits and regulates

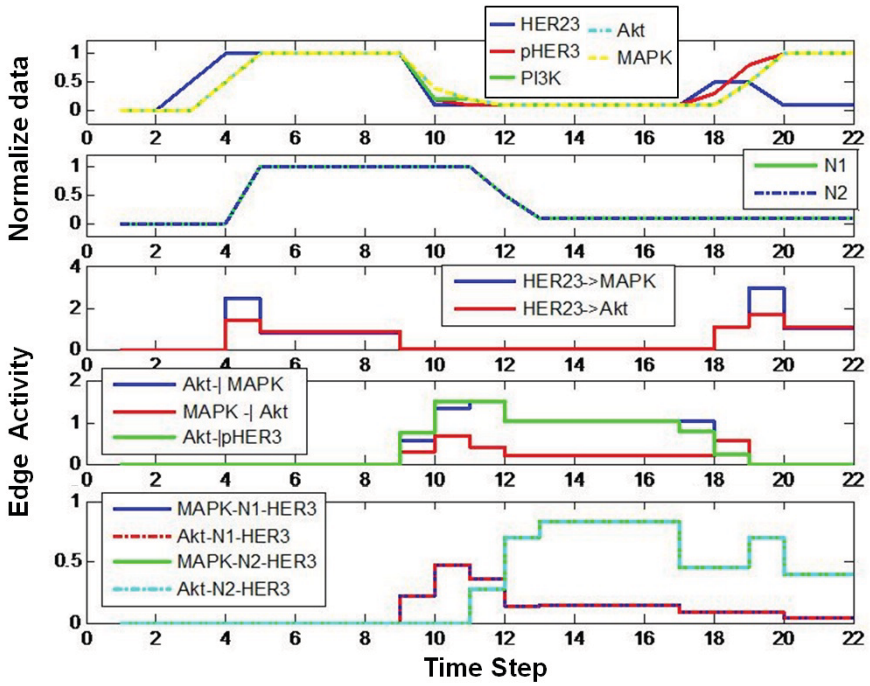


Figure 4.13: The upper two figures show the normalized biological data and the assumed nucleus level. The other (lower) figures show the strength of the downstream edges. For example, the edge connecting HER23 to MAPK (middle figure) is activated from step 4 to step 9 but deactivated from step 9 to 18.

downstream components. Moreover, the biological network shows PN Feedback which effectively modulates signal responses. Finally, for long term behavior, even if a small triggering signal is introduced (because of TKI inhibition, step 17-step 20), the downstream components are not regulated but are activated because the biological network evolves to Positive Positive (PP) Feedback which induces a slower but amplified signal response and enhances bi-stability.

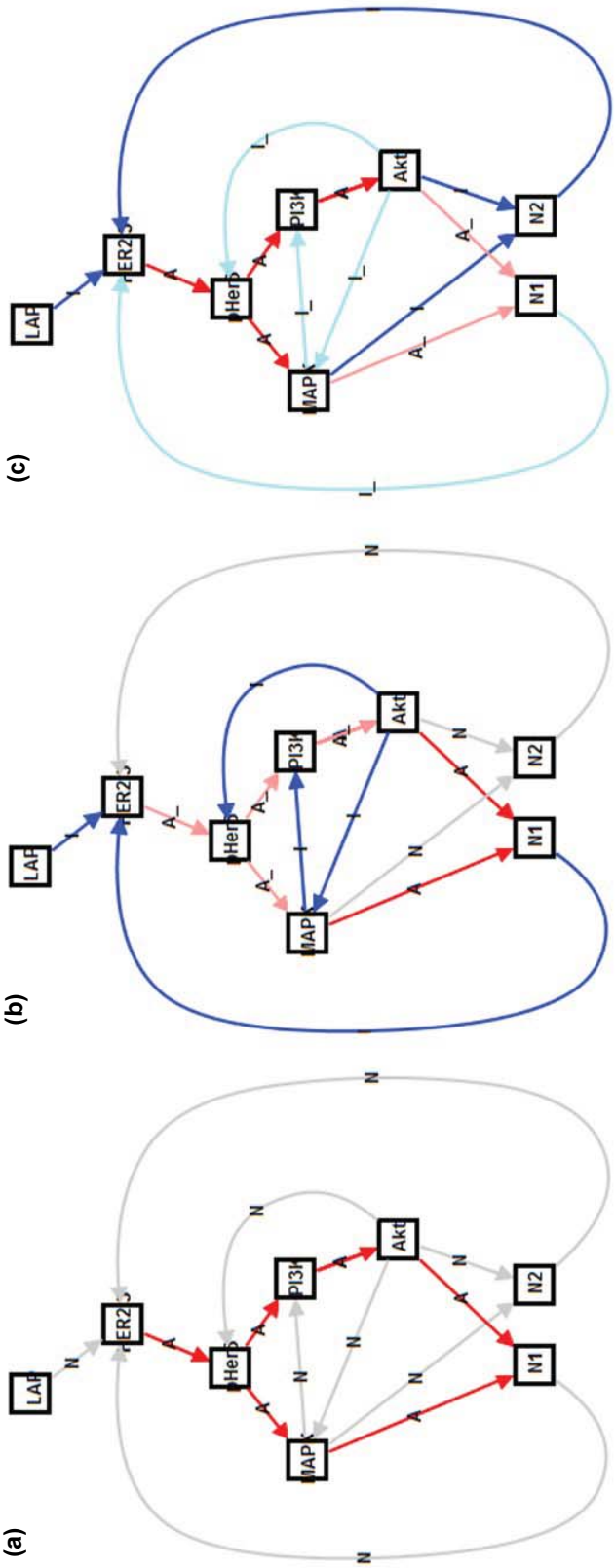


Figure 4.14: (a) Signal is triggered and propagated along activation edges (b) After TKI is introduced (short term), downstream components such as phospo-HER3, PI3K, Akt and MAPK are regulated because TKI inhibits and regulates downstream components (Positive Negative Feedback) (c) For long term behavior, even though a small triggering signal is introduced, the downstream components are not regulated but are activated because the biological network evolves to Positive Positive Feedback (gray: not triggered edge, red : activation edge, blue : inhibition edge, light red/blue: deactivated edges after once activated)

4.4 Continuous Time Case

As we discussed, we infer discrete time dynamics from a set of data with *a priori* graph structure information and focus on how to find the most reasonable model among many possible configurations [36]. We can extend this idea for the continuous time case. The identification problem has led to Linear Quadratic (LQ) Optimal control problem with two main penalty functions by which we can match the experimental data with a sparse representation using *a priori* information of structure.

As we mentioned earlier, since a graph model is a natural way to represent a biological signal pathway, it does not require any constraints on dynamics such as mass action kinetics or Hill function representations, used in Ordinary Differential Equation (ODE) models. Also, many different measurement techniques are developed which allow us continuous data acquisition. Therefore, the inference scheme for the continuous case can be useful to build models with fine-sampled data set and identify general systems using graphical representation.

4.4.1 Problem Statement

Recall biological network model (4.1) where the components of which represent concentration of proteins and p is a parameter set. As we mentioned earlier, many studies in systems biology impose a structure on $f(\cdot)$, such as mass action kinetics or Hill function dynamics, and identify parameters using least-square criteria. However, in the previous section, we presented a precursor to this algorithm for the discrete time-varying influence map which can be formulated as a discrete time-varying linear system. Here, we basically extend this idea for the continuous case. The nonlinear dynamic system (4.1) can be approximated by a time-varying linear system based on forming the Jacobian around steady states as shown below:

$$\begin{bmatrix} \delta\dot{x}_1(t) \\ \delta\dot{x}_2(t) \\ \dots \\ \delta\dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \dots \\ \delta x_n \end{bmatrix} = G(t)\delta x(t) \tag{4.20}$$

where we assume there is no parameter variation ($\delta p = 0$). A system in the form of (4.20) can be considered as a temporally evolving weighted directed graph. Then, $G(t)$ is a time-varying adjacency matrix, or influence map, of dimension $n \times n$ which describes the temporal evolution of the edges with strength change. In general, $G(t)$ is a sparse matrix [10]:

$$G_{i,j}(t) = \frac{\partial f_i}{\partial x_j} = \begin{cases} \neq 0 & \text{if node } j \text{ can affect node } i \text{ directly} \\ = 0 & \text{otherwise} \end{cases} \tag{4.21}$$

where $G_{i,j}(t)$ is non-zero if there exists a direct connection between node j (input node) and node i (output node). Otherwise, $G_{i,j}(t)$ is zero.

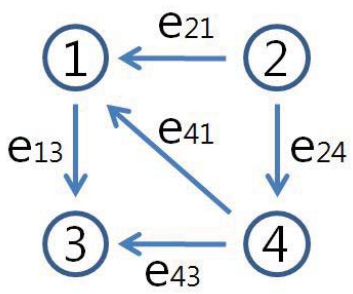


Figure 4.15: A simple graph model.

Definition. (*Component*) Let $G(t)$ be a time-varying adjacency matrix which represents a dynamic graph with n nodes and k edges where k is the number of candidate edges from *a priori* information. The *component* of $G(t)$, denoted $e(t) = comp(G(t))$, is a $k \times 1$ vector whose elements are the nonzero entries of $G(t)$.

Example. Consider the dynamic graph shown in Figure 4.15. Following the conventions introduced above, the corresponding adjacency matrix $G(t)$ has the form:

$$G(t) = \begin{bmatrix} 0 & e_{21} & 0 & e_{41} \\ 0 & 0 & 0 & 0 \\ e_{13} & 0 & 0 & e_{43} \\ 0 & e_{24} & 0 & 0 \end{bmatrix} \tag{4.22}$$

Its component $e(t)$ is constructed by extracting the nonzero elements from each column, which produces the vector:

$$e(t) = [e_{13}, e_{21}, e_{24}, e_{41}, e_{43}]^T \triangleq [e_1, e_2, e_3, e_4, e_5]^T \tag{4.23}$$

Using $e(t)$, we can reformulate system (4.20) as follows (for example, $n = 4, k = 5$):

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}_{n\times 1} &= \begin{bmatrix} 0 & e_{21} & 0 & e_{41} \\ 0 & 0 & 0 & 0 \\ e_{13} & 0 & 0 & e_{43} \\ 0 & e_{24} & 0 & 0 \end{bmatrix}_{n\times n} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}_{n\times 1} = \begin{bmatrix} 0 & x_2 & 0 & x_4 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ x_1 & 0 & 0 & 0 & x_4 \\ 0 & 0 & x_2 & 0 & 0 \end{bmatrix}_{n\times k} \begin{bmatrix} e_{13}(t) \\ e_{21}(t) \\ e_{24}(t) \\ e_{41}(t) \\ e_{43}(t) \end{bmatrix}_{k\times 1} \\ &= A(x)e(t) \\ &= e_1(t) \begin{bmatrix} 0 \\ 0 \\ x_1 \\ 0 \end{bmatrix} + e_2(t) \begin{bmatrix} x_2 \\ 0 \\ 0 \\ 0 \end{bmatrix} + e_3(t) \begin{bmatrix} 0 \\ 0 \\ 0 \\ x_2 \end{bmatrix} + e_4(t) \begin{bmatrix} x_4 \\ 0 \\ 0 \\ 0 \end{bmatrix} + e_5(t) \begin{bmatrix} 0 \\ 0 \\ x_4 \\ 0 \end{bmatrix} \end{aligned} \tag{4.24}$$

where $A(x) \in \mathbb{R}^{n\times k}$ is a linear function of x, which can be constructed from *a priori* information, representing possible influence modes of biological networks. For example, the first

mode, $[0\ 0\ x_1\ 0]^T$ in equation (4.24) shows that node 1 activates node 3 (i.e, $x_1 \rightarrow \dot{x}_3 \rightarrow x_3$). Also, each $e_i(t)$ represents a time-varying coefficient or an activity of i-th mode $\in \mathbb{R}^{n\times 1}$. Therefore, we can assign the network topology by adding edges, for example, if there are suspicious interactions among nodes. Here, we are using a similar notion to modal analysis in mechanical vibration systems but the main difference is that our mode is constructed by a graph model. For example, each $e_i(t)$ is similar to a generalized coordinate in modal analysis and each column vector (influence mode) represents the eigenvector in modal analysis.

In order to find all $e_i(t)$ which drive our system dynamics with influence modes, we formulate a Linear Quadratic (LQ) optimal control problem, as a regulation problem ($x(t) \rightarrow x_d(t)$) with control inputs both $e(t)$ and $\dot{e}(t)$. Once we solve the LQ problem, $e_i(t)^*$ shows the optimal activity or sequence of each mode over time which drives our dynamic system to match biological data.

4.4.2 Time-Varying Linear System

In order to formulate the LQ optimal control problem, we define the controlled system as follows:

$$\frac{dx(t)}{dt} = A(x)e(t) \tag{4.25}$$

and the optimal control is sought to minimize the quadratic performance index as follows:

$$\begin{aligned} J &= \frac{1}{2}(x(t_f) - x_d(t_f))^T S_1 (x(t_f) - x_d(t_f)) \\ &\quad + \frac{1}{2}\int_0^{t_f} \{[x(t) - x_d(t)]^T Q_1 [x(t) - x_d(t)] + \dot{e}(t)^T R \dot{e}(t) + e(t)^T Q_2 e(t)\} dt \end{aligned} \tag{4.26}$$

where S_1, Q_1 and Q_2 are positive semidefinite matrices and R is a positive definite matrix.

The LQ problem as formulated above is concerned with tracking of the desired trajectory ($x_d(t)$, biological data). In the performance index J, the first term penalizes the deviation of $x(t_f)$ from the desired trajectory at the final time. Inside the integral, the first term penalizes the transient deviation of $x(t)$ from the desired trajectory $x_d(t)$ which represents the error dynamics. The second penalizes the change of activity of edges (dynamic graph) which attempts to minimize the variation of activity of edges over time (smoothly evolving). Also, the third term penalizes the activities of edges. Therefore, the second and third term attempt to achieve a sparse and smoothly evolving biological network.

In order to use a general LQ framework, first, we define $\dot{e}(t) = v(t)$ and $\bar{x}(t) = x(t) - x_d(t)$. Here, we assume that we know $x_d(t)$ and $\dot{x}_d(t)$ for $0 \le t \le t_f$ because once we have $x_d(t)$ then we can get $\dot{x}_d(t)$ by using the derivative of a polynomial fitting. We define an $(n+k) \times 1$-dimensional state $X(t) = [\bar{x}(t)^T, e(t)^T]^T$. Then, the state equation for the enlarged state vector can be formulated as follows:

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix}\bar{x}(t)\\ e(t)\end{bmatrix} &= \begin{bmatrix}0_{n\times n} & A(x)_{n\times k}\\ 0_{k\times n} & 0_{k\times k}\end{bmatrix}\begin{bmatrix}\bar{x}(t)\\ e(t)\end{bmatrix} + \begin{bmatrix}-\dot{x}_d(t)\\ v(t)\end{bmatrix} = \mathcal{A}(x)X(t) + \begin{bmatrix}-\dot{x}_d(t)\\ 0_{k\times 1}\end{bmatrix} + \begin{bmatrix}0_{n\times 1}\\ v(t)\end{bmatrix} \\ &= \mathcal{A}(x)X(t) + W(t) + V(t) \end{aligned} \tag{4.27}$$

where $\mathcal{A}(x)$ is also a linear function of x. Note that the augmented system is still a linear system because there is no multiplication between $A(x)$ and $\bar{x}(t)$. Also, the performance index (4.26) can be written as follows:

$$\begin{aligned} J &= \frac{1}{2}X(t_f)^T \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix} X(t_f) + \frac{1}{2}\int_0^{t_f} \{X(t)^T \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} X(t) + V(t)^T \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix} V(t)\} dt \\ &= \frac{1}{2}X(t_f)^T \mathcal{S} X(t_f) + \frac{1}{2}\int_0^{t_f} \{X(t)^T \mathcal{Q} X(t) + V(t)^T \mathcal{R} V(t)\} dt \end{aligned} \tag{4.28}$$

The problem is now reformulated as a standard LQ problem with the exception of $\mathcal{R}$ which is a singular matrix. However, we are interested in $v(t)$ so the solution of the continuous time LQ problem is given by the state feedback control law as shown below:

$$V(t) = -\mathcal{R}^+ P(t) X(t) = -\begin{bmatrix} 0 & 0 \\ 0 & R^{-1} \end{bmatrix} P(t) X(t) = \begin{bmatrix} 0 \\ v^*_{opt}(t) \end{bmatrix} \tag{4.29}$$

$$-\frac{dP(t)}{dt} = \mathcal{A}(x)^T P(t) + P(t)\mathcal{A}(x) - P(t)^T \mathcal{R}^+ P(t) + \mathcal{Q} \tag{4.30}$$

where $P(t_f) = \mathcal{S}$ and equation (4.30) is a Riccati equation and proofs can be found in the Appendix.

Note that the Riccati equation (4.30) includes $A(x)$ term in $\mathcal{A}$, yet we can handle this easily by replacing x by x_d: this trick is reasonable because our optimal control input, $v^*_{opt}(t)$, drives $x(t)$ to $x_d(t)$ by choosing proper Q_1, Q_2 and R. Otherwise, we would have to solve a Two Point Boundary Value Problem (TPBVP) by numerical iteration.

Proposition. The Riccati equation (4.30) can be solved by replacing x by x_d, using $\mathcal{Q}, \mathcal{R}$ which drive x to x_d.
proof : Consider $\mathcal{L} = X^T P_2 X$ as a Lyapunov function, then we can differentiate a Lyapunov function as follows:

$$\dot{\mathcal{L}} = X^T\{\mathcal{A}(x)^T P_2 + \dot{P}_2 + P_2\mathcal{A}(x)\}X + (W+V)^T P_2 X + X^T P_2 (W+V) \tag{4.31}$$

Select a Riccati equation which satisfies $\mathcal{A}(x_d)^T P_2 + \dot{P}_2 + P_2\mathcal{A}(x_d) = -\mathcal{Q} + P_2^T \mathcal{R}^+ P_2$ and consider $\Delta\mathcal{A}$ satisfying $\mathcal{A}(x_d) = \mathcal{A}(x) + \Delta A$, then we can reformulate the Riccati equation as follows:

$$\mathcal{A}(x)^T P_2 + \dot{P}_2 + P_2\mathcal{A}(x) = -\mathcal{Q} + P_2^T \mathcal{R}^+ P_2 - (\Delta\mathcal{A}^T P_2 + P_2 \Delta\mathcal{A}) \tag{4.32}$$

where intuitively, if $x \to x_d$, $\Delta\mathcal{A}$ becomes zero matrix. Also, we can reformulate $\dot{\mathcal{L}}$ as follows:

$$\begin{aligned} \dot{\mathcal{L}} &= X^T\{-\mathcal{Q} + P_2^T\mathcal{R}^+ P_2 - (\Delta\mathcal{A}^T P_2 + P_2\Delta\mathcal{A})\}X + (W+V)^T P_2 X + X^T P_2 (W+V) \\ &= -X^T\mathcal{Q}X + (\mathcal{R}^+ P_2^T X + (W+V))^T \mathcal{R}(\mathcal{R}^+ P_2^T X + (W+V)) - (W+V)^T \mathcal{R}(W+V) \\ &\quad -X^T(\Delta\mathcal{A}^T P_2 + P_2\Delta\mathcal{A})X \end{aligned} \tag{4.33}$$

We pick the optimal input $V = -\mathcal{R}^{+}P_2^T X$ and we can use the relations:

$$(W+V)^T\mathcal{R}(W+V) = V^T\mathcal{R}V = X^T P_2^T \mathcal{R}^{+} P_2 X \text{ and } W^T\mathcal{R}W = 0$$

By using these relations, we can formulate 4.33:

$$\begin{aligned}\dot{\mathcal{L}} &= -X^T\mathcal{Q}X + W^T\mathcal{R}W - (W+V)^T\mathcal{R}(W+V) - X^T(\Delta\mathcal{A}^T P_2 + P_2\Delta\mathcal{A})X \\ &= -X^T(\mathcal{Q} + P_2^T\mathcal{R}^{+}P_2 + \Delta\mathcal{A}^T P_2 + P_2\Delta\mathcal{A})X \end{aligned} \tag{4.34}$$

Thus, by choosing $\mathcal{Q}, \mathcal{R}$ large enough to guarantee $\dot{\mathcal{L}} < 0$, we can drive $x \to x_d$ (i.e., $\Delta\mathcal{A}$ becomes zero matrix). Also, we can evaluate the dynamic graph $e(t)$ by integration:

$$e^*(t) = \int_0^t v_{opt}^*(\tau)d\tau \tag{4.35}$$

Therefore, this proposed LQ optimal control framework allows us to capture pivotal development events and dynamics of the temporally evolving system.

4.5 Numerical Example

In this section, we consider numerical examples to illustrate the proposed scheme. In order to understand the method, we first present the procedure step by step with a well-known unicycle model which gives us intuition. Then, we apply our scheme to biological systems.

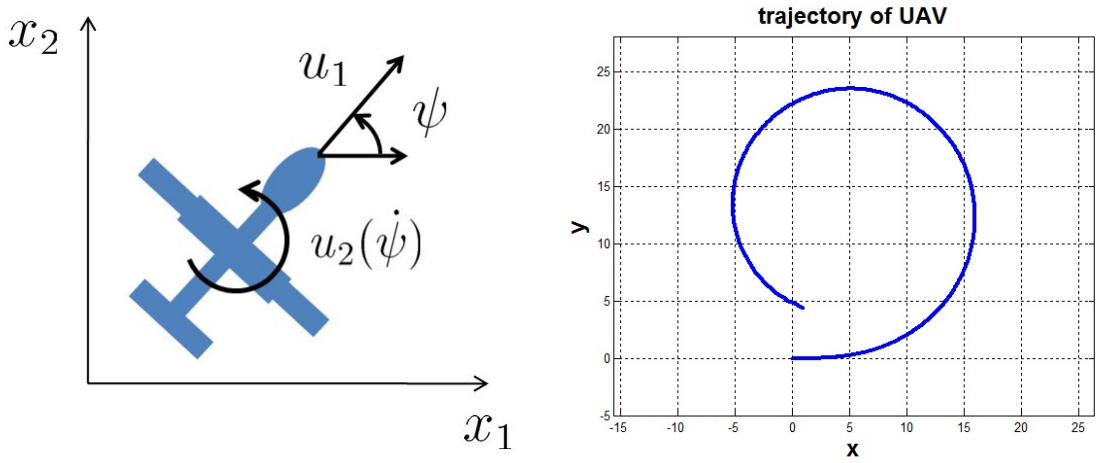
Example 2. *(A Unicycle Model)* Consider the kinematic model of a UAV:

$$\begin{aligned}\dot{x}_1 &= v\cos\psi \\ \dot{x}_2 &= v\sin\psi \\ \dot{\psi} &= u_2 \end{aligned} \tag{4.36}$$

where (x_1, x_2) is the Cartesian location of the UAV and (v, u_2) is the control input encompassing the linear and angular velocities, as shown in Figure 4.16. By introducing another state, we can extend the equations as follows:

$$\begin{aligned}\dot{x}_1^d &= x_3^d\cos x_4^d \\ \dot{x}_2^d &= x_3^d\sin x_4^d \\ \dot{x}_3^d &= u_1 \\ \dot{x}_4^d &= \dot{\psi} = u_2 \end{aligned} \tag{4.37}$$

where x_3^d is the velocity (v) of UAV and x_4^d represents the yaw angle of UAV. Suppose that we do not know the equation of motion of a UAV system (4.37) but we can access the output $y(t) = [x_1^d, x_2^d, x_3^d, x_4^d, u_1, u_2]^T = x_d(t)$. Although we do not know the exact dynamics, we can



(a) (b)

Figure 4.16: (a) A simple unicycle model (b) trajectory of the UAV.

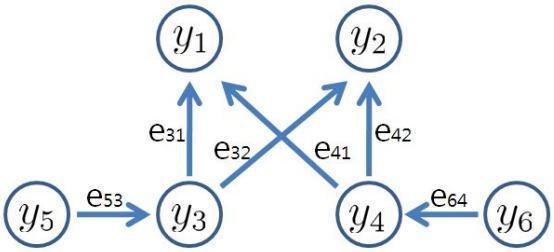
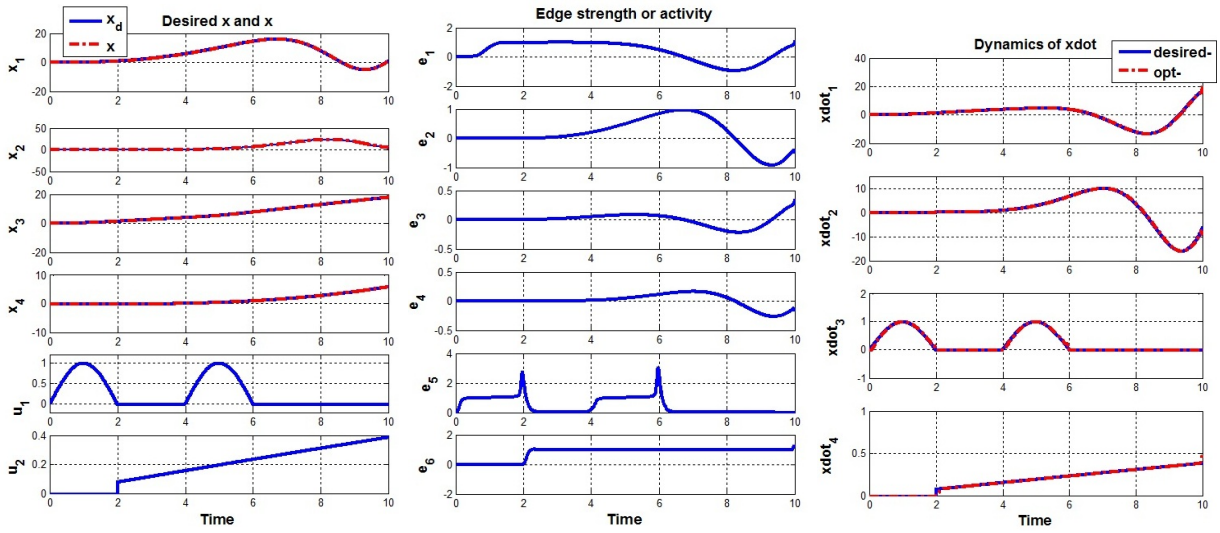


Figure 4.17: A graphical model of a simple unicycle model.

draw the *a priori* influence map shown in Figure 4.17 which shows the relation or interaction between each state. For example input $y_5(u_1)$ affects the linear velocity (y_3) of UAV and the velocity affects positions (y_1, y_2) of the UAV. Based on this connectivity map, we can formulate a linear system as follows:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} x_3 & 0 & x_4 & 0 & 0 & 0 \\ 0 & x_3 & 0 & x_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & x_6 \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \\ e_4(t) \\ e_5(t) \\ e_6(t) \end{bmatrix} = A(x)e(t) \tag{4.38}$$

Can we find the activities of all edges $e(t) = [e_{31}(t), e_{32}(t), e_{41}(t), e_{42}(t), e_{53}(t), e_{64}(t)]^T = [e_1(t), e_2(t), e_3(t), e_4(t), e_5(t), e_6(t)]^T$ based on the measurements? We can apply the proposed



(a) (b) (c)

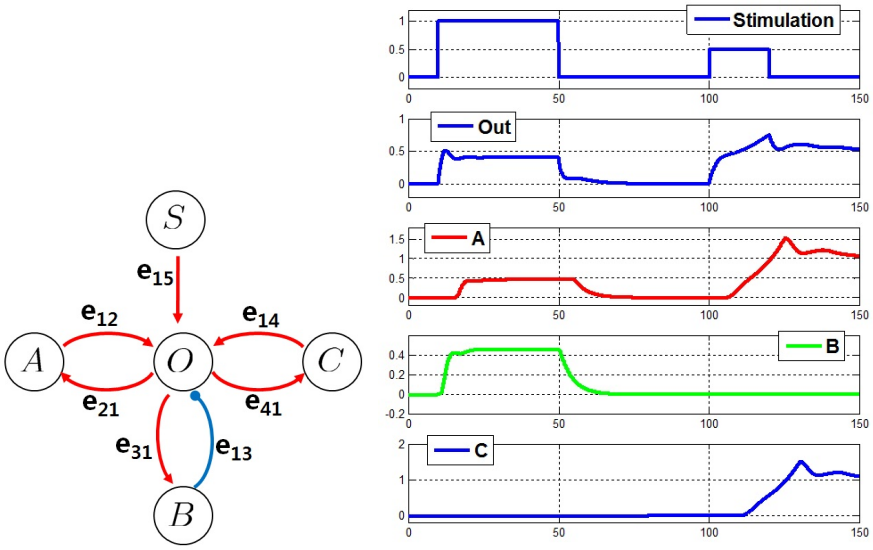
Figure 4.18: (a) x_d and x based on the optimal solution (b) the connectivity and strength of each edge (optimal solution) (c) $\dot{x}_d$ and $\dot{x}$ based on the optimal solution.

LQ optimal control problem by replacing x by measurements x_d and finding the optimal solution which shows the connectivity and strength of each edge. Also, we can compare the true nonlinear dynamic system (4.37) with the optimal solution in Figure 4.18 (c).

Figure 4.18 shows the optimal solution for example 2. For example, Figure 4.18 (b) represents the activity of the edges and we can see $e_{53}(e_5)$ and $e_{64}(e_6)$ which is close to 1 because our dynamics is nothing but integration ($\dot{x}_3 = u_1 = 1 \times y_5$ and $\dot{x}_4 = u_2 = 1 \times y_6$). Also, the piecewise linear system with the optimal solution is consistent with the true nonlinear dynamic system. Therefore, although we do not have an exact dynamic model for UAV system, we can reconstruct the time-varying linear system with *a priori* map. The optimization result would help us to understand the actual dynamics or gain intuition.

Example 3. *(Biological Signaling Pathways)* Consider the system of coupled positive and negative feedback networks as follows [96]:

$$\begin{aligned}
\frac{dA}{dt} &= k_1 h(out, \tau_A)(10 - A) - k_2 A \\
\frac{dB}{dt} &= k_1 h(out, \tau_B)(10 - A) - k_2 B \text{ (short term)} \\
\frac{dC}{dt} &= k_1 h(out, \tau_C)(10 - C) - k_2 C \text{ (long term)} \\
\frac{dO}{dt} &= (k_{sti} S + f_A A + f_C C)(10 - O) - (f_B B + k_{min}) O
\end{aligned} \tag{4.39}$$



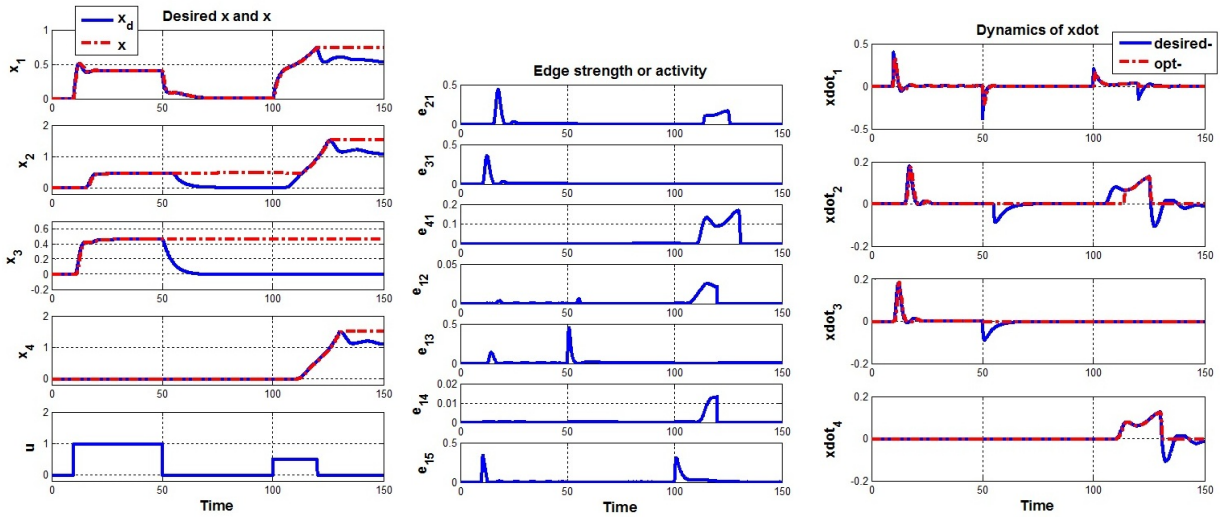
(a) (b)

Figure 4.19: (a) A graph model of simple biological network (b) simulation result (PN feedback (for short-term) and PP feedback (for long-term behavior)).

where $k_{sti} = 0.04$, $k_{min} = 0.4$, $f_A = 0.012$, $f_B = 1.5$, $f_C = 0.008$, $k_1 = 0.2$, $k_2 = 0.25$, $\tau_A = 5$, $\tau_B = 0$, $\tau_C = 10$ and $h(out, \tau) = \dfrac{out(t-\tau)^3}{out(t-\tau)^3 + 1^3}$. Here, we consider the change of topology of the network (for the short term, Positive Negative (PN) feedback (A, B only) and for the long term, Positive Positive (PP) feedback (A, C only))[96]. In Figure 4.19, we can see that output signal (O) goes to zero once the stimulation (S) or input signal goes to zero for short term behavior because of PN feedback. However, for the long term, even a small pulse input can make the output signal stay with high amplitude because of PP feedback. This abstraction model represents HER2 overexpressed breast cancer signal pathway. Define $x_d(t) = [x_1, x_2, x_3, x_4, u] = [O, A, B, C, S]$, $e(t) = [e_{21}, e_{31}, e_{41}, e_{12}, e_{13}, e_{14}, e_{15}]$ and the *a priori* map as shown in Figure 4.19 (a). Then, a piecewise linear system can be formulated as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & x_2 & x_3 & x_4 & u \\ x_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_{21} \\ e_{31} \\ e_{41} \\ e_{12} \\ e_{13} \\ e_{14} \\ e_{15} \end{bmatrix} \tag{4.40}$$

Here, we introduce constraints that all edges $e_{ij}(t)$ are positive, representing activation edges. Therefore, for inhibition edges such as $e_{13}(t)$, we replace y_3 by $-y_3$ in order to satisfy



(a) (b) (c)

Figure 4.20: (a) x and x_d (b) the activity of edges (c) $\dot{x}$ and $\dot{x}_d$

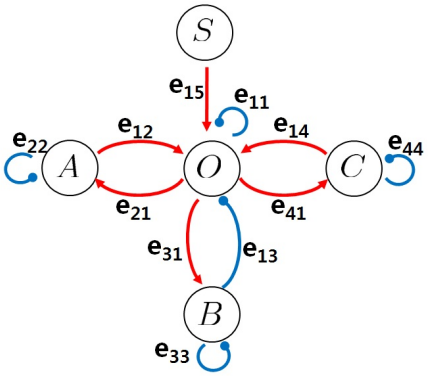
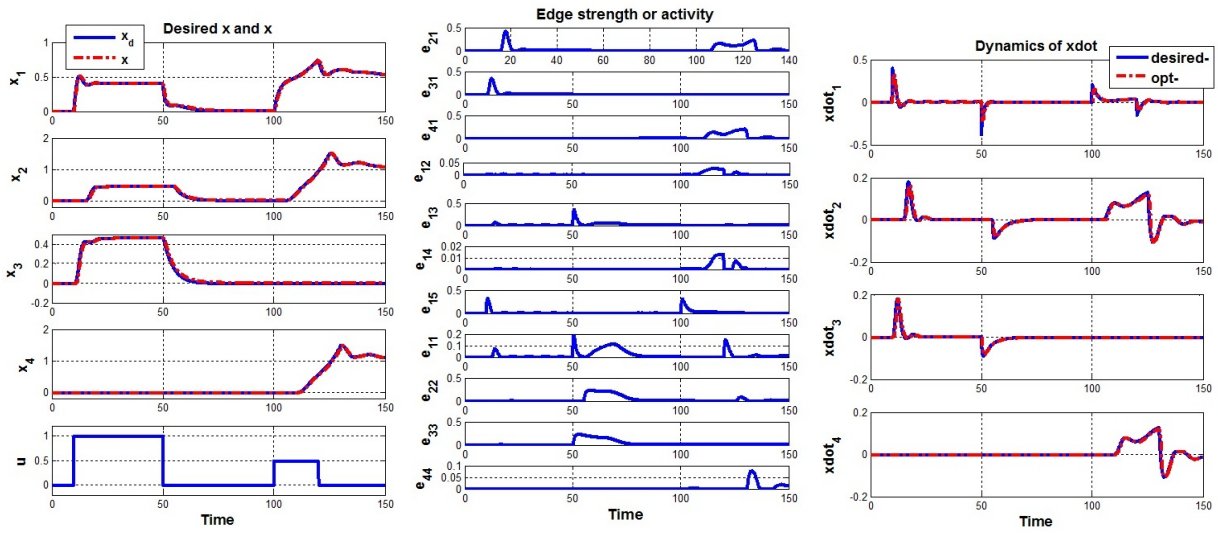


Figure 4.21: A modified graphical model of the simple biological network of Figure 4.19 (a).

our constraints. Also, we use a gradient projection method for constrained optimization which makes our feasible solution satisfy these constraints.

Figure 4.20 shows that the optimal activities of all edges ($e(t)$) cannot drive our system to be consistent with desired output (biological data) at some points. This result is expected because our simulation data are generated by equation (4.39) which includes self-degradation terms such as $-k_2A, -k_2B, -k_2C$. However, in our graphical representation in Figure 4.19, there are no self-degradation edges so there is no way to decrease the concentration of states themselves. Therefore, if our graphical model can not represent the experimental data well, our model can reveal deficiencies in the model. Then, we should suspect our graphical model and update or modify by adding new edges, which are self-degradation edges, as shown in



(a) (b) (c)

Figure 4.22: (a) x and x_d (b) the activity of edges (c) $\dot{x}$ and $\dot{x}_d$.

Figure 4.21.

Figure 4.22 shows that the optimal control input can drive our system to be consistent with desired output quite well. Here we have included degradation edges $(e_{11}, e_{22}, e_{33}, e_{44})$ so it can reflect the underlying dynamics (4.39). Also, we can understand the behavior with a systematic point of view. For example, we can see that for the short term, the edges (e_{21}, e_{31}, e_{13}) related to [A] and [B] are activated, meaning PN feedback. For long term, the edges related to [A] and [C] are activated, showing PP feedback $(e_{21}, e_{12}, e_{41}, e_{14})$. Also, for example, $e_{21}(t)$ shows a sharp pulse at the point at which [A] start to increase. We can interpret this sharp pulse as a trigger which activates the relation between x_1 and x_2. Once, $e_{21}(t)$ shows zero value meaning that the edge is deactivated (switched off). Therefore, by using continuous model, we can capture the underlying dynamics or key signal pathways over time.

4.6 Conclusions

In this chapter, we have proposed a data-driven inference scheme in order to understand and identify a model for temporally evolving biological networks. For the discrete time case, the inference problem has led to a convex optimization problem with two main penalty functions such as sparsity and reconstruction error using *a priori* information of structure. For the continuous case, the identification problem has led to an LQ control problem with two main penalty functions by which we can match the experimental data with a sparse representation using *a priori* information of structure. We show that the proposed schemes can be useful to

capture the dynamic evolution of the network and use this algorithm for studying a breast cancer signaling pathway to understand short-term and long-term behaviors by capturing the dynamic evolution of the network with a systems point of view. Also, a logical next step would be to develop a theory-driven mechanism model such as an ODE by understanding the actual dynamics.

Chapter 5

Data-driven Graph Reconstruction using Compressive Sensing

In studying biological systems, identifying the underlying gene regulatory networks from data has been important and will continue to affect the study of gene regulatory networks. We consider the problem of reconstructing a gene regulatory network structure from time series gene expression data, and in turn identifying and uncovering the underlying network structure responsible for the observed behaviors. First, we focus on a sparsely connected graph structure with all nodes accessible for measurement and no measurement noise. A method for reconstructing the graph structure without any prior knowledge of connectivity is proposed. The proposed method is based on compressive sensing, where for a measurement model $\mathcal{T} = \Omega\bar{\beta}$, we denote by $\bar{\beta}$ the network structure to be identified, Ω the sensing matrix, and $\mathcal{T}$ the measurement. The issue of coherence is discussed and it is demonstrated that incoherence in the sensing matrix can be used as a performance metric and guideline for designing effective experiments. Second, we consider a more general problem in which there might be hidden nodes which affect system dynamics. We ask whether it is still possible to reconstruct the graph structure reliably when the dynamics of a certain node is corrupted by arbitrarily large errors and in addition, all measurements are contaminated by measurement noise. By solving a two-stage convex optimization problem, the graph structure can be reconstructed reliably. In our studies, for both problems, a set of numerical examples demonstrates the performance of the proposed method. The material in this chapter is based on the work in [33][34][35].

5.1 Introduction

Mathematical modeling of biological signaling pathways can provide an intuitive understanding of network behavior in systems biology [85] [160] [102]. However, since often only incomplete knowledge of the network structure exists and the system dynamics are known to be sufficient complex, the challenge has become to show that the identified networks and

corresponding mathematical models are enough to represent the underlying system. This is also a problem of interest to the control community, with many projects related to system identification in biological networks [143][169][168][47][122][16]. There exist necessary and sufficient conditions for network reconstruction for linear time invariant (LTI) systems [70] and recent work [167] demonstrates this even while considering noise and unmodelled dynamics. However, even though there are many data-driven approaches for inferring graph structure, for time-varying or nonlinear systems, there is no statistical guarantee on how well the inferred graph structure represents the underlying system [141][77][111][110]. In order to continue to have a great impact in systems biology, identification of the graph topology from data should reveal deficiencies in the model and suggest new experimental directions [35].

Many approaches using gene expression data either focus on static data or on time series of gene expression data. The latter approach has the advantage of being able to identify dynamic relationships between genes since the spatio-temporal gene expression pattern is the result of both the network structure and integration of regulatory signals. However, data-driven reconstruction of the network structure itself remains in general a difficult problem. Moreover, measurement noise and nonlinearities in the system dynamics make this problem even more challenging. A particularly challenging problem is to identify whether or not important nodes in the graph structure are missing, how many are missing, and where these nodes are in the interconnection structure. Recent work [167] addresses the problem of data-driven network reconstruction, together with measurement noise and unmodelled, nonlinear dynamics, yet this work points out that these complications impose a limit on the reconstruction, and with stronger nonlinear terms the method fails.

Since biological regulatory networks are known to be sparse, meaning that most proteins interact with only a small number of proteins compared with the total number of possible proteins, many methods [142][77][111][110] take advantage of sparsity. Given this assumption, the methods typically use l_1-norm optimization, which leads to a sparse representation of the network and improves the ability to find the actual network structure. Even though many methods [111][110] show that the reconstruction results are fairly good, the methods cannot guarantee *exact recovery*. This stems from the fact that the *incoherence* condition for $\mathcal{T} = \Omega\bar{\beta}$ is typically not satisfied for the matrix Ω where $\bar{\beta}$ is the signal to be reconstructed, $\mathcal{T}$ is the measurement, and Ω is known as the sensing matrix. Incoherence provides a metric of performance: this is one of the motivating factors for the use of compressive sensing (CS)[23]. Such a metric can indicate us how accurately the inferred graph structure matches the true underlying structure.

In this chapter, we are interested in directed graphs such as signaling pathways and metabolic networks. We develop a new algorithm for network reconstruction based on CS. First, we focus on sparse graph structures using limited time series data with all nodes accessible and no measurement noise. In the HER2 positive breast cancer signaling pathway that we study [31][32], time series data sets consist of 8 time point measurements of 20 protein signals, and we would like to use this data to identify a graph structure which could have 20×20 or 400 edges. Applying the proposed method enables us to test the ability of the network reconstruction algorithm to accurately and efficiently recover the network structure

based on the gene expression data taken from a simulated biological pathway in which the structure is known *a priori*. Also, the limitation of the proposed method's performance is explained when the dataset has naturally high coherence, and the way to overcome this limitation by designing effective experiments is discussed.

Second, the proposed algorithm is extended to a more general but challenging problem: (1) we take into account partially corrupted data with large error and (2) we consider small measurement noise affecting all the data. For example, system dynamics may be affected by missing nodes, or data could be corrupted arbitrarily by human error. The question is whether one can still recover the graph structure reliably under these conditions. Inspired by a robust error correction method [25], the exact recovery of the graph structure can be guaranteed under suitable conditions on the node dynamics, provided that hidden nodes can affect relatively few nodes in the graph structure. We discuss whether it is possible to reconstruct the graph structure reliably when our measurement data is corrupted by noise and there exist hidden nodes in the network. Also, a set of numerical examples is provided to demonstrate the method, based on the synthetic protein expression data and RPPA (Reverse Phase Protein Assay) dataset [81] taken from the HER2 positive breast cancer pathway.

The rest of this chapter is organized as follows: Section 5.2 presents brief overview of compressive sensing. We discuss about how CS helps in reconstructing gene regulatory networks in Section 5.3 and provide numerical examples in Section 5.4. We extend the proposed algorithm to a more general problem considering hidden nodes and measurement noise in Section 5.5. Finally, conclusions are given in Section 5.6.

5.2 Compressive Sensing

Consider measurements $\mathcal{T} \in \mathbb{R}^m$ of a signal $\bar{\beta} \in \mathbb{R}^n$:

$$\mathcal{T} = \Omega\bar{\beta} \tag{5.1}$$

where $\Omega \in \mathbb{R}^{m\times n}$. Equation (5.1) may be represented in terms of the inner products of $\bar{\beta}$ with the rows Ω_i of Ω.

One key question [147] is how many measurements m are needed to exactly recover the original signal $\bar{\beta}$ from $\mathcal{T}$:

- If $m \geq n$ and Ω is a full rank matrix, then the problem is determined (or overdetermined) and may be solved uniquely for $\bar{\beta}$ (if it is overdetermined, sometimes there may be no solution).

- If $m < n$, the problem is underdetermined even if Ω has full rank. We know that $\mathcal{T} = \Omega\bar{\beta}$ restricts $\bar{\beta}$ to a subspace of $\mathbb{R}^n$, but $\bar{\beta}$ cannot be determined uniquely.

For the underdetermined case, the least squares solution $\bar{\beta}^* = \arg\min_{\bar{\beta}} ||\bar{\beta}||_{l_2} = \Omega^*(\Omega\Omega^*)^{-1}\mathcal{T}$ is typically used as the "best guess" in many applications. However, often the least squares

solution is not satisfactory, especially if $\bar{\beta}$ is sparse, meaning that many of its components are zero, because least squares regularization does not encourage sparsity but distributes the coefficients, meaning that almost all components are nonzero [147]. Yet if $\bar{\beta}$ is sparse, one might intuitively expect that fewer than n measurements are needed to recover $\bar{\beta}$. It is thus of interest to obtain a good estimator for underdetermined problems in which $\bar{\beta}$ is assumed to be *s-sparse* for some $1 \leq s \leq n$, meaning that at most s of the elements of $\bar{\beta}$ can be non-zero. Therefore, in principle, we need only s measurement to reconstruct $\bar{\beta}$. This is the main motivation for CS:

"CS theory asserts that one can recover certain signals and images from far fewer samples or measurements than traditional method use. For example, the number of measurements needed to recover $\bar{\beta} \in \mathbb{R}^n$ is a number of measurements proportional to the compressed size of the signal (S), rather than the uncompressed size (n) [24]."

To make this possible, CS relies on two properties: *sparsity*, which pertains to the signals of interest, and *incoherence*, which pertains to the sensing matrix. CS exploits the fact that many natural signals are sparse or compressible in the sense that they have concise representations when expressed in the proper basis. Incoherence expresses the idea that objects having a sparse representation in a certain basis should be spread out in the domain in which they are acquired. For example, a Dirac delta function in the time domain is spread out in the frequency domain. Much of the CS literature [23][24][27] discusses sensing mechanisms in which information about a signal $\bar{\beta} \in \mathbb{R}^n$ is obtained by a linear functional recording the values as follows:

$$\mathcal{T}_k = \langle \Omega_k, \bar{\beta} \rangle, \quad k = 1, ..., m \tag{5.2}$$

where $\mathcal{T}_k \in \mathbb{R}$ is the kth component of the measurement $\mathcal{T} \in \mathbb{R}^m$, written in terms of the inner product of signal $\bar{\beta}$ with the k-th row $\Omega_k \in \mathbb{R}^n$ of the sensing matrix $\Omega \in \mathbb{R}^{m \times n}$. Therefore, we simply measure the correlation between the object $\bar{\beta}$, which we wish to reconstruct, and the sensing matrix Ω and then, $\mathcal{T}$ is a vector of sampled values of $\bar{\beta}$ in the time or space domains. For example, if the sensing waveforms, Ω_k, are indicator functions of pixels, then $\mathcal{T}$ is the digital image data collected by image sensors. If the sensing waveforms are sinusoids, then $\mathcal{T}$ is a vector of Fourier coefficients [27]. Since the time series gene expression data are represented by integration of regulatory signals through the network structure, we could formulate the gene regulatory relation into equation (5.2) where $\mathcal{T}$ and Ω are constructed by time series gene expression data and $\bar{\beta}$ represents the network structure which is assumed to be sparse and unknown.

Proposition [147]. Suppose that any $2s$ columns of an $m \times n$ matrix Ω are linearly independent (this is a reasonable assumption if $m \geq 2s$). Then, any *s-sparse* signal $\bar{\beta} \in \mathbb{R}^n$ can be reconstructed uniquely from $\Omega\bar{\beta}$.

In fact, the proof [147] of the above proposition also shows how to reconstruct an *s-sparse* signal $\bar{\beta} \in \mathbb{R}^n$ from the measurement $\mathcal{T} = \Omega\bar{\beta}$ where $\bar{\beta}$ is the unique sparsest solution to $\mathcal{T} = \Omega\bar{\beta}$;

$$\bar{\beta} = \arg\min_{\bar{\beta}} ||\bar{\beta}||_{l_0} \text{ subject to } \mathcal{T} = \Omega\bar{\beta} \tag{5.3}$$

where $||\bar{\beta}||_{l_0} := \sum_{i=1}^{n} I(\bar{\beta}_i \neq 0)$ is the cardinality of $\bar{\beta}$. However, l_0-minimization is computationally intractable (NP-hard in general). A simple but very effective way to solve l_0-optimization is l_1-minimization (or basis pursuit) which is a convex optimization problem and can be solved by linear programming. Several theoretical results [26] ensure that basis pursuit works whenever the measurement or sensing matrix Ω is sufficiently *incoherent*, defined as follows:

Definition [23] The *coherence* of a matrix M is denoted by

$$\mu(M) = \max_{j<k} \frac{|\langle M_j, M_k \rangle|}{||M_j||_2 ||M_k||_2}$$

where M_j and M_k denote columns of M. We can say that a dictionary[1] is *incoherent* if μ is small. Coherence is a key property in the compressed sensing framework because if two columns are closely correlated, it will be impossible to distinguish whether the sparse component in the signal comes from one or the other.

Theorem 1 *[26] Let $\xi_1, ..., \xi_m \in \{1, ..., n\}$ be chosen randomly. Then with high probability, every one-dimensional s-sparse discrete signal $f : \{1, ..., n\} \to \mathbf{C}$ can be recovered from a partial collection $\hat{f}(\xi_1), ..., \hat{f}(\xi_m)$ of Fourier coefficients as long as $m > cs \log n$ for some positive constant c where $\hat{f}(\xi) := \frac{1}{n} \sum_{x=1}^{n} f(x) e^{-2\pi i x \xi / n}$.*

Numerical experiments suggest that in practice, most *s-sparse* signals are in fact recovered exactly once $m \geq 4s$ or so [23]. Therefore, if the sensing matrix Ω satisfies the incoherence condition, the signal $\bar{\beta}$, which is assumed to be sufficiently sparse and represents the graph structure in our case, can be exactly recovered from the condensed dataset without any prior knowledge such as the number of nonzero elements, their locations, and their values. On the other hand, if the incoherence condition is not satisfied, the exact recovery cannot be guaranteed [35][24]. However, by adding more informative dataset, the coherence can be reduced.

[1] A dictionary in $\mathbb{R}^d$ is a finite collection of unit-norm vectors that spans $\mathbb{R}^d$. If there are n elements in the dictionary, we say its size is n. A dictionary can be represented by a $d \times n$ matrix D whose columns are the elements of the dictionary.

5.3 How does CS help in reconstructing gene regulatory networks?

A novel sensing/sampling paradigm known as compressed sensing or CS goes against the common wisdom in data acquisition [24]. To make this possible, CS designs efficient sensing mechanisms, captures the useful information content embedded in a sparse signal and condenses it into a small amount of data. This remarkable feature of CS is that the sensing is completely nonadaptive so one simply chooses a sensing matrix or selects a collection of sensing vectors (i.e., each row of sensing matrix) *a priori* and measures correlations between the signal and sensing vectors [7].

We formulate the reconstruction problem into the CS framework by constructing the sensing matrix (Ω) and measurement ($\mathcal{T}$) from the given time series gene expression data. Here, the graph structure which is denoted by $\bar{\beta}$ is nonadaptive, or fixed, even though the strength of connection may change over time. In graph theory, a digraph can be represented by $G = (V, E)$ where V and E represent vertices and edges respectively. Since the dynamics of biological networks are typically unknown, often the best we can do is to derive a linear dynamical system representation which encodes connectivity between genes and the relative strength of the connection. For example, any arbitrary complex directed graph (G) can be represented by a time-varying linear system $\dot{x} = A(t)x + B(t)u$ where $x \in \mathbb{R}^n$ represents gene expression, $A(t) \in \mathbb{R}^{n\times n}$ is a system matrix or unknown influence map, $B(t) \in \mathbb{R}^{n\times q}$ is the known input matrix and $u \in \mathbb{R}^q$ is the control input which can directly affect certain nodes (i.e., drug treatments). Thus, any dynamical system can be represented by a graph model $G = (V, E) = (A, B)$ or vice versa [99].

Although one could consider continuous time/space signals, we restrict our attention to discrete signals. Let $X^k \in \mathbb{R}^n$ and $Y^k \in \mathbb{R}^n$ be the measured data where k is a time step index, and $U^k \in \mathbb{R}^q$ be the known input. The relationship between X^k, Y^k and U^k can be described by $Y^k \triangleq \Delta X^k/\Delta T = (X^{k+1} - X^k)/\Delta T = A^k X^k + B^k U^k$ where $B^k \in \mathbb{R}^{n\times q}$ is assumed to be known and ΔT represents a sampling time. The main goal is to find a sparse graph representation, A^k, using measured time series data. A directed graph A^k can be denoted as follows:

$$A^k = \bigcup_{i=1}^{d} \mathcal{P}^k(G_i) = \sum_{i=1}^{d} \alpha_i^k G_i = \sum_{i=1}^{d} \alpha_i^k \mathbf{s}_i \mathbf{t}_i^T \tag{5.4}$$

where d is the number of non-zero elements or edges in the directed graph, $\mathcal{P}^k$ is a projection onto the basis of the graph structure and $\alpha_i^k \geq 0$ is a scaling factor. Here, a graph structure can be denoted by $G_i (= \mathbf{s}_i \mathbf{t}_i^T \in \mathbb{R}^{n\times n})$ where $\mathbf{s}_i, \mathbf{t}_i \in \mathbb{R}^n$ are standard bases. By definition, bases of the graph structures G_i and G_j are orthogonal to each other, e.g., $G_i \perp G_j$ or $(\mathbf{s}_i^T \mathbf{s}_j)(\mathbf{t}_i^T \mathbf{t}_j) = 0$. As an example, if the jth node activates the ith node (i.e., $A_{ij}^k > 0$), this can be represented by $A_{ij}^k = \alpha^k \mathbf{e}_i \mathbf{e}_j^T$ where $\mathbf{e}_i, \mathbf{e}_j$ are the standard or canonical basis and α^k is a time-varying scaling factor. Here, different scaling represents the relative strength of the connection over time even though the structure itself is fixed. In the generic sparse setting,

union models are nonlinear but we denote the union by both time-varying scaling factor and the basis of the graph structure in equation (5.4). Figure 5.1 shows a simple example:

$$A = \begin{bmatrix} \alpha_1 \mathbf{e}_2 & \alpha_2 \mathbf{e}_3 \end{bmatrix}_{n \times d} \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \end{bmatrix}_{d \times n} \tag{5.5}$$

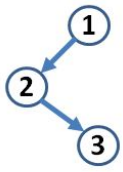
where A represents the influence map which represents both connectivity (or graph structure denoted by $\mathbf{e}_2\mathbf{e}_1^T, \mathbf{e}_3\mathbf{e}_2^T$) and the relative strength of the connection denoted by α_1 and α_2 where $\mathbf{e}_2\mathbf{e}_1^T, \mathbf{e}_3\mathbf{e}_2^T$ are the basis elements of the graph structure. Therefore, for the time-varying case, α_i can change over time by which we can handle nonlinear dynamics. The difficulty of inferring graph structure is to show that the inferred graph structure is enough to accurately represent the underlying system because of the limited measurement. CS helps us to tackle this difficulty by providing a guarantee of the exact recovery under incoherence conditions.

5.3.1 Problem Formulation

In this section, we present how to formulate the original reconstruction problem into CS framework. Consider a linear time invariant map with no control input ($U^k = \mathbf{0}$) for simplicity (without loss of generality, if there is a control input, we can subtract that term, i.e., $Y^k - B^k U^k = AX^k$):

$$Y^k = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2n} \\ \dots & \dots & \dots & \dots \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nn} \end{bmatrix} X^k = AX^k \tag{5.6}$$

where $X^k, Y^k \in \mathbb{R}^n$ are measurements at time step k, each α_{ij} represents the connection between node i and j, or the matrix A thus defines the corresponding influence map at



$$\text{(input) } \mathbf{e}_2 + \mathbf{e}_3 = \begin{bmatrix} 0 \\ + \\ + \end{bmatrix} \leftarrow \begin{bmatrix} 0 & 0 & 0 \\ + & 0 & 0 \\ 0 & + & 0 \end{bmatrix} = \alpha_1 \mathbf{e}_2 \mathbf{e}_1^T + \alpha_2 \mathbf{e}_3 \mathbf{e}_2^T = A$$

$$\downarrow$$

$$\begin{bmatrix} + & + & 0 \end{bmatrix} = \mathbf{e}_1^T + \mathbf{e}_2^T \text{ (output)}$$

Figure 5.1: A simple example ($n = 3$, $A \in \mathbb{R}^{3\times3}$): graph structure is sparse and can be represented by the direct sum of only two basis graphs or two edges ($d = 2 \ll 9$) where $(\mathbf{e}_2^T\mathbf{e}_3)(\mathbf{e}_1^T\mathbf{e}_2) = 0$.

time k (note that here, A is time invariant). However, in general, a biological system is more complex and has nonlinear behavior, so we could also consider a more general case as follows:

$$Y^k = \begin{bmatrix} \alpha_{11}\xi(x_1^k) & \alpha_{12}\xi(x_1^k) & ... & \alpha_{1n}\xi(x_1^k) \\ \alpha_{21}\xi(x_2^k) & \alpha_{22}\xi(x_2^k) & ... & \alpha_{2n}\xi(x_2^k) \\ ... & ... & ... & ... \\ \alpha_{n1}\xi(x_n^k) & \alpha_{n2}\xi(x_n^k) & ... & \alpha_{nn}\xi(x_n^k) \end{bmatrix} \begin{bmatrix} \phi(x_1^k) \\ \phi(x_2^k) \\ ... \\ \phi(x_n^k) \end{bmatrix} \tag{5.7}$$

where x_i^k represents the ith components of X^k, α_{ij} represents a graph structure with nonlinear components $\xi(\cdot)$ and $\phi(\cdot)$. In systems biology, $\xi(\cdot)$ and $\phi(\cdot)$ are nonlinear functions of the state x_i^k in many cases, for example, mass action kinetics or Hill functions $\phi(x_i^k) = \frac{{x_i^k}^H}{1+{x_i^k}^H}$. This model (5.7) can describe different model classes such as simple biochemical reactions or gene regulatory networks. For example, if we choose constants for the diagonal components, $\alpha_{ii}\xi(x_i^k) = \alpha_{ii}$; linear terms for the off-diagonal components, $\alpha_{ij}\xi(x_i^k) = \alpha_{ij}x_i^k$; and linear terms for the $\phi(\cdot)$, $\phi(x_i^k) = x_i^k$; then the dynamics of x_i can be described as $y_i^k = (x_i^{k+1} - x_i^k)/\Delta T = \alpha_{ii}x_i^k + \sum_{j=1,j\neq i}^n \alpha_{ij}x_i^k x_j^k$ where α_{ij} represents connections or edges between node i and j or simply the reaction rate of the mass action kinetics. Therefore, in principle, we can consider a model that includes all monomials, bionomials, and other combinations in dictionary. Here, for simplicity in explaining the main idea, first we consider the case in which $\xi(\cdot) = 1$, thus a time invariant map defined solely by the α_{ij}, but we still use a nonlinear term $\phi(\cdot)$. The general case will be considered with the numerical examples presented later. Equation (5.7) can be reformulated as follows:

$$Y^k = \left(\sum_{l=1}^{d} \beta_l \mathbf{e}_{v_l} \mathbf{e}_{w_l}^T \right) \Phi(X^k) \tag{5.8}$$

where $\mathbf{e}_{\{\cdot\}} \in \mathbb{R}^n$ is a unit vector, u_l, v_l, w_l are indicator indices such that $1 \leq u_l, v_l, w_l \leq n$, d is the number of nonzero elements or α_{ij} (i.e., edges of the graph structure) which is unknown but assumed to be small ($d \ll n^2$) and β_l is the corresponding nonzero α_{ij}. We denote $\Phi(X^k) = \begin{bmatrix} \phi(x_1^k) & \phi(x_2^k) & ... & \phi(x_n^k) \end{bmatrix}^T$ and define $\mathcal{Y}^k = \Psi^k Y^k$ where $\Psi^k (= \begin{bmatrix} \psi_1^k, ..., \psi_n^k \end{bmatrix})$ is a randomly chosen orthonormal matrix with independent identically distributed (i.i.d.) random variables (i.e., $\psi_i^k \perp \psi_j^k$ and $||\psi_i^k||_2 = 1$) chosen to reduce coherence in the sensing matrix which will be discussed in equation (5.12) and (5.13):

$$\begin{aligned} \mathcal{Y}^k &= \Psi^k Y^k = \Psi^k \left(\sum_{l=1}^{d} \beta_l \mathbf{e}_{v_l} \mathbf{e}_{w_l}^T \right) \Phi(X^k) \\ &= \begin{bmatrix} \psi_{v_1}^k \Phi(X^k)^T & ... & \psi_{v_d}^k \Phi(X^k)^T \end{bmatrix}_{n \times n \cdot d} \times \begin{bmatrix} \beta_1 \mathbf{e}_{w_1} \\ ... \\ \beta_d \mathbf{e}_{w_d} \end{bmatrix}_{n \cdot d \times 1} + \sum_{j=1}^{n^2-d} 0 \cdot \psi_{v_j}^k \Phi(X^k)^T \mathbf{e}_{w_j} \\ &= \begin{bmatrix} \psi_1^k \Phi(X^k)^T & ... & \psi_n^k \Phi(X^k)^T \end{bmatrix}_{n \times n^2} \bar{\beta}_{n^2 \times 1} \\ &= \mathcal{X}^k(\Psi^k, \Phi(X^k))\bar{\beta} \end{aligned} \tag{5.9}$$

where $d, u_l, v_l, w_l, \beta_l$ are unknowns. By indicating possible model classes in $\mathcal{X}^k(\cdot)$, we can estimate $\bar{\beta}$ based on l_1-norm optimization. For example, consider a 3-node directed graph ($n = 3$) as shown in Figure 5.1 and then there exist only 3 edges among possible 9 edges (n^2), i.e., the graph structure can be described in a matrix form as $\begin{bmatrix} 0 & 0 & 0 \\ \alpha_{21} & 0 & 0 \\ 0 & \alpha_{32} & 0 \end{bmatrix}$ and then $\bar{\beta} = \begin{bmatrix} 0 & 0 & 0 & \alpha_{21} & 0 & 0 & 0 & \alpha_{32} & 0 \end{bmatrix}^T \in \mathbb{R}^{9\times 1}$. Also, for M measurements, or different experimental datasets, we can stack all measurements as follows:

$$\mathcal{T} = \begin{bmatrix} \mathcal{Y}^1 \\ \mathcal{Y}^2 \\ \dots \\ \mathcal{Y}^M \end{bmatrix}_{M\cdot n\times 1} = \begin{bmatrix} \mathcal{X}^1 \\ \mathcal{X}^2 \\ \dots \\ \mathcal{X}^M \end{bmatrix}_{M\cdot n\times n^2} \bar{\beta}_{n^2\times 1} = \Omega(\mathcal{X})\bar{\beta} \tag{5.10}$$

where $\Omega(\mathcal{X})$ is denoted as our sensing matrix. In (5.10), $\bar{\beta}$ represents the s-sparse network structure which we want to reconstruct from the data by solving the l_1-norm optimization:

$$\min |\bar{\beta}|_{l_1} \quad \text{subject to} \quad \mathcal{T} = \Omega(\mathcal{X})\bar{\beta} \tag{5.11}$$

5.3.2 Reconstructing GRNs (Problem Solution)

In this section, the condition for exact recovery of $\bar{\beta}$ in (5.10) will be discussed.

Proposition. If the sensing matrix $\Omega(\mathcal{X})$ which we construct from time series data multiplied by a randomly chosen orthonormal matrix, Ψ^k, has $2s$ linearly independent columns, then any *s-sparse* network structure $\bar{\beta}$ can be reconstructed uniquely from $\mathcal{T} = \Omega(\mathcal{X})\bar{\beta}$.

proof: (Suppose not) then there are two *s-sparse* graph structures $\bar{\beta}_1, \bar{\beta}_2$ with $\Omega(\mathcal{X})\bar{\beta}_1 = \Omega(\mathcal{X})\bar{\beta}_2$ (or $\Omega(\mathcal{X})(\bar{\beta}_1 - \bar{\beta}_2) = 0$). However, $\bar{\beta}_1 - \bar{\beta}_2$ is $2s$-*sparse*, so there is a linear dependence between $2s$ columns of $\Omega(\mathcal{X})$ (contradiction).

The above proposition asserts that if the unknown *s-sparse* signal $\bar{\beta}$ is reasonably sparse, it is possible to recover $\bar{\beta}$ by convex optimization under the incoherence condition on the sensing matrix which is required by the traditional CS assumption. For example, $\Omega(\mathcal{X}) \in \mathbb{R}^{M\cdot n\times n^2}$ has $2s$ linearly independent columns and if $M \cdot n < 2s$, the number of data samples M can be increased by generating $\{\mathcal{X}^k, \mathcal{Y}^k\}$ based on the given dataset $\{X^k, Y^k\}$ with different Ψ^k. Although the sensing matrix $\Omega(\mathcal{X})$ is composed of the redundant dictionaries $\Phi(X^k)^T$, the coherence of the sensing matrix can be reduced by multiplying a randomly chosen orthonormal matrix Ψ^k at each time step k. For example, consider the following case

in which two columns from the sensing matrix are chosen:

$$\langle \begin{bmatrix} \phi(x_1^1)\psi_1^1 \\ \phi(x_1^2)\psi_1^2 \\ \dots \\ \phi(x_1^M)\psi_1^M \end{bmatrix}, \begin{bmatrix} \phi(x_1^1)\psi_2^1 \\ \phi(x_1^2)\psi_2^2 \\ \dots \\ \phi(x_1^M)\psi_2^M \end{bmatrix} \rangle = \phi(x_1^k)^2 \sum_{k=1}^{M} {\psi_1^k}^T \psi_2^k = 0 \tag{5.12}$$

where the two columns of the sensing matrix are orthogonal so linearly independent (zero coherence). However, some parts of the matrix $\Omega(\mathcal{X})$ might have high correlation because of redundant dictionaries. As an example, consider the first and second columns of the sensing matrix:

$$\langle \begin{bmatrix} \phi(x_1^1)\psi_1^1 \\ \phi(x_1^2)\psi_1^2 \\ \dots \\ \phi(x_1^M)\psi_1^M \end{bmatrix}, \begin{bmatrix} \phi(x_2^1)\psi_1^1 \\ \phi(x_2^2)\psi_1^2 \\ \dots \\ \phi(x_2^M)\psi_1^M \end{bmatrix} \rangle = \sum_{k=1}^{M} \phi(x_1^k)^T \phi(x_2^k) \tag{5.13}$$

$$\mu_{1,2} = \frac{\sum_{k=1}^{M} \phi(x_1^k)^T \phi(x_2^k)}{\sqrt{\sum_{k=1}^{M} \phi(x_1^k)^2} \sqrt{\sum_{k=1}^{M} \phi(x_2^k)^2}} \tag{5.14}$$

What if time series gene expressions are correlated each other? Is it still possible to reduce coherence of the sensing matrix in this case? For example, if time-course gene expression of two different genes x_1^k, x_2^k are closely correlated over time, the corresponding sensing matrix is not incoherent. Intuitively, as an example, if the gene expression data of two different genes are exactly the same over time, it is impossible to distinguish whether the influence comes from one or the other. This is a natural property in the CS framework. In order to reduce the coherence, a more informative dataset should be added to our sensing matrix. In biological systems, this can be done by inhibiting or stimulating a certain gene by applying inhibitors or enzymes which can manipulate the gene expression.

5.3.3 A simple linear map case

Consider a simple linear model which can explain the link between coherence and eigenvalues, eigenvectors or initial conditions. Since biological systems are typically much more complex than a simple linear model, this cannot give us an exact recipe for generating the incoherent sensing matrix; however, this simple linear model can be used to illustrate how inhibition or stimulation helps reduce the coherence. A simple linear model can be denoted by:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{5.15}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times n}$. Let λ_i be an eigenvalue of $\mathbf{A}$ and $\mathbf{r}_i$ be the corresponding eigenvector (i.e., $\mathbf{A}\mathbf{r}_i = \lambda_i \mathbf{r}_i$). If $\mathbf{A}$ is diagonalizable, then any vector in an n-dimensional

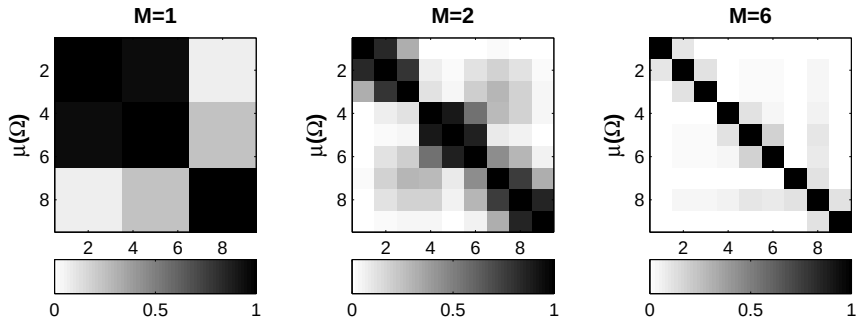


Figure 5.2: Reducing coherence of sensing matrix ($n = 3$, $A \in \mathbb{R}^{3\times 3}$): we randomly generate kth time step input and stack measurements (M). By increasing the number of measurements, M, we can reduce the coherence of our sensing matrix Ω even though we generate the input data randomly. If we add more informative dataset in a systematic way, we can reduce coherence more effectively with fewer M.

space can be represented by a linear combination of the right and left eigenvector (denoted $\mathbf{l}_i$) of the matrix $\mathbf{A}$:

$$\mathbf{x}(t) = \sum_{i=1}^{n} (\mathbf{l}_i \cdot \mathbf{x}_0)\mathbf{r}_i e^{\lambda_i t} = \sum_{i=1}^{n} c_i \mathbf{r}_i e^{\lambda_i t} \tag{5.16}$$

where $\mathbf{x}_0 = \mathbf{x}(t=0)$ and $c_i \triangleq \mathbf{l}_i \cdot \mathbf{x}_0$. Consider the first and second columns of the sensing matrix and calculate the coherence (note $\phi(x_i^k) = x_i$ for this simple linear case):

$$\begin{aligned} &\left\langle \begin{bmatrix} \phi(x_1^1)\psi_1^1 \\ \phi(x_1^2)\psi_1^2 \\ \dots \\ \phi(x_1^M)\psi_1^M \end{bmatrix}, \begin{bmatrix} \phi(x_1^1)\psi_1^1 \\ \phi(x_1^2)\psi_1^2 \\ \dots \\ \phi(x_1^M)\psi_1^M \end{bmatrix} \right\rangle = \sum_{k=1}^{M} x_1^k x_2^k \\ &= \sum_{k=1}^{M} \left(\sum_{p=1}^{n} \sum_{q=1}^{n} c_p c_q e^{\lambda_p k\Delta T + \lambda_q k\Delta T} \mathbf{r}_{p,1}\mathbf{r}_{q,2} \right) \end{aligned} \tag{5.17}$$

where $x_1^k = x_1(k\Delta T)$, $x_2^k = x_2(k\Delta T)$, $\mathbf{r}_{p,1}$ and $\mathbf{r}_{q,2}$ represent the first component of $\mathbf{r}_p$ and the second component of $\mathbf{r}_q$ respectively. Therefore, the coherence between separate time series vectors depends on the initial conditions (c_p, c_q), eigenvalues $(e^{(\lambda_p+\lambda_q)k\Delta T})$ and eigenvectors $(\mathbf{r}_{p,1}, \mathbf{r}_{q,2})$. The result indicates that if the gene expression data of two different genes are highly correlated, they should be isolated from each other. For example, when the dynamics of two nodes are exactly the same because they have the same eigenvector, then experiments to inhibit or stimulate certain genes could be used to differentiate them [35]. Figure 5.2 shows that the coherence of the sensing matrix is decreased as we increase the number of measurements M.

5.4 Numerical Examples

In this section, numerical examples are illustrated, and the incoherence condition and reconstruction result will be discussed.

Example 1.(*Time invariant map A whose nonzero elements are either* 1 *or* -1): a random graph structure is generated in which there is no isolated node and the elements of A are also randomly chosen ($a_{ij} \in \{-1, 0, +1\}$) as shown in Figure 5.3(a) and the data is generated by simulating a linear model $Y^k = A\Phi(X^k) = AX^k$ in Figure 5.3(b). From now on, throughout the paper, the performance of CS will be compared with the performances of the $||\cdot||_{l_1}$ and $||\cdot||_{l_2}$ norm optimizations, defined below[1]:

$$||\cdot||_{l_1} : \min_A \quad \mathbf{1}^T|A|\mathbf{1} \quad \text{s.t. } Y^k = AX^k \quad {}^\forall k = 1, ..., M$$
$$||\cdot||_{l_2} : \min_A \quad ||A||_{l_2} \quad \text{s.t. } Y^k = AX^k \quad {}^\forall k = 1, ..., M$$

Figure 5.3 (c) shows the result of the inferred graph structure based on given $\{X^k, Y^k\}$ without any *a priori* information. Here, there are 7 states ($n = 7$) so the directed graph structure A has 49 elements. So, the X-axis of Figure 5.3 (c) shows each element of graph structure. For example, when A is in $\mathbb{R}^{7\times7}$, there are 49 elements in the graph structure. l_2-regularization does not encourage sparsity but distributes the coefficients to be more similar to each other. Also, for the small M ($\leq n$) case, both l_1-optimization and CS give similar results (exact recovery).

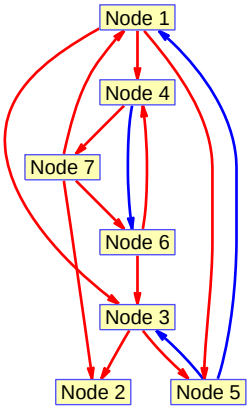
Example 2.(*Time invariant map A whose elements are randomly chosen*): a more general graph structure is considered whose elements are randomly chosen (not binary) and more data samples ($M > n$).

Figure 5.4 (c) shows the inferred graph structure. The inferred graph structure using CS is quite accurate as compared to l_1- and l_2-norm optimizations but it is not an exact reconstruction because here, the incoherent condition is not satisfied. Also, for l_1-norm optimization when $M > n$, the solution is infeasible because of redundancy (see Appendix). So, the l_1-norm optimization problem is modified as follows when $M > n$:

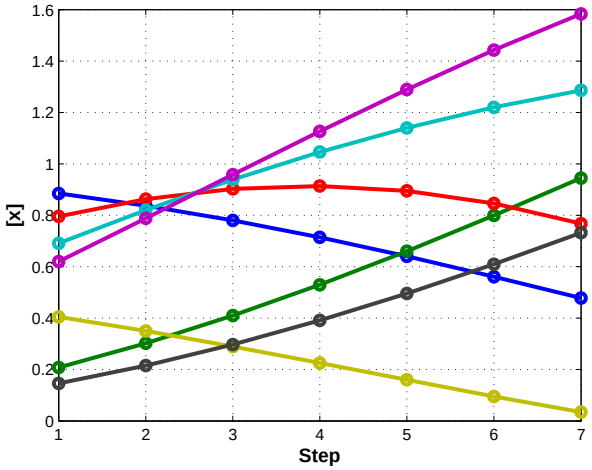
$$\min \mathbf{1}^T|A|\mathbf{1} + \gamma \sum_{k=1}^{M} ||Y^k - AX^k||_2$$

where γ is a tuning parameter or trade-off between sparsity and reconstruction error.

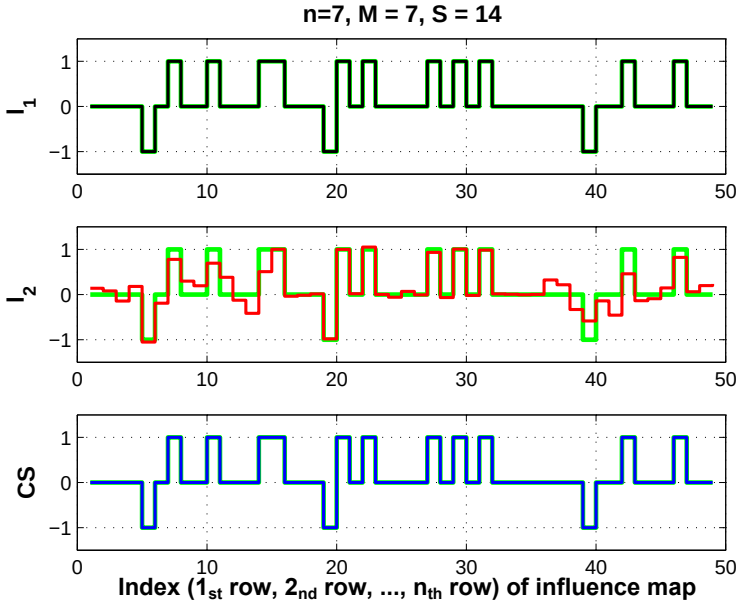
[1] $||\cdot||_{l_1}$ refers to the element-wise sum of absolute values (note that it is not the operator l_1 norm). $||\cdot||_{l_2}$ refers to the Frobenius norm.



(a)

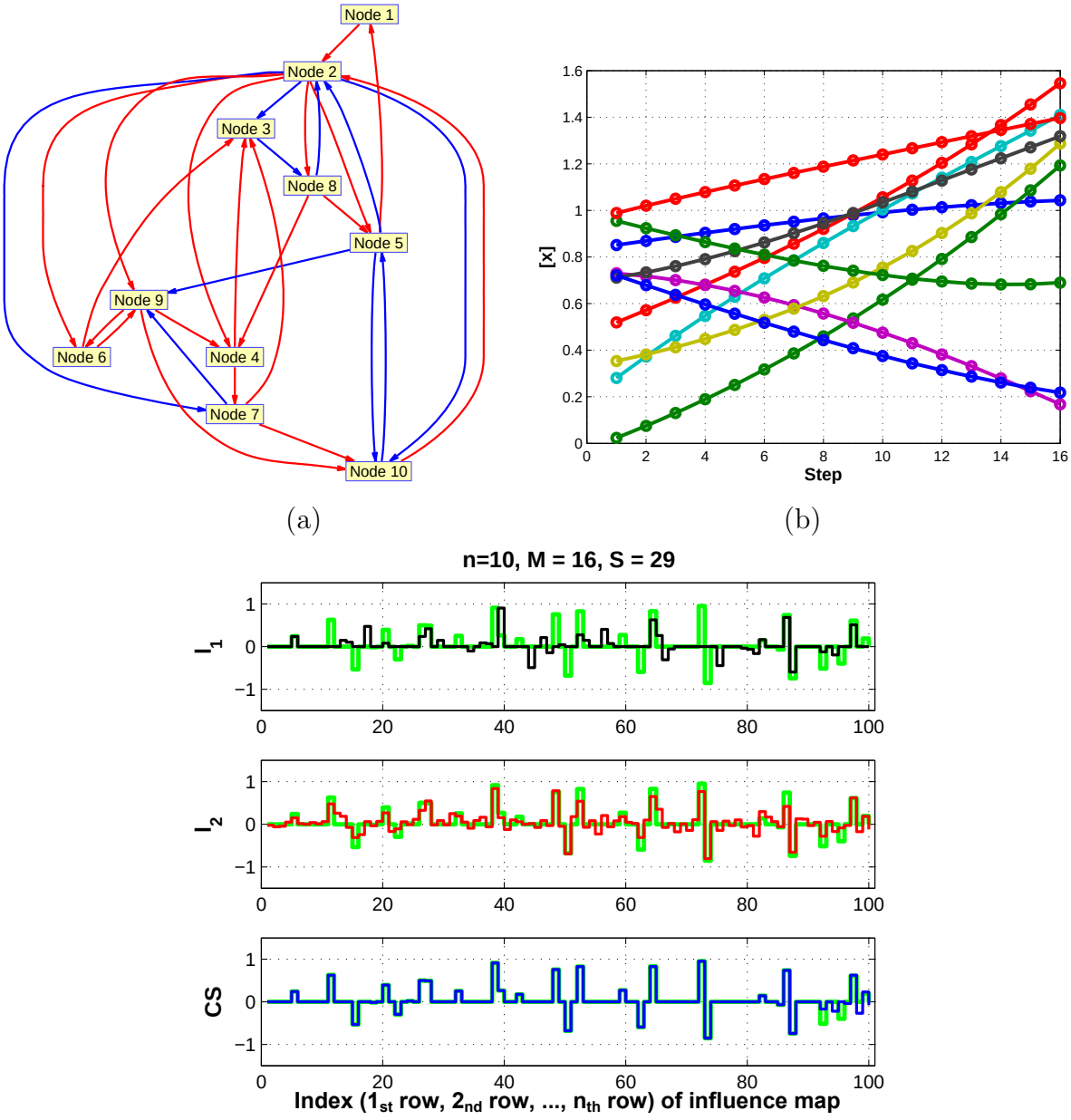


(b)



(c)

Figure 5.3: (Example 1) 7 states, 14-sparsity ($14 \ll 49$), 7 time points: (a) randomly generated graph (red:activation, blue:inhibition), (b) simulated data based on (a), (c) reconstruction based on optimization l_1, l_2 and CS where the green line represents the true structure.



(a) (b)

(c)

Figure 5.4: (Example 2) 10 states, 29-sparsity ($29 \ll 100$), 16 time points: randomly generated graph (red:activation, blue:inhibition), (b) simulated data based on (a), (c) reconstruction based on optimization l_1, l_2 and CS where the green line represents the true structure.

5.4.1 Incoherent Sensing Matrix: design a new biological experiment in a smart way

As we mentioned above, if the sensing matrix has high coherence, we need to acquire more measurements or make our sensing matrix the satisfy the requirement $m > cs \log n$ (prac-

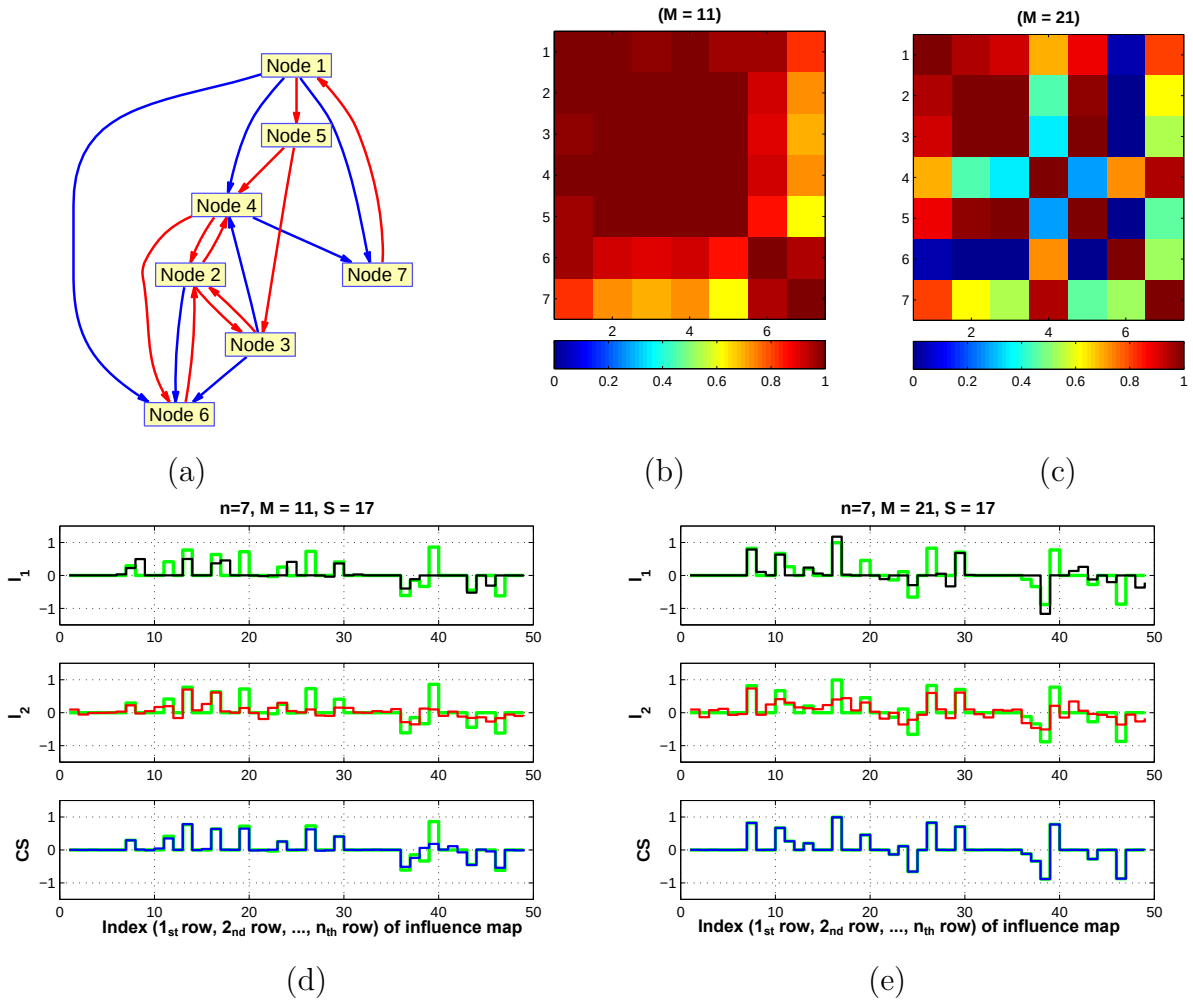


Figure 5.5: (Example 3) By reducing coherence, we can get better performance (7 states, 17-sparsity ($17 \ll 49$), 11/21 time points: (a) randomly generated graph (red:activation, blue:inhibition), (b) coherence of sensing matrix $\Omega(\mathcal{X})$ $M = 11$, (c) coherence of sensing matrix $M = 21$, (d) recovery of graph structure $M = 11$, (e) recovery of graph structure $M = 21$ where the green line represents the true structure in (d) and (e).

tically, $m \geq 4s$) [24]. In the following example, we illustrate that the incoherence of the sensing matrix can be used as not only a good metric to guarantee exact recovery but also a guideline for designing new experiments.

Example 3.(*Increasing performance by forcing incoherence on the sensing matrix*): Figure 5.5 shows the result for a 17-sparse network with 7 states. The comparison of coherences of the sensing matrices is shown in Figure 5.5 (b) and (c), and the reconstructed results are presented in (d) and (e) respectively. Since data itself has high correlation in Figure 5.5(b),

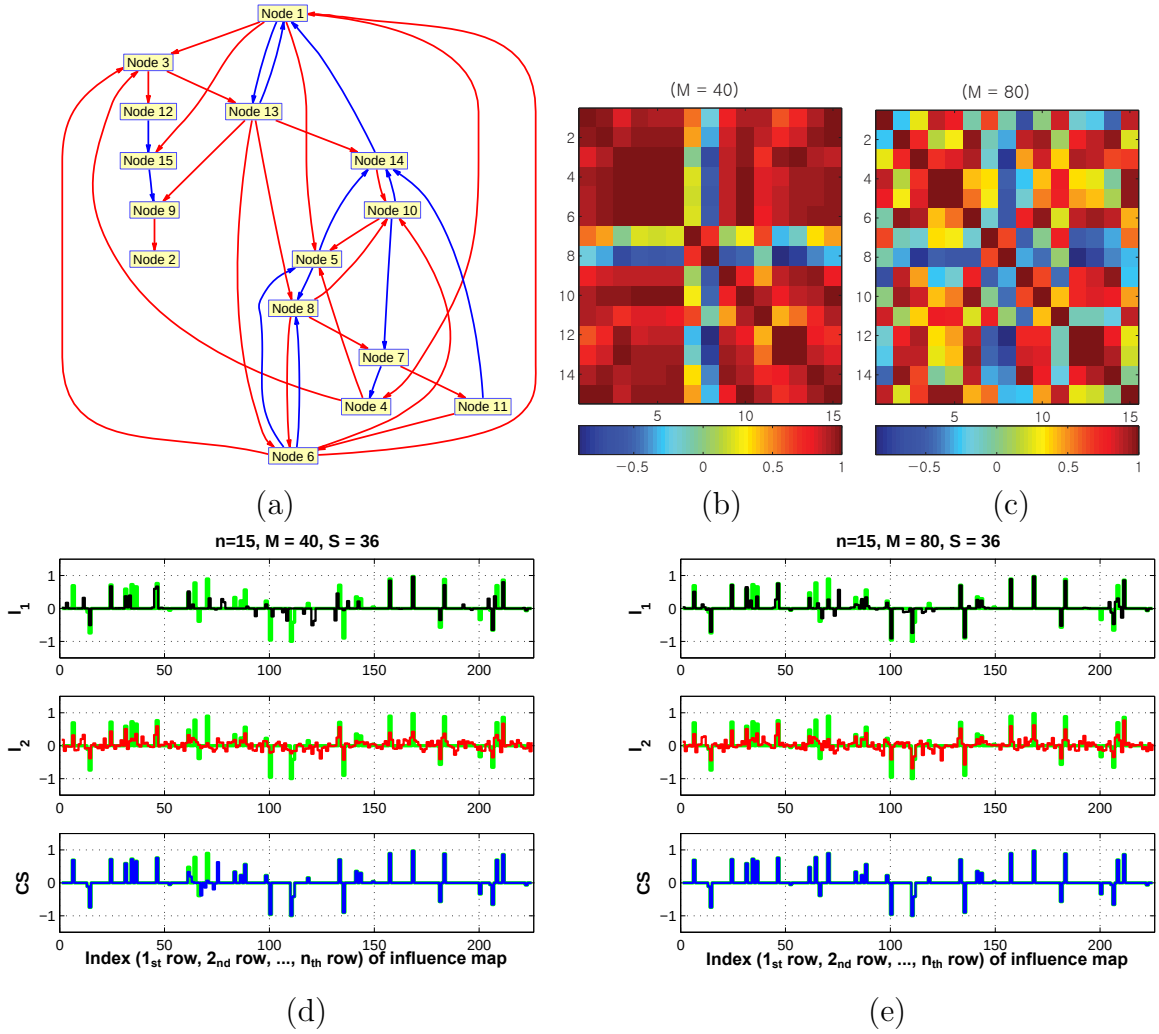


Figure 5.6: (Example 3') By reducing coherence, we can get better performance (15 states, 36-sparsity ($36 \ll 225$), 40/80 time points: (a) randomly generated graph (red:activation, blue:inhibition), (b) coherence of sensing matrix $\Omega(\mathcal{X})$ $M = 40$, (c) coherence of sensing matrix $M = 80$ (d) recovery of graph structure $M = 40$, (e) recovery of graph structure $M = 80$ where the green line represents the true structure in (d) and (e).

the reconstruction result shows some errors in (d). However, for $M = 21$ (which means adding new data points), coherence is reduced in Figure 5.5 (c) and the exact recovery is presented in (e).

Similarly, Figure 5.6 shows the result for a large network which has 36 edges with 15 states. We can compare the coherences of the sensing matrices in Figure 5.6 (b) and (c) and the recovered results in (d) and (e) respectively. The entries of the sensing matrix in (c) are more uniformly distributed in magnitude than those of (b). Also, only 80 measurements are

needed to reconstruct the graph structure exactly (instead of 225).

Numerical examples show that if the sensing matrix is incoherent enough, the reconstruction of the exact graph structure is almost always guaranteed. Here, coherence was decreased by adding more data points, but in practice, we could design new experiments to add more data points. This idea can be helpful for designing new experiments. For example, if time series data sets show high correlation between x_i and x_j or the sensing matrix has high coherence, new experiments can be designed which inhibit or stimulate either x_i or x_j. Then, by stacking all measurements as shown in equation (5.10), coherence between x_i and x_j can be reduced. On the other hand, if we do experiments which do not affect x_i and x_j, new experimental data set would not help to recover the graph structure because there would still exist a high coherence between x_i and x_j. Therefore, the coherence metric can not only guarantee exact graph structure recovery, but also provide a guidance for new experiments in a smart way.

Moreover, as the number of states (n) increases, the CS method is more efficient in reconstructing the graph structure because we estimate an s-sparse signal among n^2 elements and need m measurements ($n^2 \gg m \geq 4s$, if n is big enough). For example, if $n = 30$ and $s = 80$, we need at most $4s = 320$ measurements instead of $n^2 = 900$.

5.4.2 Linear Time Varying Case

As we mentioned above, the proposed method can handle a time-varying influence map which has nonlinear terms $\xi(\cdot), \phi(\cdot)$ in equation (5.7). The graph structure is randomly generated in Figure 5.7 (a) including nonlinear terms as shown in equation (5.18) in which there are 6 linear terms and 8 nonlinear terms among 6 states and an 18 time points data set is used for reconstruction. After considering nonlinear terms in our dictionary, the proposed method is applied. Figure 5.7 (b) shows time series data and (c) shows the exact reconstruction of the graph structure (there are 72 elements; the first 36 elements are related to linear terms and the other 36 elements are related to nonlinear terms). In general, for the nonlinear case as shown in equation (5.18), the influence map or strength of the connection is time-varying so nonlinear terms should be included in dictionary. Also, in order to guarantee exact recovery, the coherence condition should be handled carefully because this sensing matrix is more likely to be coherent.

$$\frac{\Delta X}{\Delta T} = \begin{bmatrix} 0 & 0 & -0.136 & 0 & 0.464 - 0.35x_1 & 0 \\ 0 & 0 & 0 & 0 & 0.13x_2 & 0 \\ 0 & 0.44x_3 & 0 & 0 & 0 & 0 \\ -0.4x_4 & 0 & 0 & 0 & -0.49x_4 & 0 \\ 0 & 0 & 0 & 0.662 & 0 & -0.17x_5 \\ 0 & 0.105 & 0.594 - 0.42x_6 & 0.333 & -0.42x_6 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = A^k X^k \tag{5.18}$$

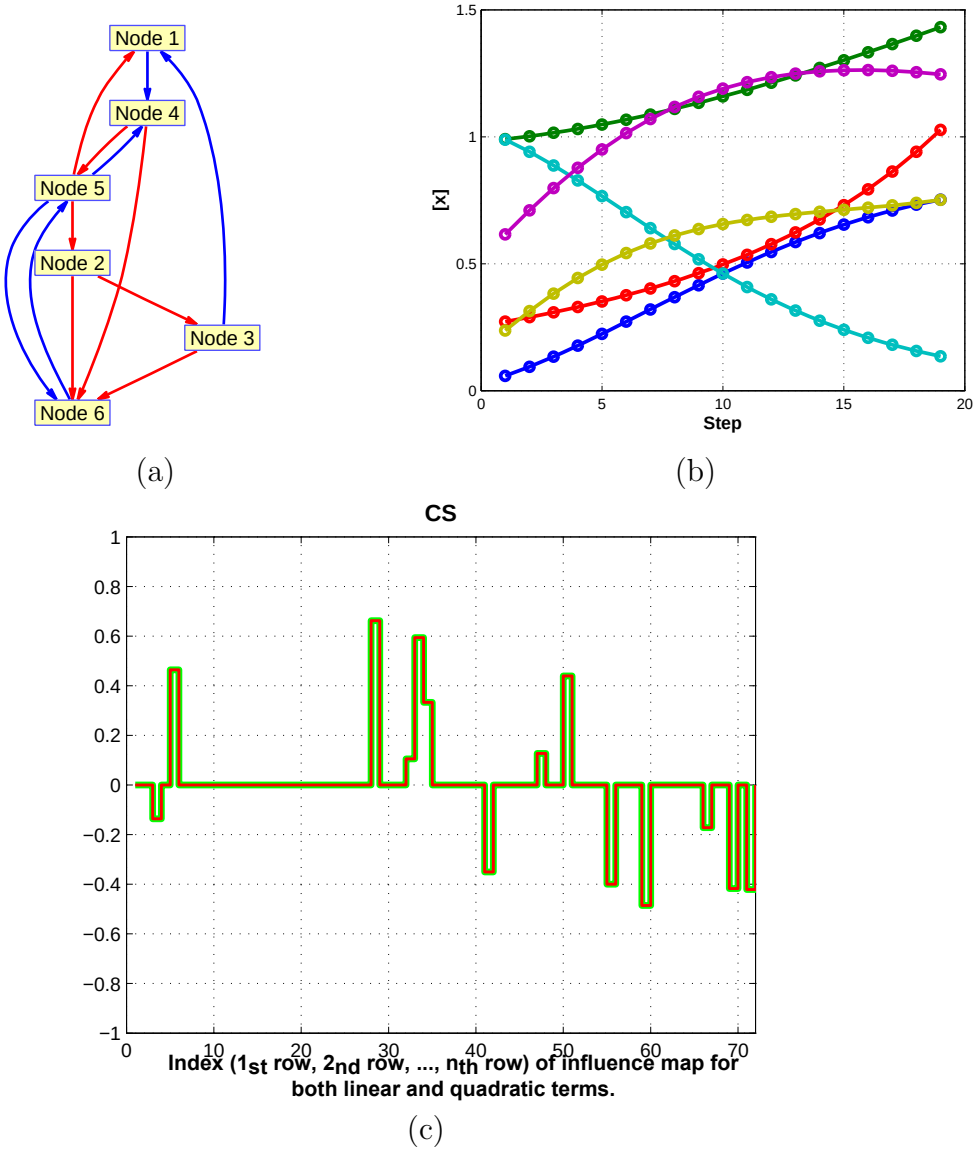


Figure 5.7: Time Varying case (6 states, 18 time points, (6+8)-Sparsity ($14 \ll 72$ (36 linear+ 36 nonlinear)): (a) randomly generated graph (red:activation, blue:inhibition), (b) experimental data (concentration), (c) recovery of graph structure where the green line represents the true structure.

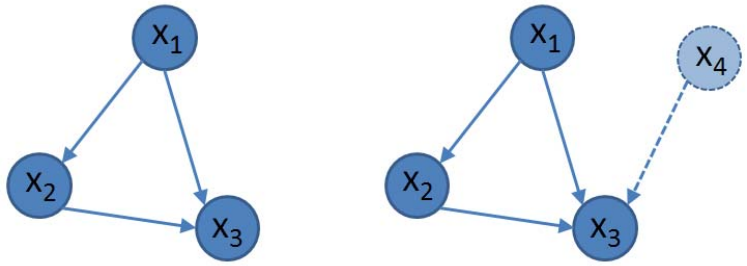
5.5 Graph Reconstruction with Hidden Nodes

One of the main contributions of the proposed method in the previous section is the conversion of the problem of inferring graph structure into the CS framework and the demonstration that one could recover sparse graph structures from only a few measurements. However, for

practical use, the proposed method needs to be able to deal with both sparsely corrupted signals and measurement noise. In general, the assumption of accessibility or observability of all nodes [35] is not satisfied; this is especially true in biological systems. For example, consider the following set of biochemical reactions: $x_1 \rightarrow x_2 \dashv x_3$, i.e., x_1 induces a direct positive regulation (activation) on x_2, x_2 induces a direct negative regulation (inhibition) on x_3. Now assume that x_2 is a hidden node, and we would like to infer the relationship between x_1 and x_3. From the above regulation relationship, we know that x_1 induces an indirect negative regulation on x_3 (through the hidden node x_2). In this case, even though x_2 is not observable, the indirect negative regulation between x_1 and x_3 can be inferred. A more interesting and challenging example is one in which an unknown protein (i.e., x_4) affects an observable protein (i.e., x_3) directly, as shown in Figure 5.8.

In this study, we focus on the case in which the hidden node affects observable nodes directly as shown in Figure 5.8 (b). Also, without loss of generality, this hidden node dynamics could be any arbitrary dynamic models; for example, the number of the hidden nodes can be larger than the number of observable states. Or, even if there is no hidden node, a small portion of the biological experiment dataset could be contaminated by large error in practice, for example, mislabeling, or improper use of markers or antibody. Moreover, all biological datasets are contaminated by at least a small amount of noise from measurement devices in general. Therefore, the proposed method should be robust.

Here, the question is whether it is still possible to reconstruct the graph structure reliably when measurements are corrupted. Since hidden nodes and measurement noise are considered, the number of time points is assumed to be greater than that of the previous case [35] (i.e., no corruption and no measurement noise). Thus, the number of rows of the sensing matrix is assumed to be greater than the number of columns. In [25], two decoding strategies for recovering $\bar{\beta} \in \mathbb{R}^n$ from a measurement $\mathcal{T} = \Omega\bar{\beta} + e + z$ is introduced by Candès and Randall, where $\Omega \in \mathbb{R}^{m \times n} (m \geq n)$, e is a possible sparse vector of large errors and z is a vector of small error affecting all the entries. It is shown that two decoding schemes allow the recovery of $\bar{\beta}$ with nearly the same accuracy as if no sparse large errors occurred. Our contribution is converting the problem of inferring the graph structure with hidden nodes



(a) (b)

Figure 5.8: (a) Simple graph structure without a hidden node (b) and with a hidden node x_4.

into the highly robust error correction method framework [25], in order to recover the sparse graph structure from measurement data corrupted by hidden node dynamics.

5.5.1 Sparse corruption

Consider the graph structure shown in Figure 5.8 (right) with a hidden node (i.e., x_4) and no measurement noise. Suppose the system dynamics are as follows (for simplicity, we consider a linear time invariant case in this section):

$$y(k) = \frac{1}{\Delta T}\begin{bmatrix}\Delta x_1(k)\\ \Delta x_2(k)\\ \Delta x_3(k)\\ \Delta x_4(k)\end{bmatrix} = \begin{bmatrix}A_{o,o} & A_{o,uo}\\ A_{uo,o} & A_{uo,uo}\end{bmatrix}\begin{bmatrix}x_1(k)\\ x_2(k)\\ x_3(k)\\ x_4(k)\end{bmatrix} = \begin{bmatrix}A_{o,o} & A_{o,uo}\\ A_{uo,o} & A_{uo,uo}\end{bmatrix}\begin{bmatrix}X_o(k)\\ X_{uo}(k)\end{bmatrix} \quad (5.19)$$

where $X_o(k) = \begin{bmatrix}x_1(k) & x_2(k) & x_3(k)\end{bmatrix}^T$ represents observable states and $X_{uo}(k) = x_4(k)$ is an unobservable or hidden node. Also, an influence map can be decomposed into 4 categories: $A_{o,o}, A_{o,uo}, A_{uo,o}$ and $A_{uo,uo}$. For example, $A_{o,o}$ represents the influence map from observable states to observable states and $A_{o,uo}$ shows the influence map in which hidden states can affect observable states' dynamics, etc.. As per our assumption, the influence of unobservable on observable nodes is assumed to be sparse (i.e., $A_{o,uo}$ is sparse). The observable node's dynamics can be denoted by

$$y_o(k) = A_{o,o}\begin{bmatrix}x_1(k)\\ x_2(k)\\ x_3(k)\end{bmatrix} + A_{o,uo}x_4(k) = A_{o,o}X_o(k) + u_k \quad (5.20)$$

where $y_o(k)$ are observable states from $y(k)$, $A_{o,o} \in \mathbb{R}^{n\times n}$ represents the influence map from the observable states, n is the number of observable states and the influence map from hidden nodes ($A_{o,uo}x_4(k)$) can be denoted by input $u_k \in \mathbb{R}^n$ which is assumed to be sparse and unknown. In other words, hidden nodes can affect only a few nodes' dynamics (intuitively, if hidden nodes affect all nodes, there is no way to reconstruct the graph structure). Therefore, in order to satisfy this assumption, we need some knowledge of the graph structure. Roughly speaking, our methodology breaks down large-scale networks into small networks and then infers the network structure.

Our sensing matrix can be constructed by using equation (5.9):

$$\mathcal{Y}_k = \Psi^k y_o(k) = \mathcal{X}_k\bar{\beta} + \Psi^k u_k \quad (5.21)$$

where $\Psi^k \in \mathbb{R}^{n\times n}$ is a randomly chosen orthonormal matrix with i.i.d. random variables, $\mathcal{X}_k$ was defined in the section 5.3 and $\bar{\beta}$ represents the unknown graph structure. Also, M time point measurements can be stacked as follows:

$$Y = \begin{bmatrix}\mathcal{Y}_1\\ \mathcal{Y}_2\\ ...\\ \mathcal{Y}_M\end{bmatrix} = \Omega\bar{\beta} + diag\{\Psi^1, \Psi^2, ..., \Psi^M\}\bar{u} = \Omega\bar{\beta} + \tilde{\Psi}\bar{u} = \Omega\bar{\beta} + e \quad (5.22)$$

where $Y \in \mathbb{R}^{M \cdot n}$, $\Omega \in \mathbb{R}^{M \cdot n \times n^2}$ and $\tilde{\Psi} \in \mathbb{R}^{M \cdot n \times M \cdot n}$ are known, $\bar{\beta} \in \mathbb{R}^{n^2 \times 1}$ represents the sparse graph structure and $\bar{u} = \begin{bmatrix} u_1^T & u_2^T & ... & u_M^T \end{bmatrix}^T \in \mathbb{R}^{M \cdot n \times 1}$ is a sparse vector with possibly large errors from hidden nodes. Without loss of generality, $\bar{u}$ can also represent arbitrary corruptions by human error during biological experiment.

5.5.2 Two-step refinements for sparse large corruption

Consider the recovery of the graph structure $\bar{\beta}$ from the corrupted vector Y. Suppose the number of rows of the sensing matrix Ω is greater than the number of columns (i.e., $M > n$, we need more time points if hidden nodes may exist) and consider a matrix Q^* which annihilates the sensing matrix Ω on the left ($Q^*\Omega = 0$) where Q^* is any $(M \cdot n - n^2) \times N \cdot n$ matrix whose kernel is the range of Ω in $\mathbb{R}^{N \cdot n}$ ($rank(Q^*) + nullity(Q^*) = N \cdot n$):

$$\tilde{Y} = Q^*Y = \cancelto{0}{Q^*\Omega} \cdot \bar{\beta} + Q^*e = Q^*e$$

Then, the following two-step optimization problem enables us to reconstruct graph structure exactly.

1) **Step 1**: the corrupted signal or unmodelled dynamics is filtered out from the measurement:

$$u^{\#} = \arg\min |\bar{u}|_{l_1} \quad \text{s.t.} \quad \tilde{Y} = Q^*e = Q^*\tilde{\Psi}\bar{u} = \tilde{Q}\bar{u} \tag{5.23}$$

If we could somehow get an accurate estimate $\hat{e} = \tilde{\Psi}u^{\#}$ from equation (5.23), the following equation represents the problem of reconstructing the graph structure $\bar{\beta}$.

2) **Step 2**: a sparsely connected graph structure is inferred using CS [35]:

$$\min |\bar{\beta}|_{l_1} \quad \text{s.t.} \quad Y - \hat{e} = \Omega\bar{\beta} \tag{5.24}$$

We assume all nodes are accessible for measurement (meaning that there is no hidden node, $e = 0$) [35], so we can solve **Step2** directly without **Step1** since $Y = \Omega\bar{\beta}$. Also, the intuition is that $Y(= \Omega\bar{\beta} + e)$ can be decomposed as the superposition of an arbitrary element in $V(= Y - QQ^*e)$ and of an element in $V^{\perp}(= QQ^*e)$ as shown in Figure 5.9. In other words, $\mathcal{P}_{V^{\perp}}(Y) = QQ^*$ where $\mathcal{P}_{V^{\perp}}$ is the orthonormal projector onto $V^{\perp}$:

- $\tilde{Y} = 0$: there is no hidden node $\implies Y \in \mathcal{R}(\Omega)$
- $\tilde{Y} \neq 0$: Y cannot be represented by $\Omega\bar{\beta}$ so there might be hidden nodes or our dictionaries in the sensing matrix Ω are not sufficient to represent Y (revealing deficiencies in our model).

The two step convex optimization problems (5.23) and (5.24) are l_1-norm optimization problem in CS. Thus, if the sensing matrices $\tilde{Q}$ and Ω satisfy the incoherence condition, signals $\bar{u}$ and $\bar{\beta}$ can be recovered exactly [35][24]. Here, there are many possible choices of Q^* but we

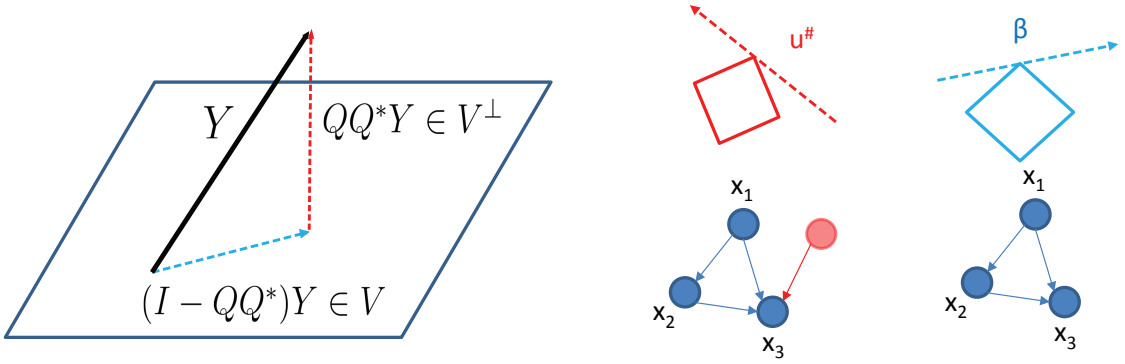


Figure 5.9: Geometric view of two consecutive l_1-norm optimization programmings.

have to choose Q^* to satisfy the coherence condition for the exact recovery of $\bar{u}$ [35]. To choose such a Q^*, observe that Ω can be denoted as follows using Singular Value Decomposition (SVD):

$$\Omega = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma V_1^T \tag{5.25}$$

where $U_1 \in \mathbb{R}^{N \cdot n \times n^2}, U_2 \in \mathbb{R}^{N \cdot n \times (N \cdot n - n^2)}, \Sigma \in \mathbb{R}^{n^2 \times n^2}, V_1 \in \mathbb{R}^{n^2 \times n^2}$ and $V_2 \in \mathbb{R}^{n^2 \times (N \cdot n - n^2)}$. Suppose we choose Q^* such that $Q^* = WU_2^T$. Then:

$$\begin{aligned} Q^*Y &= Q^*\Omega\bar{\beta} + Q^*e \\ (WU_2^T)Y &= (WU_2^T)U_1\Sigma V_1^T\bar{\beta} + (WU_2^T)e \\ WU_2^TY &= WU_2^T\tilde{\Psi}\bar{u} \end{aligned}$$

where $W \in \mathbb{R}^{(N \cdot n - n^2) \times (N \cdot n - n^2)}$ can be used as a tuning matrix for satisfying the incoherence condition.

5.5.3 Sparse corruption with measurement noise

While considering influence from hidden nodes is interesting, it still may not be realistic to assume that except for hidden nodes, one is able to measure the node dynamics with infinite precision. A better model would assume that there is measurement noise. Consider the problem of recovering the graph structure $\bar{\beta}$ from the vector Y which is corrupted by measurement noise n:

$$Y = \Omega\bar{\beta} + z \tag{5.26}$$

where $z = e + n = \tilde{\Psi}\bar{u} + n$, n is a Gaussian noise $\mathcal{N}(0, \sigma)$ assumed to be bounded $||n||_{l_2} \leq \epsilon$. In general, we can consider any corruption decomposed into sparse large error $\bar{u}$ and small

magnitude error n. Then, modified two-step refinements can be applied as follows:

$$\tilde{Y} = Q^*\Omega\bar{\beta} + Q^*z = Q^*e + Q^*n$$
$$u^{\#} = arg\min |\bar{u}|_{l_1} \quad \text{s.t} \quad ||\tilde{Y} - Q^*\tilde{\Psi}\bar{u}||_{l_2} \leq \epsilon_1$$
$$\beta^{\#} = arg\min |\bar{\beta}|_{l_1} \quad \text{s.t.} \quad ||Y - \Omega\bar{\beta} - \hat{e}||_{l_2} \leq \epsilon_2, \hat{e} = \tilde{\Psi}u^{\#}$$

where the parameters ϵ_1, ϵ_2 above depend on the magnitude of the small errors ϵ, which can be determined as in [25].

5.5.4 Numerical Examples

Consider simple examples in which we have a linear time invariant influence map (we note this goes beyond the results in [70] due to the consideration of hidden node dynamics with measurement noise). For the LTI model, we generate a random graph structure whose elements are randomly chosen. Here, we either randomly pick a node whose dynamics is corrupted by a hidden node, or choose an arbitrary time point data in which data is contaminated by an arbitrarily large error. For the hidden node's dynamics, random signals with Gaussian-distributed random variables are used for representing highly nonlinear system dynamics and the amplitude is chosen to be similar to the magnitude of observable node dynamics $\dot{x}$ as shown in Figure 5.10(c) and Figure 5.11(c) (middle). Also, all the measurements are assumed to be contaminated by small measurement noise. Finally, the proposed method is applied for studying HER2 overexpressed breast cancer signaling pathway using an RPPA (Reverse Phase Protein Assay) dataset [81] where both linear and nonlinear quadratic terms are considered in the sensing matrix. The performance of CS is compared with the performances of the $||\cdot||_{l_1}$ and $||\cdot||_{l_2}$ norm optimizations.

Example 4.(*With measurement noise and arbitrary corruption by human error*): Figure 5.10 (a) shows a randomly generated graph structure which has 5 nodes and 10 edges and (b) represents the dataset $\dot{x}$, corruption signal u_i and noise n_i. As shown in Figure 5.10 (c), a small portion of the dataset is assumed to be largely corrupted by human error (u_i) and measurement noise (n_i) is assumed to be small (2% of the magnitude of the dynamics). Figure 5.10 (b) and (d) show the result of two-step refinements respectively. By estimating and filtering out $\hat{z}$ from Y, the graph structure can be reconstructed quite accurately as shown in Figure 5.10 (d) where the X-axis represents indices of the influence map (i.e., $1_{st}, 2_{nd}, ..., n_{th}$ rows of influence map).

Example 5.(*Corrupted by a hidden node with measurement noise*): a specific node is selected randomly and corrupted by a hidden node as shown in Figure 5.11 (i.e., node 4, $e_4 \approx 15\%$ of the magnitude of $\dot{x}$). Also, measurement noise is considered and assumed to be small (1% of the magnitude of the dynamics). Figure 5.11 (a) shows a randomly generated graph structure which has 5 nodes and 11 edges and (c) represents the dataset $\dot{x}$, corruption signal u_i and noise n_i. By estimating $\hat{z}$ from Y, the corrupted signal can be filtered out and reconstructed

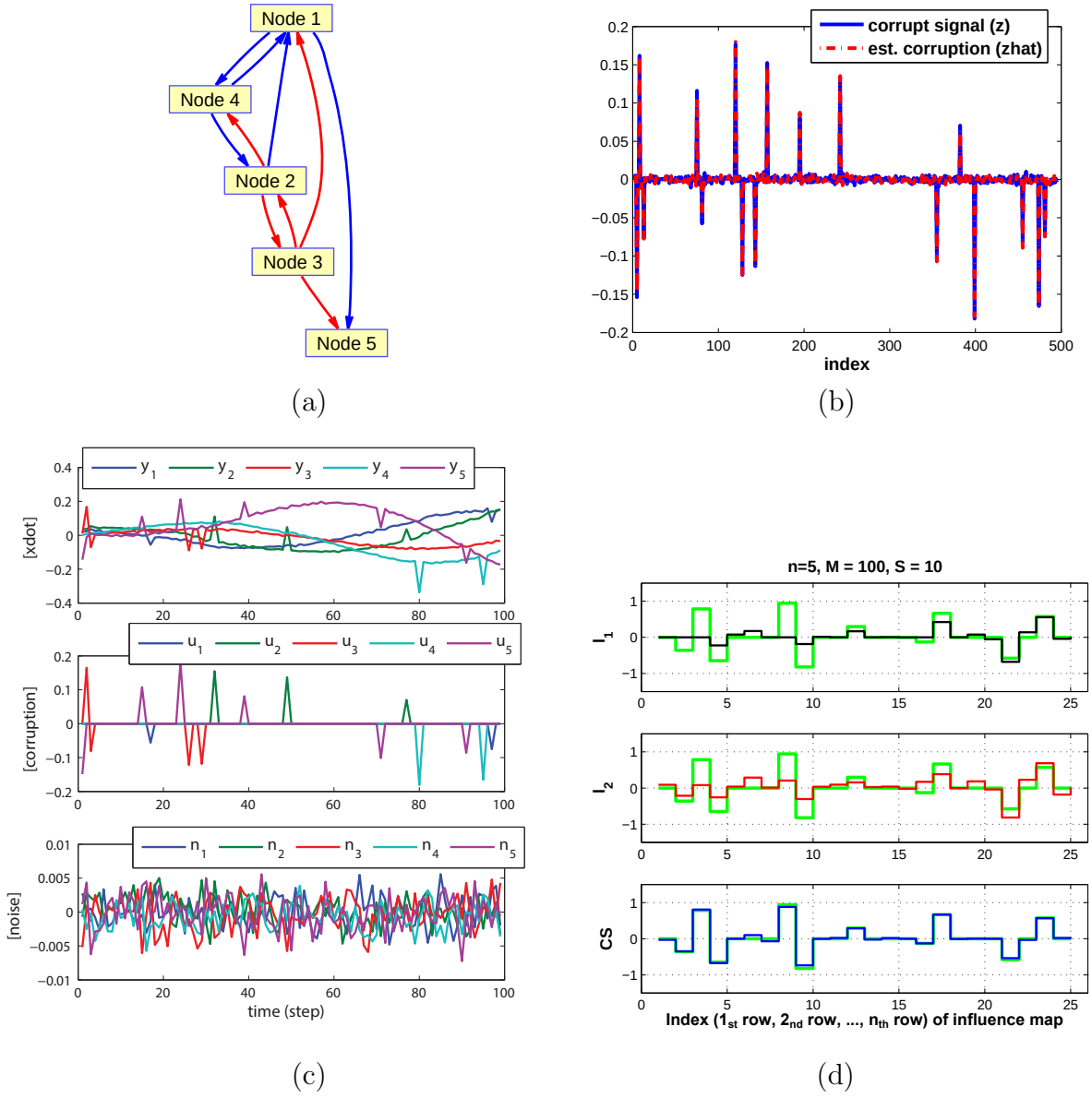


Figure 5.10: (Example 4) Time invariant case with measurement noise (5 states, 100 sample points, 10-Sparsity (a) randomly generated graph (red:activation, blue:inhibition), (b) recovery of corrupting signal $z = e + n$ where the magnitude of u, n are about 60%, 2% of the magnitude of $\dot{x}$ respectively, (c) experimental data $\dot{x}$, corrupting signal e and measurement noise n, (d) recovery of graph structure where the green line represents the true structure.

graph structure is shown in Figure 5.11 (d). In this case, only node 4 is corrupted by a hidden node so we can reconstruct the graph structure exactly, except the corrupted node (indices 16-20 in Figure 5.11(d)).

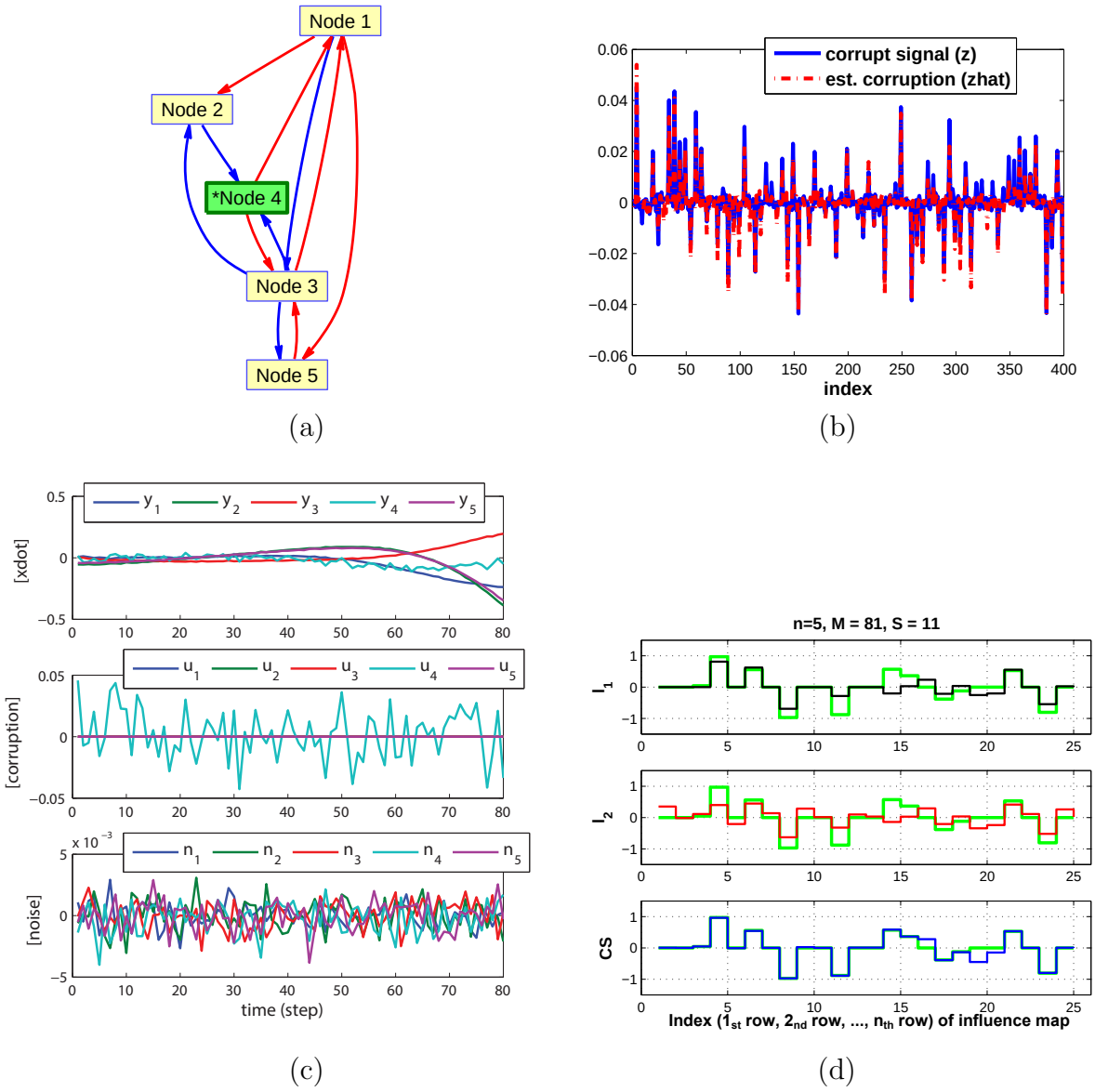


Figure 5.11: (Example 5) Time invariant case with measurement noise (5 states, 80 sample points, 11-Sparsity (a) randomly generated graph (red:activation, blue:inhibition, Node 2 is corrupted by hidden node dynamics) (b) recovery of corrupting signal $z = e + n$ where the magnitude of u, n are about 15%, 1% of the magnitude of $\dot{x}$ respectively, (c) experimental data $\dot{x}$, corrupting signal e and measurement noise n, (d) recovery of graph structure where the green line represents the true structure.

Example 6. *(HER2 Overexpressed breast cancer)*: we apply the proposed algorithm to study a breast cancer signaling pathway by reconstructing the graph structure using an RPPA

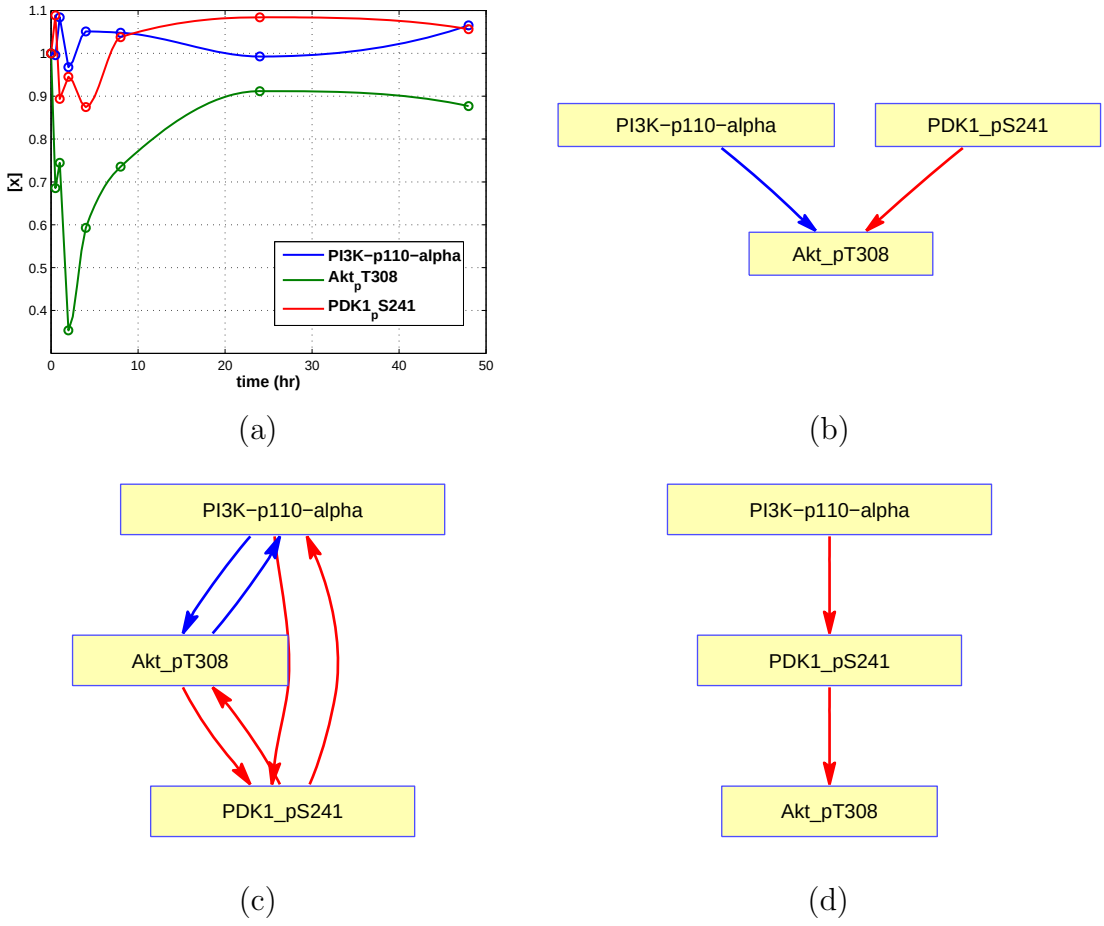


Figure 5.12: (Example 6: biological system) RPPA dataset (SKBR3 cell line, Serum [81]) (a) gene expression date [0-48hr] (b) l_1 optimization result (c) l_2 optimization result (d) CS reconstruction result.

dataset [81]. Here, the simply connected part is chosen from the large scale network (3 nodes, known to be simply connected, i.e., PI3K $\rightarrow$ PDK $\rightarrow$ Akt and PDK $\rightarrow$ Akt $\rightarrow$ mTOR) in order to satisfy our assumption that the influence on observable nodes from a hidden node should be sparse. For the simply connected case, the CS reconstruction result matches the true structure exactly while l_1- and l_2-optimizations fail to reconstruct the known structure as shown in Figure 5.12 and Figure 5.13.

Also, an abstract model of breast cancer signal pathway proposed by Dr. Moasser is considered as shown in Figure 5.14 (b) where PHLPP isoforms are a pair of protein phosphatases, PHLPP1 and PHLPP2, which are important regulators of Akt serine-threonine kinases (Akt1, Akt2, Akt3) and conventional protein kinase C (PKC) isoforms. PHLPP may act as a tumor suppressor in several types of cancer due to its ability to block growth factor-induced signaling in cancer cells [20]. PHLPP dephosphorylates Ser473 (the hydrophobic motif) in Akt,

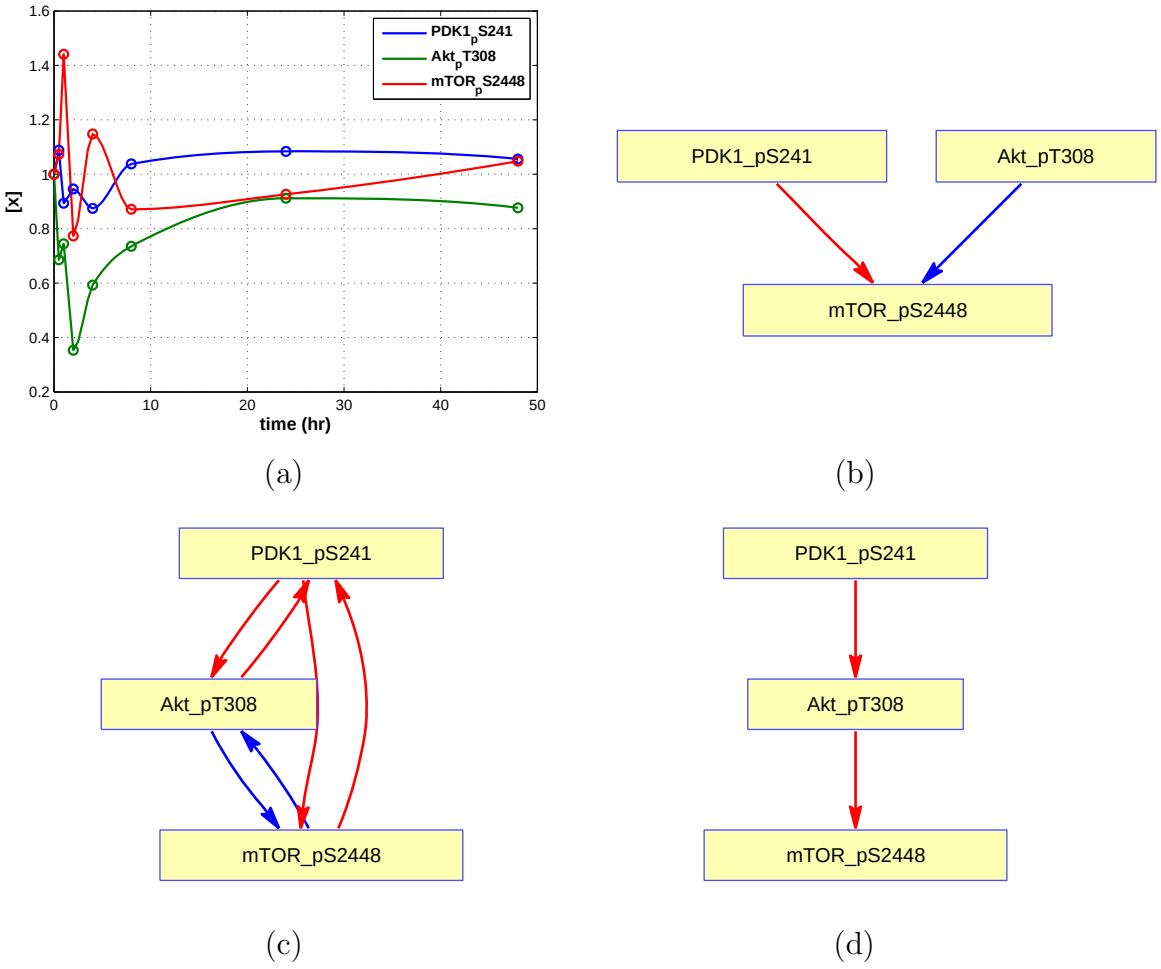
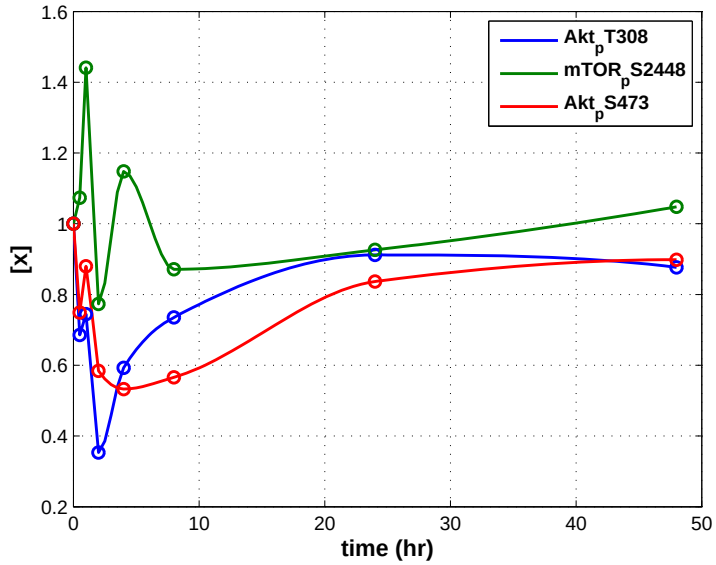


Figure 5.13: (Example 6: biological system) RPPA dataset (SKBR3 cell line, Serum [81]) (a) gene expression date [0-48hr] (b) l_1 optimization result (c) l_2 optimization result (d) CS reconstruction result.

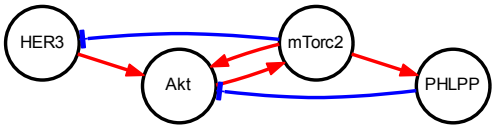
thus partially inactivating the kinase [62]. Unfortunately, in our RPPA dataset, we do not have PHLPP so we simply consider three nodes ($\mathsf{Akt}_{\mathsf{pT308}}$, $\mathsf{Akt}_{\mathsf{pS473}}$ and mTOR). Figure 5.14 (c) shows the result of proposed method using RPPA dataset and the reconstructed graph structure matches up to known structure. Especially, our result can capture the partial inactivating characteristics of PHLPP (i.e., $\mathsf{mTOR}(\rightarrow \mathsf{PHLPP}) \dashv \mathsf{Akt}_{\mathsf{pS473}}$).

5.6 Conclusion

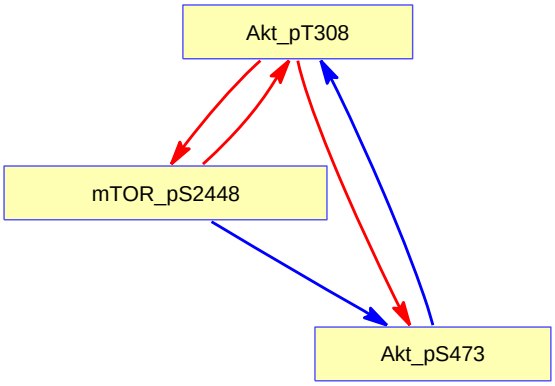
The proposed methods for reconstructing sparse graph structures based on time series gene expression data without any *a priori* information is useful and performs quite well. First, the



(a)



(b)



(c)

Figure 5.14: (Example 6: biological system) RPPA dataset (SKBR3 cell line, Serum [81]) (a) gene expression date [0-48hr] (b) abstract model by Dr. Moasser (c) *CS* reconstruction result.

time invariant influence map is presented and the importance of incoherence is discussed. Also, we illustrate that incoherence in the sensing matrix can be used as a guideline for designing effective experiments. The proposed method is also applied to the time-varying (nonlinear) case and the results are discussed.

Second, the proposed method is extended to the cases in which the dynamics is corrupted by hidden nodes and the measurement is corrupted by human error in addition to

measurement noise. Using a two-step refinement procedure, we demonstrate good performance for the reconstruction of graph structure. A set of numerical examples is implemented to illustrate the method and its performance. Also, a simple biological example of HER2 overexpressed breast cancer using an RPPA dataset is studied. We are currently applying our method to recover HER2 signaling pathway, where a significant part of the network is currently unknown.

Chapter 6

Mode Identification based on Neural Activity (I)(II)

Introduction

Decoding the map between neural spike signals in the brain and movements of the arm is of great interest in the area of neuroscience. The main goal is to advance our understanding of fundamental principles in the neural control of movements. Ultimately, such a decoder can be used as a brain-machine interface (BMI), allowing paralyzed patients, for example, to control an exoskeleton attached to their arms. Many brain-machine interface (BMI) researchers have demonstrated the feasibility of using (adaptive) input-output models for reconstructing hand trajectories [30][48][112]. For example, some recent research has focused on the relationship between kinematic parameters such as reach direction and speed, and measured response of motor cortex neurons. In this view, neural response encodes movement parameters in a straightforward manner such as the spatial tuning of cortical response for reach direction [65][64]. All of the current proposed models of neural encoding basically demonstrate the ability to encode and store the relationships between neural firing rates and hand trajectories. In order to design a decoder, model parameters are adjusted to minimize the error between model output and actual hand movements based on a statistical criterion such as mean square error.

Although the individual neural response per neuron shows temporal properties, until recently most studies have focused on the spatial, rather than temporal structure of the neural response. Several recent studies [41][42][165] focused on the temporal complexity and heterogeneity of single-neuron activity in the premotor and motor cortices. In [41], Churchland *et al.* showed that neural activity patterns in the primary motor cortex and premotor cortex associated with nearly identical velocity profiles can be very different. Yu *et al.* considered the problem of extracting smooth, low-dimensional neural trajectories that summarize the activity recorded simultaneously from many neurons during individual experimental trials [165]. They proposed a novel method for extracting neural trajectories

– Gaussian-process factor analysis (GPFA) – which unifies smoothing and dimensionality-reduction operations in a common probabilistic framework.

Much of the motor control literature has dealt with the possibility that movements may be programmed via combinations of discrete motor primitives [93][21][121][63][117][115][113]. Several studies suggested forms of motor primitives which are united in providing the nervous system with a putative mechanism for reducing both the number of degrees of freedom it has to control, and the frequency at which this control needs to be executed. One example of a motor primitive is a submovement, which is the focus of this chapter. In [71], the hand speed profile as a function of time resulting from arm movements is a sum of bell-shaped functions, each of which called "submovement". The speed profile of a simple, well-controlled movement looks like a single bell-shape. The speed profile of a more complex movement looks like a sum of several bell-shapes. This suggests that the control of hand position may involve discrete mode transitions in the brain. Neural spike signals from the brain may encode information about the occurrence of these mode transitions, which could signify the onset of submovements.

For BMI, movements need to be predicted in real time. This motivated the use of a hybrid systems approach, where movement prediction occurs whenever discrete mode transition is detected. Determining the discrete modes of operation of the brain would be a big step towards a hybrid systems approach to decoding the map between neural activity and hand kinematics. The purpose of this chapter is to present a new spatiotemporal perspective in analyzing neural activity data, which consists of the spike rates of a group of neurons as a function of time. The spike rate of neuron, or unit, is obtained by counting its action potentials within a time bin and dividing by the duration of that bin. The data are collected from a lab monkey which is instructed to repeatedly perform a simple task. Rows of the data matrix will represent the spatial dimension and columns will present the temporal dimension.

In Part I, we use unsupervised segmentation of neural activity via Sparse Subspace Clustering (SSC) to identify the mode. Once we interpret and extract the features of our systems such switching conditions, the number of modes and sequences of modes, then we could use a conventional system identification for a single LTI system. Also, we demonstrate that the monkey brain does in fact operate in various discrete modes when controlling arm movements.

In Part II, robust principal component analysis (RPCA) will be applied to the data matrix to extract temporal characteristics of neural activity by decomposing the matrix into a low-rank component and a sparse component. This allows us to identify the similarities in a spatial and temporal structures when the monkey performs similar tasks at different times. This differs from an approach like principal component analysis (PCA), which breaks down under large data corruption, and cannot extract temporal information [42]. In the context of neural activity, the low-rank matrix corresponds to the common features of neural activity across similar motor primitives, which are submovements in this study; the sparse matrix represents the uncommon component across submovements, and is interpreted as noise or as signal unrelated to submovement onset. Using RPCA, we test whether submovements are associated with neural activity and how well neural activity reflects the onset of submove-

ments across different task conditions. These findings open doors to many potential areas of research, and is a step in both understanding how the brain works and improving the quality of life of many neurological patients. The material in this chapter is based on the work in [38][115].

The rest of this section is organized as follows: Section 6.1 presents the main motivation for the hybrid system approach for brain model. In Section 6.2, we introduce sparse subspace clustering and show how we apply this algorithm to a neural dataset. Section 6.3 shows the results of applying SSC, with discussion. From Section 6.4, we consider a low-rank representation of neural activity for detecting submovement. In Section 6.4, we characterize the task-relevant pattern and introduce RPCA. Section 6.5 discusses disentanglement of the low-rank and sparse components and the necessity of random projection. In Section 6.6, the application of neural activities is presented. Finally, conclusions are given in Section 6.7.

I. Mode Identification of Neural Response via Sparse Subspace Clustering

6.1 Motivation

Hybrid systems are heterogeneous dynamics systems whose behaviors are determined by interacting continuous and discrete dynamics. The main advantage of hybrid models is that they can be used to approximate continuous phenomena by concatenating different models from a simple class using discrete variables. For example, a nonlinear dynamical system can be approximated by switching among various linear models.

Due to their many potential applications, hybrid systems have had great impact on many areas, such as air traffic control and systems biology, during the last decade [149][150][69]. However, most of the theoretical developments have used an assumption that a hybrid model is available. In some cases, it is possible to obtain the model starting from theoretical or physical knowledge, or intuition. On the other hand, it is often too complicated or even impossible to derive a theory driven model. For example, the brain might generate a combination or sequence of discrete, time-varying motor programs to effect motion. However, because the number of motor programs, the model parameters, the discrete state, and the transition condition are all unknown, the identification problem is challenging in that there is a coupling between the estimation of the model parameters and the segmentation of data into discrete states. For such systems, a model needs to be identified on the basis of experimental data first.

In this section, we assume a hybrid linear time-invariant (LTI) system and are concerned with offline data analysis for identification of a hybrid model of brain. There are various approaches to identify hybrid LTI systems: iterative algorithms which sequentially estimate the parameters of the model and classify the data, statistical clustering of measured data via a Gaussian mixture model, support vector machines and so on [119][123][109][79]. Here, we use unsupervised segmentation of neural activities via Sparse Subspace Clustering (SSC). Once we interpret and extract the features of our system, then we might use conventional identification for a single LTI system. Also, we show that SSC techniques strongly suggest that the brain does in fact operate in discrete modes.

6.2 Sparse Subspace Clustering

In some studies, the neural signals used for decoding movement have been reduced in dimensionality, for example, by locally linear embedding (LLE; [44]), or by a modified form of factor analysis known as Gaussian-process factor analysis (GPFA; [130] [165] [43]). Also, studies using neural activities to decode movement parameters frequently use a population vector algorithm (PVA), linear regression (Wiener) decoders [158] [157], or Kalman-based filters [103] [112].

We have attempted to classify discrete behavioral states from neural activities related to discrete motor program as shown in Figure 6.1. For example, motor cortical neurons convey some information about kinematic parameters or speed profiles, but except in the average of many neurons and trials, their firing is poorly correlated with speed profiles. Also, the brain might generate a combination or sequence of discrete, time-varying motor programs. If we can classify the sequence of different motor programs, this information can be used to infer the exact transitions or pick effective neural activities for both a discrete and continuous decoder. For example, for estimating the speed profile using a discrete decoder, we consider neural activities around the experimental "Go" cue, but sometimes actual movements occur with delay. Moreover, for actual application of the closed-loop BMI, we do not have a "Go" cue so we have to detect this triggering signal from the neural activity. Similarly, for the continuous decoder, there might exist a finite number of subsystems where switching or transitions might occur, so if we can cluster neural activities based on the corresponding

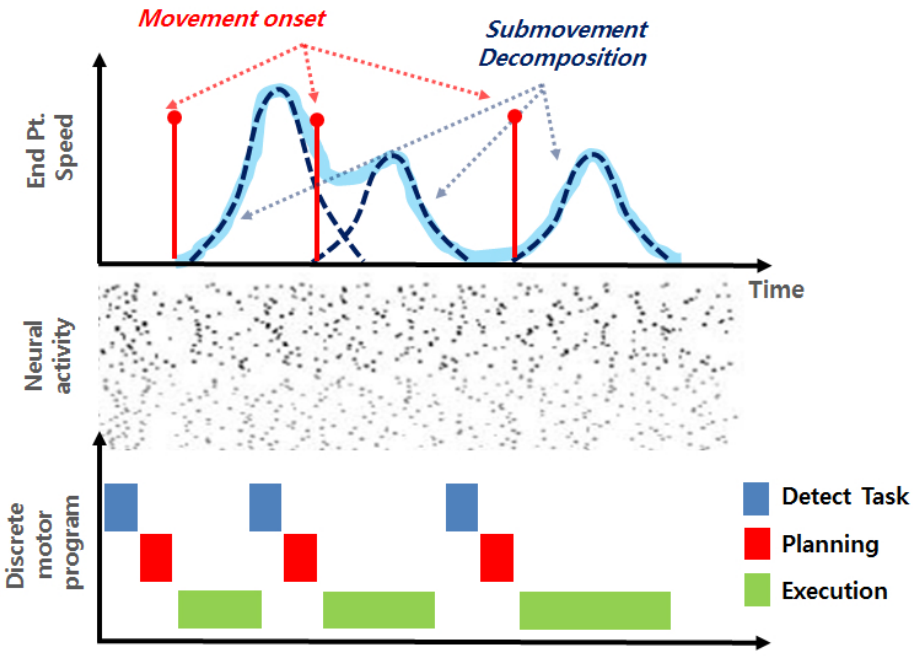


Figure 6.1: Conceptual diagram of discrete motor programs.

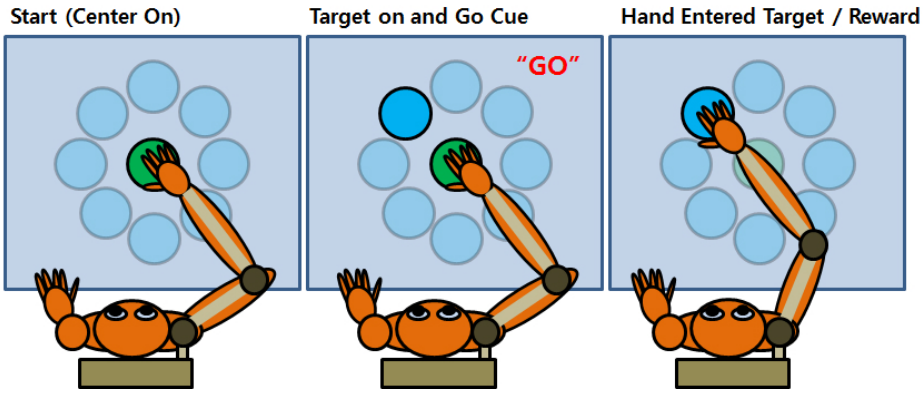


Figure 6.2: Center-out task in manual control.

subsystems, this might be helpful to better understand our system. In this study, we consider the problem of clustering a collection of unlabeled neural activities assumed to lie near a union of lower dimensional spaces. For example, each neural activity corresponding to a different motor program might lie on a different subspace.

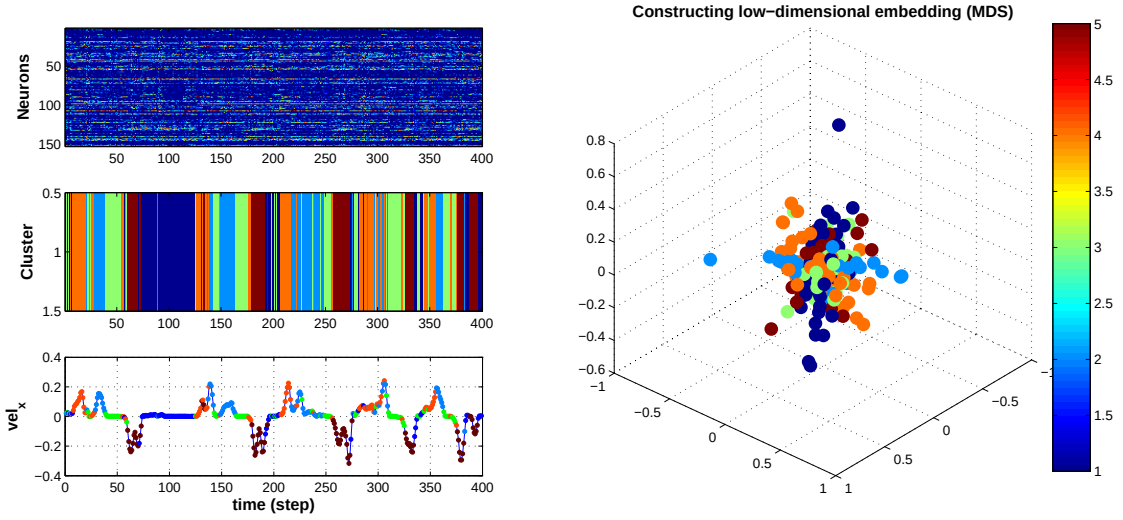
6.2.1 Experimental Setup

Center-out data involves a monkey performing center-out tasks, in which the monkey first places its hand at the center of a playing surface as shown in Figure 6.2 (left). Then, one of the eight targets around the center lights up, prompting the monkey to prepare to move its hand there. Finally, a "Go" cue signals the monkey to move its hand from the center location to the indicated target as shown in Figure 6.2 (middle, right).

All procedures were conducted in compliance with the National Institute of Health Guide for Care and Use of Laboratory Animals and were approved by the University of California, Berkeley Institutional Animal Care and Use Committee. While the monkey is either moving freely in one-dimension or performing center-out tasks, single unit activity (SUA) of approximately 150 neurons was recorded; this data will be referred to as "neural data". Roughly speaking, SUA is the rate at which a neuron fires signals called neural spikes. The two-dimensional hand position and velocity is also recorded; this data will be referred to as "kinematics data".

6.2.2 Method

In general, we do not know in advance how many subspaces there are nor any information about their dimensionality. Here, we apply sparse subspace clustering (SSC). Suppose the observed neural activities are the columns of a matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N] \in \mathbb{R}^{n \times N}$, where N is the total number of time points, n is the number of neurons and $\mathbf{u}_i$ represents neural activity at time i. In this study, we assume that we are given neural activities $\mathbf{u}_i \in \mathbb{R}^n$ over



(a) (b)

Figure 6.3: Results of SSC applied to neural data set. Each colored dot represents a different mode, so that at each time, the brain is in one of the specified number of modes. (a) The clustered group is highly correlated with kinematics (x-velocity)((a) bottom) and we construct low-dimensional embedding using classical multidimensional scaling (MDS) as shown in (b) [Carmena Lab data, paco, center-out task, 100ms bin].

time which are distributed on a union of unknown linear subspaces $S_1 \cup S_2 \cup ... \cup S_L$ where there are L subspaces of $\mathbb{R}^n$ of unknown dimensions $d_1, d_2, ..., d_L$. More precisely, we have a group $\mathcal{X} \subset \mathbb{R}^n$ consisting of N points in $\mathbb{R}^n$, which may be partitioned as follows:

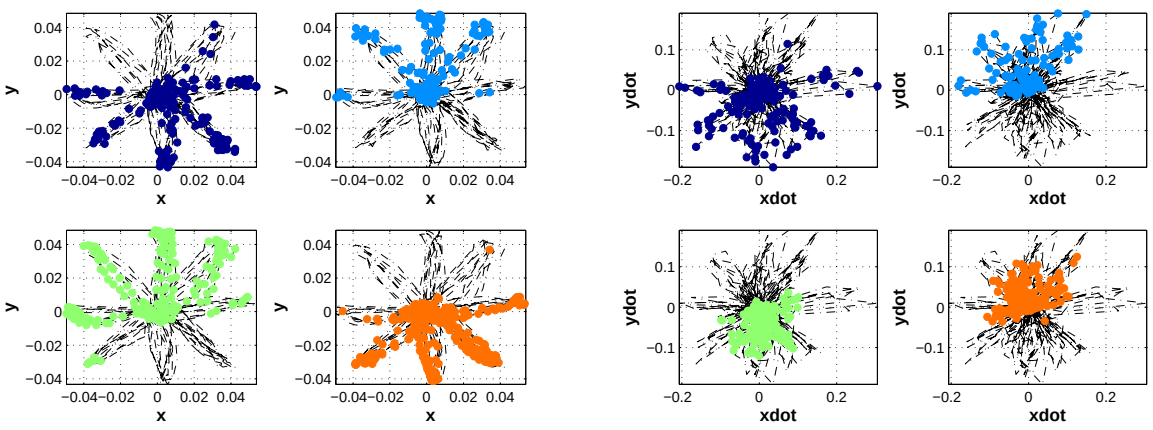
$$\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1 \cup ... \cup \mathcal{X}_L \tag{6.1}$$

where $\mathcal{X}_0$ accounts for possible outliers and for each $l \geq 1$, $\mathcal{X}_l$ is a collection of N_l unit-normed vectors chosen from S_l. A basic assumption is that neural activities for different motor programs might lie on different subspaces so if we could classify each neural activity into different groups, we could recover all the unknown subspaces or sequences of neural activities.

The key idea of the SSC algorithm [56] is to find the sparsest expansion of each column $\mathbf{u}_i$ of $\mathbf{U}$ as a linear combination of all the other columns. We expect that the sparsest representation of $\mathbf{u}_i$ would only select vectors from the subspace in which $\mathbf{u}_i$ happens to lie. This motivated Elhamifar and Vidal [56] to consider the following optimization problem:

$$\min_{z \in \mathbb{R}^N} ||z||_{l_1} \quad \text{subject to} \quad \mathbf{U}z = \mathbf{u}_i \quad \text{and} \quad z_i = 0 \tag{6.2}$$

where the idea is that whenever $z_j \neq 0$, $\mathbf{u}_i$ and $\mathbf{u}_j$ belong to the same subspace. Once we solve the optimization problem, we form the affinity graph G with nodes representing the N



(a) (b)

Figure 6.4: Results of SSC when applied to a portion center-out data (1000 time bins). Each colored dot represents a different mode. The colored dots are overlaid on top of 2D position and velocity plots (black lines) for visualization (a) $x - y$ (b) $\dot{x} - \dot{y}$.

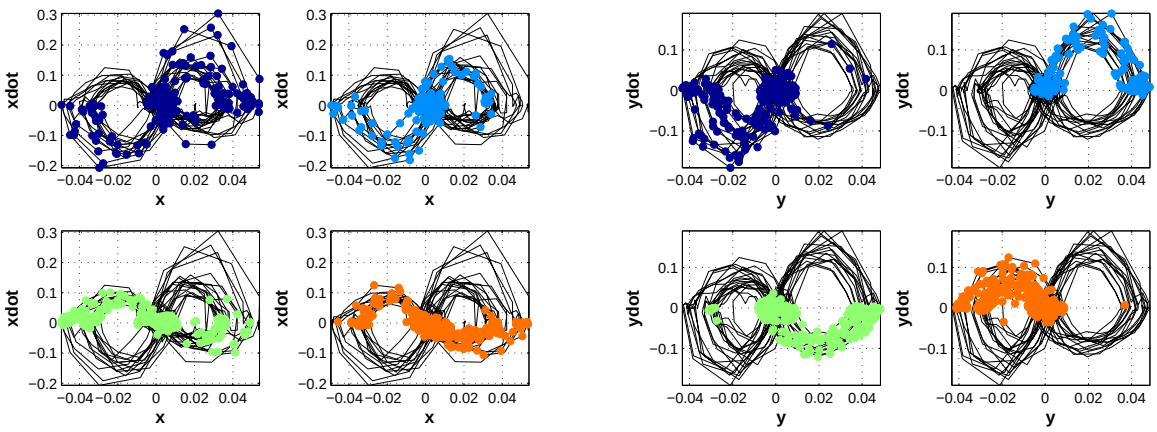
data points and sort the eigenvalues of the normalized Laplacian of G in descending order. Then, we apply spectral clustering techniques to the affinity graph. Figure 6.3 shows the sequence of clustered group by applying SSC where each dot is overlaid on top of kinematics data for visualization. Each clustered group shows high correlation with actual velocity profile (we color-coded into velocity profile to see correlation).

SSC is not limited to only being able to determine a certain number of modes. When the number of modes is increased, we again see the clustered group corresponding to the end point kinematics. For example, if five modes are chosen, we have the positive velocity (orange, light blue), negative velocity (dark red), and zero velocity (dark blue, light green). The negative velocity profile is highly correlated with the dark red group. The orange mode seems to correspond to the beginning of a series of movements and the dark red mode seems to correspond to the end of a series of movements. Transition between two modes may be indicative of the onset of submovements.

Also, we modified our input matrix including previous step to consider dynamics of neural activities as follows:

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_N \\ \mathbf{u}_0 & \mathbf{u}_1 & \dots & \mathbf{u}_{N-1} \end{bmatrix} \in \mathbb{R}^{2n \times N} \tag{6.3}$$

Then, the clustered result is shown in Figure 6.4 (a) and (b) show trajectories of position (x, y) and velocity $(\dot{x}, \dot{y})$ respectively. Here, we allow four subspaces group and we color the plot line with corresponding subspaces or cluster groups to interpret the result easily. Also, Figure 6.5 (a) and (b) represent phase portraits of x and y respectively. Obviously, we can see that neural activities are highly correlated with end point kinematics or dynamics.



(a) (b)

Figure 6.5: Results of SSC when applied to a portion of center-out data (1000 times bins). Each colored dot represents a different mode. The colored dots are overlaid on top of $x - \dot{x}$ and $y - \dot{y}$ plots (black lines) for visualization (a) $x - \dot{x}$ (b) $y - \dot{y}$.

6.3 Result and Discussion

6.3.1 Mode Identification via Sparse Subspace Clustering

Compared to one-dimensional free movements, center-out tasks are much more complicated for the brain to perform because the monkey needs to wait for and respond to specific instructions that it is trained to follow, and execute hand movements satisfying externally imposed requirements.

From the results, we show that SSC can separate the neural data vectors into groups or clusters, and it can reveal the modes which are highly correlated with dynamics of the end point as shown in Figure 6.5. Each figure summarizes the result of SSC when applied to 1000 time bins (100 seconds) of the center-out data where the colored dots are overlaid on top of various kinematics plots.

The two plots in Figure 6.4 show kinematics where each of the four modes occur with respect to position and velocity. The dark blue group seems to correspond to the hand being in the bottom half plane in both $x - y$ space and $\dot{x} - \dot{y}$ space. This suggests that there is a mode corresponding to the hand moving away from the center towards one of the lower targets. The orange group, on the other hand, suggests that there is a mode corresponding to the hand moving back to the center away from the lower targets. The light blue and light green groups make similar suggestions for the upper targets.

The two plots in Figure 6.4 show that perhaps the modes are related to both the position and velocity at the same time, so we overlaid the colored dots on the phase portrait, for example, $x - \dot{x}$ and $y - \dot{y}$ plots in Figure 6.5. While either time plots shown in Figure

6.3 or overlaid on top of position and velocity plots as shown in Figure 6.4 does not reveal clear correlations between kinematics and the discrete modes identified by SSC, a simple overlaying the different colored dots on top of phase portraits represents clear correlations as shown in Figure 6.5. Here, we see in Figure 6.5 that each mode corresponds to one quadrant in the $y - \dot{y}$ space. The figure reveals a correlation between identified discrete modes and kinematics in the x direction that was not as obvious: two of the modes have an S shape in $x - \dot{x}$ space, while the other two modes have a reflected S shape.

6.3.2 Mode Prediction

Each mode that SSC identifies in the center-out data has an associated range space in which the neural spike vectors lie. Once the modes and the range space of each mode are determined from the training data set, we can predict the modes of the neural data vectors in the test set as shown in Algorithm 1:

Algorithm 1: Mode Prediction via SSC

Input: training set $[X_1, X_2, ..., X_k]$, n_{sub} (the number of modes) and test data set $[X_{k+1}, X_{k+2}, ...]$.
Output: Range space U_j and predicted mode.

Mode Identification:
    Run SSC for training set then we can get clustered result.
    Calculate range space U_j for each subspace n_{sub}
Prediction: calculate error for test data X_{k+1}
    $e_j = ||(I - P_{u_j})X_{k+1}|| = ||(I - U_j U_j^T)X_{k+1}||$
    set $X_{k+1} \in \mathcal{R}(U_l)$ where $l = arg\min_j e_j$

As an example, for the center-out data set, we used the first 1000 time bins as a training set to establish the range spaces of each mode. Afterwards, we used the range spaces to predict the modes of the remaining 4000 time bins. Then, we compared the predicted modes to the obtained modes by clustering all 5000 time bins at once. The prediction accuracy is 98.5 %. The result of mode prediction in the center-out data is shown in Figure 6.6. The prediction shows the same patterns in $x - \dot{x}$ and $y - \dot{y}$ space with a few misclassifications. This result shows that SSC is able to identify discrete modes which the brain continues to exhibit over 400 seconds when performing center-out tasks. This implies that the brain uses the same time-invariant motor program so that SSC prediction is applicable over the duration of 400 seconds.

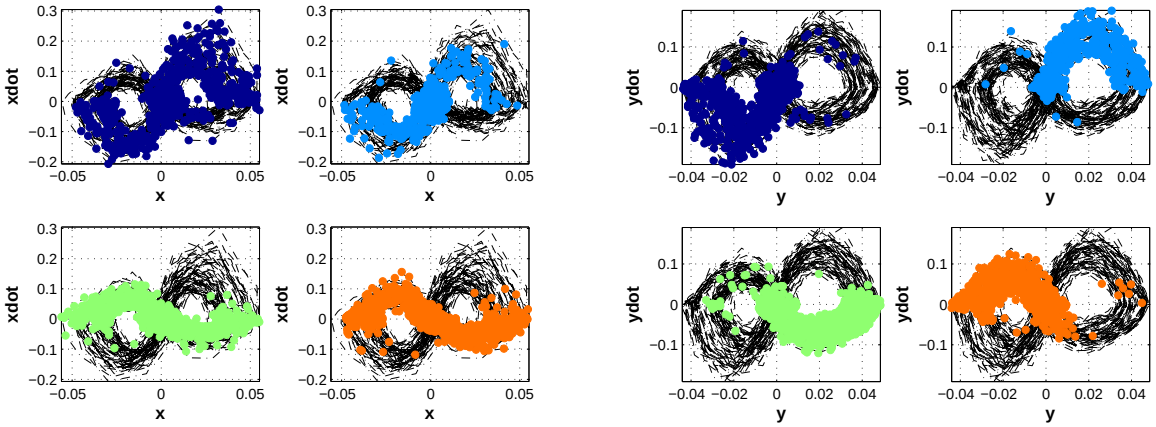


Figure 6.6: SSC prediction results in a portion of center-out data (1000 time bins used for training to predict modes of 4000 time bins). Each colored dot represents a different mode. The colored dots are overlaid on top of $x - \dot{x}$ and $y - \dot{y}$ plots (black lines) for visualization.

II. Low-rank Representation of Neural Activity and Detection of Submovement

In this study, Robust Principal Component Analysis (RPCA) is applied to neural spike datasets to extract neural signatures that signify the onset of submovements, a type of motor primitive. Given neural activity recorded from rhesus macaques during a set of reaches between targets in a horizontal plane, we aim to identify common event-related neural features and validate submovement-based motor primitives inferred from the hand velocity profiles. Such features represent common dynamic patterns across many experimental trials and may be used as a signature to detect of discrete events or state transitions such as submovement onset. We present RPCA, a method well suited for extracting data matrices' low-rank component, which represents dynamically meaningful structure. When applied to experimental data collected from monkeys, this method allows (1) removal of sparse corruption signal or task-irrelevant signal from data, (2) identification of task-related dynamic patterns, and (3) detection and prediction of submovements. We also explored using the Random Projection (RP) technique, a powerful method for dimensionality reduction. Applying RP to data prior to applying RPCA improved the submovement prediction performance by de-sparsifying neural data while preserving certain statistical characteristics of aggregate neural activity. The material in this part is based on the work presented in [38][115].

6.4 Characterizing Task-relevant Patterns

Neural activity is typically studied by averaging noisy spiking activity across multiple experimental trials to obtain an approximate neural firing rate that varies smoothly over time. However, if neural activity is more a reflection of internal neural dynamics rather than response to external stimulus, the time series of neural activity may differ even when an animal is performing nominally identical tasks [165]. This is particularly true of behavioral tasks involving perception, decision making, attention, or motor planning. In these settings, it is critical that the neural data not be averaged across trials, but instead be analyzed on a trial-by-trial basis [42]. Moreover, stimulus representations in some sensory systems are char-

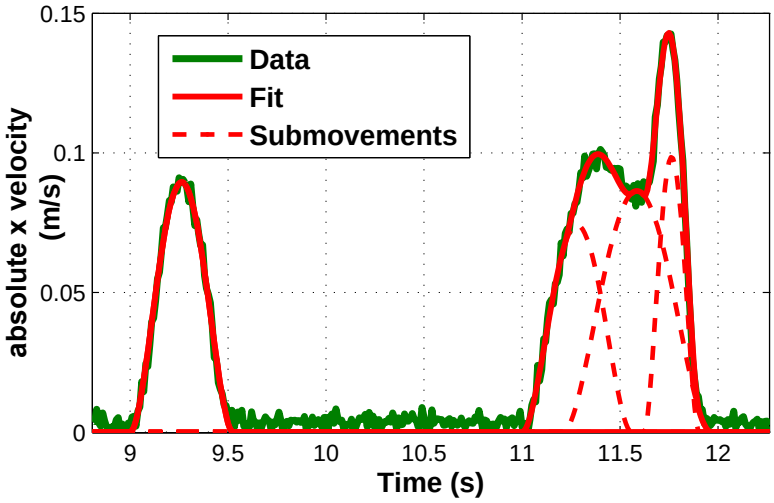


Figure 6.7: Velocity profile of a complex movement following a simple movement (experimental data). Thick green line: data. Thick red line: Reconstruction of data following submovement decomposition. Dashed red lines: individual submovements.

acterized by the precise spike timing of a small number of neurons [66][140][101], suggesting that the details of operations in the brain are embedded not only in the overall neural spike rate, but also the timings of spikes.

Neural models of supervised learning are usually concerned with processing static spatial patterns of intensities. For example, traditional PCA may be applied to the neural population activity. However, PCA does not find dimensions relevant to dynamical structure because traditional PCA begins by calculating data covariance, which describes the ellipsoid that best fits the data without regard to temporal information [42]. In the following section, we will show that the low-rank component of the data, which is related to the dynamical features in the population activity, can be extracted by using RPCA. Then, we will use the low-rank neural features to detect and predict the onset of submovements.

6.4.1 Submovement Decomposition

According to Gordon *et al.* [71] and Fishbach *et al.*[59], the hand speed profile as a function of time resulting from arm movements can be represented by a sum of bell-shaped functions, each of which is called a "submovement". One biological interpretation of submovement decompositions is that instead of applying continuous control to the arm, the brain controls arm movements by initiating discrete submovements. The velocity profile of a more complex movement looks like a sum of several bell-shapes, where submovements following the first can be interpreted as corrections to the initial submovement. A complex movement following a simple one is shown in Figure 6.7.

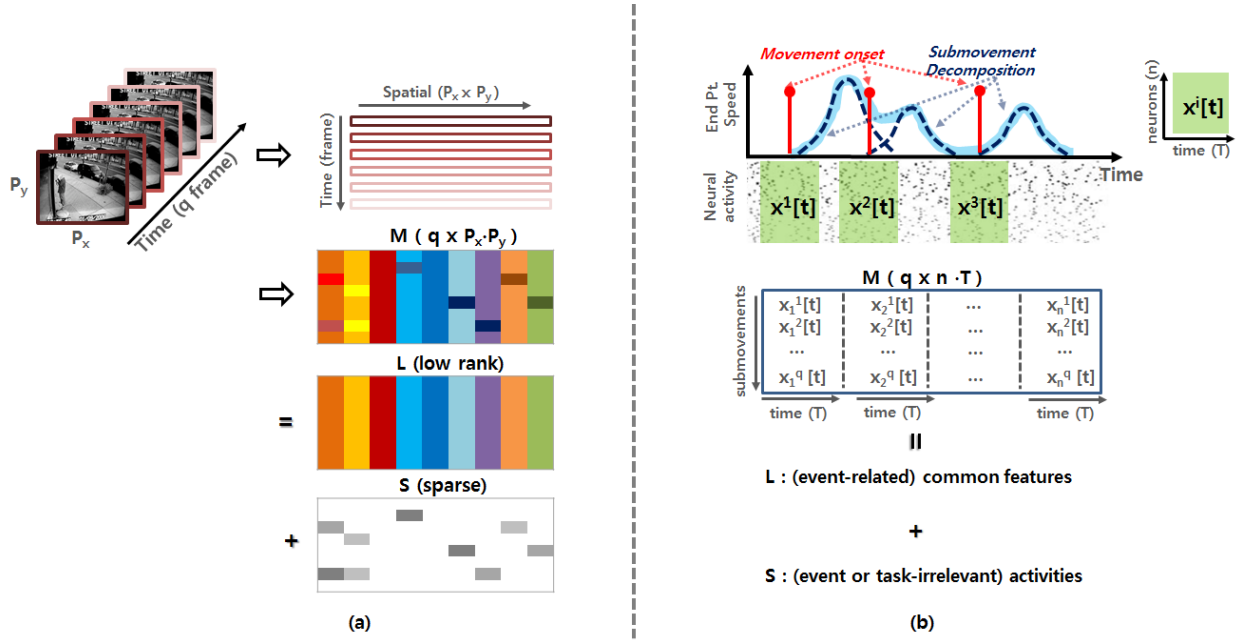


Figure 6.8: (a) RPCA applied to computer vision. A typical example of video surveillance where the low-rank component represents the unchanging background and the sparse component represents the movements in the foreground. (b) RPCA applied to neural systems. The low-rank component putatively represents (submovement relevant) neural signatures and the sparse component represents neural activity unrelated to submovement onset.

Rohrer and Hogan outline various types of roughly bell-shaped functions representing submovements and present algorithms for fitting sums of bell-shaped functions to kinematic data [125][126]. The types of bell-shape functions include the Gaussian curve, support-bounded log-normal curve, and the minimum jerk curve [1][60][152][71]. In this study, the planar velocity of the monkey's hand is decomposed using methods similar to in [126] into minimum jerk curves. In the subsequent RPCA analysis, submovements with small amplitude and/or long duration are ignored to avoid artifacts of overfitting.

6.4.2 Robust Principal Component Analysis (RPCA)

Suppose we are given a large data matrix $\mathbf{M}$, which has common features in the low-rank component and may contain some anomaly in the sparse component as shown in Figure 6.8 (a). It is natural to model the common variations as approximately the low-rank component $\mathbf{L}$, and the anomaly as the sparse component $\mathbf{S}$. For example, in video surveillance, we need to identify activities that stand out from the background given a sequence of video frames [28]. Figure 6.8 (a) shows that if we stack the video frames as rows of a matrix $\mathbf{M} \in \mathbb{R}^{q \times P_x \cdot P_y}$ where q is the number of frames, and P_x and P_y represent the number of pixels of 2-D images respectively, then the low-rank component $\mathbf{L}$ corresponds to the stationary background and

the sparse component $\mathbf{S}$ captures the moving objects in the foreground. Using RPCA, we can decompose $\mathbf{M}$:

$$\mathbf{M} = \mathbf{L} + \mathbf{S} \tag{6.4}$$

We can formulate this as follows:

$$\min_{L,S} ||\mathbf{L}||_* + \lambda||\mathbf{S}||_1 \text{ subject to } \mathbf{M} = \mathbf{L} + \mathbf{S} \tag{6.5}$$

where $||\mathbf{L}||_*$ denotes the nuclear norm of the matrix $\mathbf{L}$, i.e., the sum of the singular value of $\mathbf{L}$, and $||\mathbf{S}||_1 = \sum_{ij} |\mathbf{S}_{ij}|$ represents l_1-norm of $\mathbf{S}$. Choosing the tuning parameter λ to be $\lambda = 1/\sqrt{\max(n_1, n_2)}$, works well for incoherent matrices where n_1, n_2 represent the dimension of matrix $\mathbf{M}$ [28]. For practical problems, however, it is often possible to improve performance by choosing λ according to prior knowledge about the solution. Once we decompose $\mathbf{M}$ as $\mathbf{L} + \mathbf{S}$, then the low-rank component ($\mathbf{L}$) represents the common features and the sparse component ($\mathbf{S}$) represents anomalies in the dataset $\mathbf{M}$.

6.4.3 Neural population dynamics

The motor and premotor cortex have been extensively studied but their basic response properties are poorly understood [42]. Also, there is debate about whether neural activity relates to muscles or to abstract movement features. One possibility is that motor cortical activity represents movement parameters as per equation (6.6) and dynamical system that generates movements as per equation (6.7) [42]:

$$x_i(t) = f_i(\text{param}_1(t), \text{param}_2(t), \text{param}_3(t), ...) \tag{6.6}$$

$$\dot{\mathbf{x}}(t) = g(\mathbf{x}(t)) + \mathbf{u}(t) \tag{6.7}$$

In (6.6), $x_i(t)$ is the firing rate of neuron i at time t, f_i is its tuning function, and each param_j may represent a movement parameter such as hand velocity, target position or direction. In (6.7), $\mathbf{x} \in \mathbb{R}^n$ is a vector describing the firing rate of all neurons where n is the number of neurons, $\dot{\mathbf{x}}$ is its derivative, g is an unknown function, and $\mathbf{u}$ is an external input. In (6.7), neural activity is governed by some underlying dynamics and dynamical features should be present in the population activity.

Although neural population dynamics is typically much more complex than a simple linear model, here we simply consider a linear model because a piecewise linear modeling is the process of developing a series of locally linear models which approximates a nonlinear system such as (6.7). This example can be used to illustrate how we construct our input matrix $\mathbf{M}$ for analyzing neural activity data and in this setting, the low-rank component corresponds to the characteristics of dynamical system. Consider a simple linear model where unit activities reflect the underlying dynamical features as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{u}(t) \tag{6.8}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ represents the system or plant matrix and $\mathbf{u} \in \mathbb{R}^n$ represents an external input such as sensory input or task cue which affects the neural activity. We will align neural activity with the times that a monkey is engaged in a repeated task, so for simplicity, we assume no external input ($\mathbf{u} = \mathbf{0}$). Then, if the initial vector $\mathbf{x}_0 = \mathbf{x}(t = 0)$ is aligned with eigenvector $\mathbf{r}_k$ of the matrix $\mathbf{A}$, the dynamics is simple, $\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{r}_k = \lambda_k \mathbf{r}_k$ where λ_k is the corresponding eigenvalue; the solution of this equation is $\mathbf{x}(t) = \mathbf{r}_k e^{\lambda_k t}$. If $\mathbf{A}$ is diagonalizable, then any vector in an n-dimensional space can be represented by a linear combination of the right and left eigenvectors (denoted $\mathbf{l}_k$) of the matrix $\mathbf{A}$:

$$\mathbf{x}_0 = \sum_{k=1}^{n} (\mathbf{l}_k \cdot \mathbf{x}_0)\mathbf{r}_k \implies \mathbf{x}(t) = \sum_{k=1}^{n} \eta_k \mathbf{r}_k e^{\lambda_k t} \tag{6.9}$$

where $\eta_k \triangleq \mathbf{l}_k \cdot \mathbf{x}_0 \in \mathbb{R}$ and $\mathbf{x}(t) = \big[x_1(t), x_2(t), ..., x_n(t)\big]^T$.

Consider high dimensional time series data $\mathbf{x}^j[t] \triangleq \big[\mathbf{x}^j(1) \quad \mathbf{x}^j(2) \quad ... \quad \mathbf{x}^j(T)\big] \in \mathbb{R}^{n\times T}$ where the superscript represents jth submovement ($j = 1, ..., q$), $(\cdot)$ represents the sample time points, n is the number of neurons and T is the number of time points. By (6.9), we can write $\mathbf{x}^j[t]$ as

$$\mathbf{x}^j[t] = \left[\sum_{k=1}^{n} \eta_k^j \mathbf{r}_k e^{\lambda_k 1\cdot\Delta T} \quad \sum_{k=1}^{n} \eta_k^j \mathbf{r}_k e^{\lambda_k 2\cdot\Delta T} \quad ... \quad \sum_{k=1}^{n} \eta_k^j \mathbf{r}_k e^{\lambda_k T\cdot\Delta T}\right] \tag{6.10}$$

where η_k^j represents the initial condition for jth submovement and ΔT represents a sampled time or time bin (in this study, we choose $\Delta T = 50$ ms). Therefore, neural activity $\mathbf{x}^j[t]$ reflects underlying dynamics related to eigenvalues (λ_k) and eigenvectors ($\mathbf{r}_k$), even though the initial condition may be different across submovements (η_k^j). If we stack the neural activities $\mathbf{x}^j[t]$ across all the submovements and represent these data as a matrix $\mathbf{M} \in \mathbb{R}^{q\times n\cdot T}$ as shown in (6.11) and Figure 6.8 (b), then we can extract common dynamical features.

$$\mathbf{M} = \begin{bmatrix} x_1^1[t] & x_2^1[t] & ... & x_n^1[t] \\ x_1^2[t] & x_2^2[t] & ... & x_n^2[t] \\ ... & ... & ... & ... \\ x_1^q[t] & x_2^q[t] & ... & x_n^q[t] \end{bmatrix} = \big[\mathcal{X}_1 \quad \mathcal{X}_2 \ ... \ \mathcal{X}_n\big] \triangleq \mathbb{X} \tag{6.11}$$

where $\mathcal{X}_i \triangleq \big[\mathbf{e}_i^T\mathbf{x}^1[t]; \mathbf{e}_i^T\mathbf{x}^2[t]; ...; \mathbf{e}_i^T\mathbf{x}^l[t]\big] \in \mathbb{R}^{q\times T}$ represents the temporal neural activity of the ith neuron, $\mathbf{e}_i \in \mathbb{R}^n$ is a unit vector, and q is the number of submovements across all data. Note that we aligned each time series data $\mathbf{x}^j[t]$ to the same temporal condition (submovement onset) as shown in Figure 6.8 (b) but we do not separate different types of submovement. For example, submovements with different reach directions, or with different ordinal positions in an overlapped series of submovements, are combined in our input matrix $\mathbb{X}$. Even though the activity of each neuron may vary significantly across submovements, some portion of the variability may reflect common features.

6.5 Disentangling the low-rank and sparse components

In the previous section, we showed how to construct our input matrix $\mathbb{X}$. However, it is still not clear whether separation into a low-rank and a sparse component makes sense. In [28], Candès *et al.* discuss the identifiability issue. For example, suppose the matrix $\mathbf{M}$ is equal to $\mathbf{e}_1\mathbf{e}_1^*$. Then, we cannot decide whether it is low-rank or sparse since $\mathbf{M}$ is both sparse and low-rank. To make the problem meaningful, Candès *et al.* require that the low-rank component $\mathbf{L}$ must not be sparse. In other words, the singular vectors of the low-rank matrix $\mathbf{L}$ should be reasonably spread out. Another identifiability issue arises if the sparse matrix has low-rank. For example, this will occur if all the nonzero entries of $\mathbf{S}$ occur in a few columns or in a few rows. To avoid degenerate situations, they assume that the sparsity pattern of the sparse component is by random variables from a uniform distribution. In many applications in image and video analysis, practical low-rank and sparse separation gives visually appealing solutions.

However, only a small subset of the whole ensemble of neurons is active at any moment, as seen in the dataset shown in Figure 6.9 (a) and Figure 6.11 (a). This implies that the singular vectors of the low-rank component may not be reasonably spread out due to the sparsity of the input matrix $\mathbf{M}$. This observation about the neural data makes the solution to the separation problem seem ambiguous. However, random projection (RP) can both succinctly summarize our sparse data in a lower-dimensional space, while at the same time de-sparsify the data so that they can be reliably separated into a low-rank and a sparse component.

6.5.1 Random Projection (RP)

Recent theoretical work has identified random projection as a promising dimensionality reduction technique[49]. Projecting the data onto a random lower-dimensional subspace preserves the similarity of different data vectors, for example, the distances between the points are approximately preserved. Also, RP can reduce the dimension of data while keeping clusters of data points well-separated [49]. Moreover, using RP is computationally significantly less expensive than using techniques such as PCA because RP is data-independent.

The idea of RP is that a small number of random linear projections can preserve key information. Theoretical work [49][17][54] guarantees that with high probability, all pairwise Euclidean and geodesic distances between points on a low-dimensional manifold are well-preserved under the mapping $\Psi : \mathbb{R}^n \to \mathbb{R}^m, m < n$. Consider a linear signal model

$$\mathbf{y}(t) = \mathbf{\Psi}\mathbf{x}(t) = \sum_{i=1}^{n} x_i(t)\psi_i \in \mathbb{R}^m \tag{6.12}$$

where $\Psi = \begin{bmatrix}\psi_1 & \psi_2 & ... & \psi_n\end{bmatrix}$ is an $m \times n$ projection matrix whose elements are drawn randomly from independent identical distributions. First, note that the dimensionality of the

data $\mathbf{x}$ is reduced since $m < n$. Also, if we define $\mathcal{Y}_i \triangleq \left[\bar{\mathbf{e}}_i^T\mathbf{y}^1[t]; \bar{\mathbf{e}}_i^T\mathbf{y}^2[t]; ...; \bar{\mathbf{e}}_i^T\mathbf{y}^l[t]\right] \in \mathbb{R}^{q\times T}$ where $\bar{\mathbf{e}}_i \in \mathbb{R}^m$ and $\mathbb{Y} \triangleq \begin{bmatrix}\mathcal{Y}_1 & \mathcal{Y}_2 & ... & \mathcal{Y}_m\end{bmatrix}$, then

$$\mathbb{Y}^T = (\mathbf{\Psi} \otimes \mathbf{I}_T)\mathbb{X}^T \quad \text{or} \quad \mathbb{Y} = \mathbb{X}(\mathbf{\Psi}^T \otimes \mathbf{I}_T) \tag{6.13}$$

where $\otimes$ represents the Kronecker product[1] and $\mathbf{I}_T \in \mathbb{R}^{T\times T}$ is an identity matrix.

6.5.2 Identifiability, dimensionality reduction and benefits

By using RP, we can handle the identifiability issue because our input matrix $\mathbb{Y} \in \mathbb{R}^{q\times m\cdot T}$ is denser than the original input matrix $\mathbb{X} \in \mathbb{R}^{q\times n\cdot T}$. RP also preserves the similarity of the data vectors well and reduces the dimension. Furthermore, because our data are sparse, no information is lost from performing RP on $\mathbb{X}$.

(a) Identifiability

Suppose our input $\mathbb{X}$ in equation (6.11) can be decomposed as $\mathbb{X} = \mathbf{L} + \mathbf{S}$:

$$\mathbb{X} = \mathbf{L} + \mathbf{S} = \sum_{i=1}^{d_L} \sigma_i \mathbf{u}_i \mathbf{v}_i^* + \sum_{i=1}^{d_S} \lambda_i \mathbf{a}_i \mathbf{b}_i^* \tag{6.14}$$

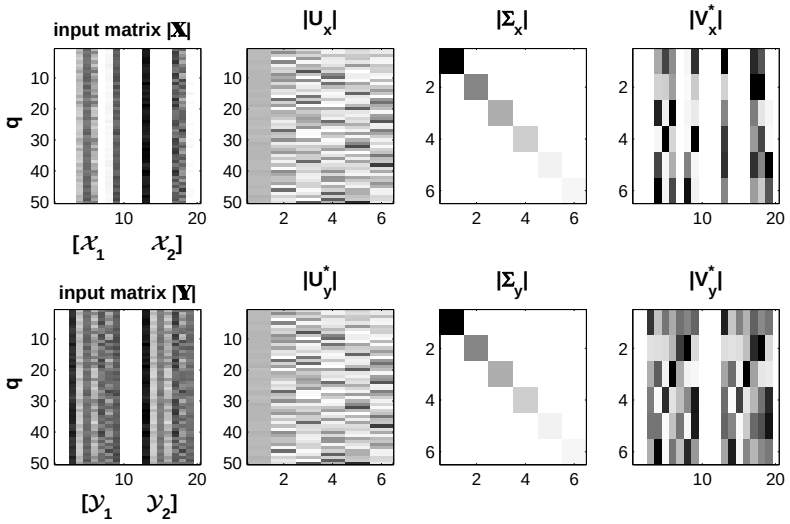
where σ_i are the positive singular values, $\mathbf{u}_i \in \mathbb{R}^{q\times 1}, \mathbf{v}_i^* \in \mathbb{R}^{1\times n\cdot T}$ are the left- and right-singular vectors of $\mathbf{L}$, and d_L represents the rank of the matrix $\mathbf{L}$. d_S is the number of sparse components in $\mathbf{S}$, and $\mathbf{a}_i \in \mathbb{R}^q, \mathbf{b}_i \in \mathbb{R}^{n\cdot T}$ are sparse with only one nonzero entry respectively. By using RP, we have for $\mathbb{Y}$,

$$\begin{aligned}
\mathbb{Y} &= \mathbb{X}(\mathbf{\Psi}^T \otimes \mathbf{I}_T) = \mathbb{X}\mathbf{R} = \mathbf{L}\mathbf{R} + \mathbf{S}\mathbf{R} \\
&= \sum_{i=1}^{d_L} \sigma_i \mathbf{u}_i (\mathbf{R}^T\mathbf{v}_i)^* + \sum_{i=1}^{d_S} \lambda_i \mathbf{a}_i (\mathbf{R}^T\mathbf{b}_i)^* \\
&= \sum_{i=1}^{d_L} \sigma_i \mathbf{u}_i \tilde{\mathbf{v}}_i^* + \sum_{i=1}^{d_S} \lambda_i \mathbf{a}_i \tilde{\mathbf{b}}_i^*
\end{aligned} \tag{6.15}$$

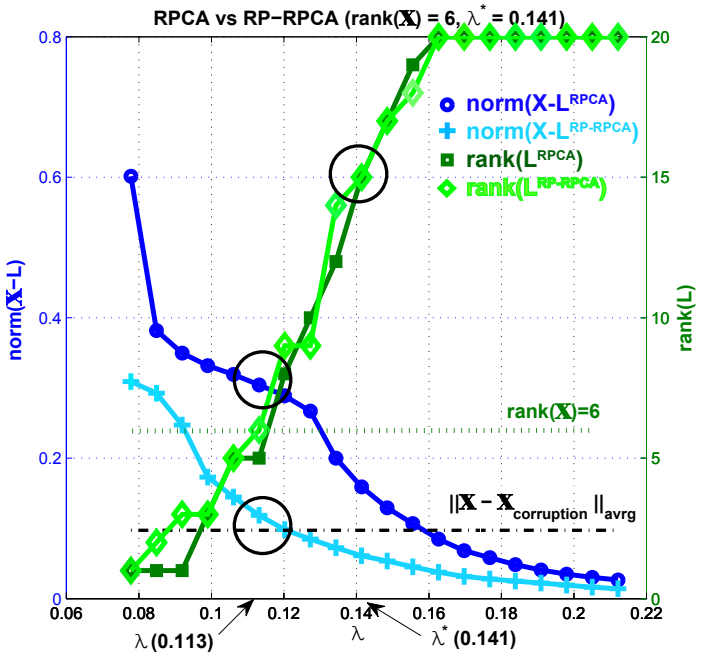
where we denote $(\mathbf{\Psi}^T \otimes \mathbf{I}_T)$ by $\mathbf{R}$. As we mentioned above, our input $\mathbb{X}$ is sparse, so the singular vectors of the low-rank matrix $\mathbf{L}$ might not be reasonably spread out. This is especially true for $\mathbf{v}_i^*$, shown in Figure 6.9(a) (upper). However, by using RP (multiplying by $\mathbf{R}$), the singular vectors $\tilde{\mathbf{v}}_i$ of the resulting matrix become reasonably spread out, as shown in Figure 6.9(a) (lower).

[1] If $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is a $p \times q$ matrix, then the Kronecker product $\mathbf{A} \otimes \mathbf{B}$ is the $mp \times nq$ block matrix:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

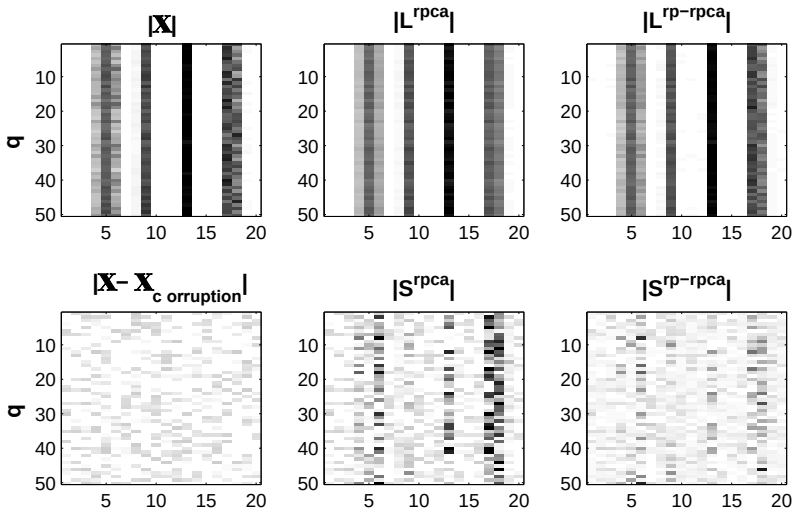


(a)

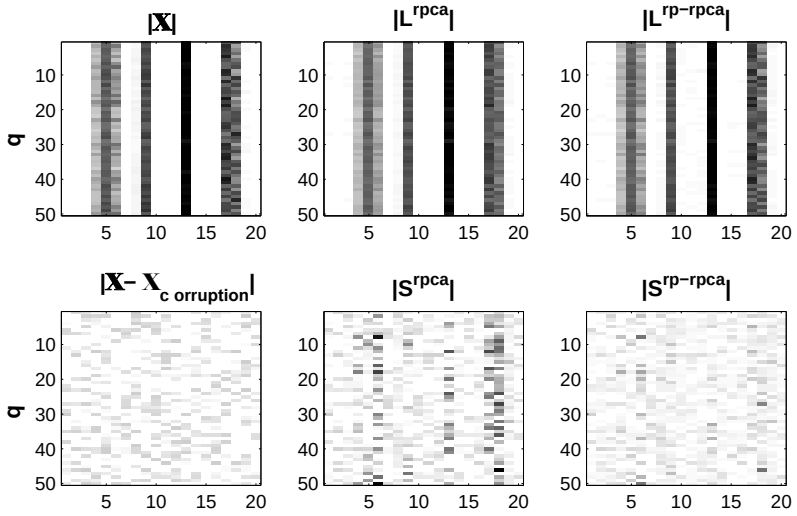


(b)

Figure 6.9: (a) (upper) Input matrix $\mathbb{X}$ and singular value decomposition (SVD) ($\mathbb{X} = \mathbf{U}_x\mathbf{\Sigma}_x\mathbf{V}_x^*$). (lower) Randomly projected input matrix $\mathbb{Y}$ and SVD ($\mathbb{Y} = \mathbf{U}_y\mathbf{\Sigma}_y\mathbf{V}_y^*$). Note that since rank($\mathbb{X}$)=6, $\mathbf{U}_x \in \mathbb{R}^{q\times 6}$, $\mathbf{\Sigma}_x \in \mathbb{R}^{6\times 6}$, $\mathbf{V}_x^* \in \mathbb{R}^{6\times n\cdot T}$. In order to show how well singular vectors are spread out, we show the absolute value of each component. White represents zero value. (b) RPCA results. We run RPCA for sparsely corrupted $\mathbb{X}_{corruption}$, $\mathbb{Y}_{corruption}$. (We added sparse corruption to $\mathbb{X}$ as shown in Figure 6.10.) Left y-axis represents the norm of $\mathbb{X} - \mathbf{L}$ and the right y-axis shows the rank of $\mathbf{L}$.



(a)



(b)

Figure 6.10: RPCA output. (a) For $\lambda = 0.113$, both $\mathbf{L}^{\text{rpca}}$ and $\mathbf{L}^{\text{rp-rpca}}$ have rank 6 ($\approx$ rank($\mathbb{X}$)) as shown in Figure 6.9(b). There is a big difference between $\mathbf{S}^{\text{rpca}}$ and the constructed corrupted signal ($\mathbb{X} - \mathbb{X}_{corr}$) (b) For $\lambda^* = 0.141$, $\mathbf{S}^{\text{rp-rpca}}$ is close to $\mathbb{X} - \mathbb{X}_{corr}$ but the low-rank components are misidentified by both RPCA and RP-RPCA because both $\mathbf{L}^{\text{rpca}}$ and $\mathbf{L}^{\text{rp-rpca}}$ have rank 15.

(b) Compressive Sensing (CS) and Recovery of the exact signal

Compressive Sensing (CS) theory states that with high probability, every k-sparse signal $z \in \mathbb{R}^n$ (i.e., every signal z having only k nonzero components) can be recovered from just $m(\geq ck\log(n/k))$ linear measurements $b = Az$. Here, A represents an $m \times n$ measurement (or encoding) matrix drawn randomly from an acceptable distribution for satisfying incoherence condition and c is a small constant [13]. CS decoding involves recovering the signal $z \in \mathbb{R}^n$ from its measurements $b = Az \in \mathbb{R}^m$ where $m < n$ in general. Although such inverse problems are generally ill-posed, CS recovery algorithms exploit the additional assumption of sparsity in the basis A to identify the correct signal z from an uncountable number of possibilities.

In equation (6.13), we can consider $(\mathbf{\Psi} \otimes \mathbf{I}_T)$ as a measurement matrix A. Then, since we choose a incoherent matrix $\mathbf{\Psi}$ drawn randomly with i.i.d. and the coherence of $\mathbf{\Psi}$ is exactly the same as that of $\mathbf{\Psi} \otimes \mathbf{I}_T$ (i.e., $\mu(\mathbf{\Psi}) = \mu(\mathbf{\Psi} \otimes \mathbf{I}_T)$), we can recover the exact signal $\mathbb{X}$ from the randomly projected signal $\mathbb{Y}$. Recall that the jth columns of $\mathbb{X}^T$ and $\mathbb{Y}^T$ represent the high dimensional time series data and projected time series data of the jth submovement respectively. Since the measurement matrix $\mathbf{\Psi} \otimes \mathbf{I}_T$ satisfies the coherence condition and the jth column of $\mathbb{X}^T$, $\mathbb{X}^T_{(:,j)}$, is known to be sparse, we can recover the original data from $\mathbb{Y}^T_{(:,j)}$ by solving the l_1-minimization:

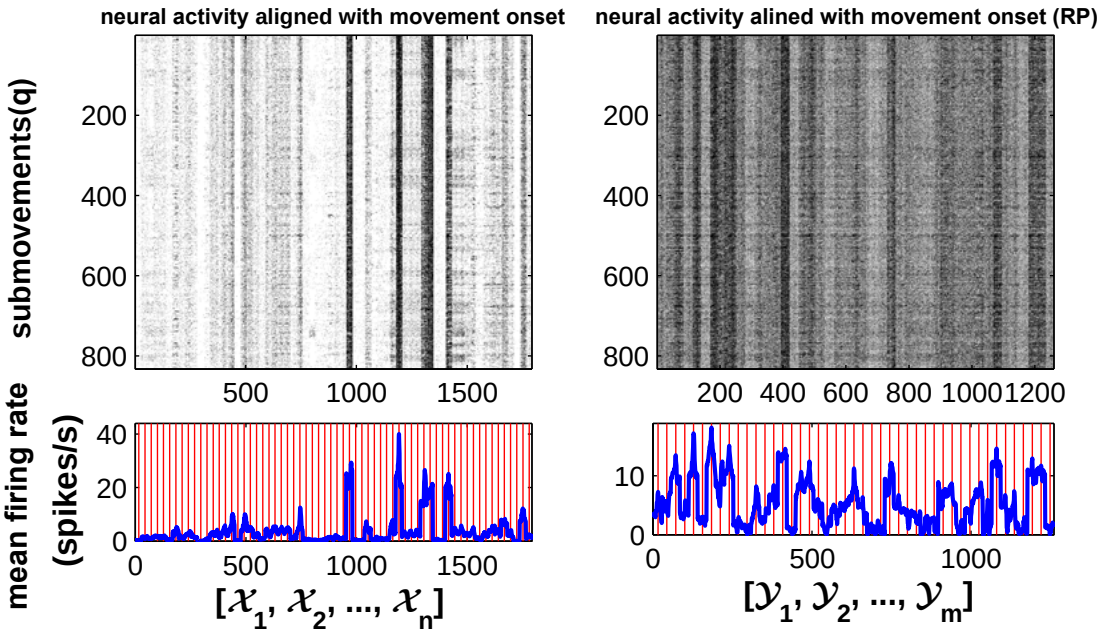
$$\min_z ||z||_1 \quad \text{subject to} \quad \mathbb{Y}^T_{(:,j)} = (\mathbf{\Psi} \otimes \mathbf{I}_T)z \tag{6.16}$$

Therefore, since our input $\mathbb{X}$ is sparse and the measurement matrix is incoherent by choice of $\mathbf{\Psi}$, RP does not lose any information even though dimensionality is reduced.

(c) Dimension Reduction and Eccentric Distribution

The dimension of the original input $\mathbb{X}$ is $q \times (n \cdot T)$, where q is the number of submovements, n is the number of neurons, and T is the number of time points. The dimension of the new input $\mathbb{Y}$ after application of RP to $\mathbb{X}$ is $q \times (m \cdot T)$ where $m < n$. For example, suppose our data contain the neural activity of 100 cells ($n = 100$), a typical value. Also, suppose that we consider 20 time points for each neuron ($T = 20$) and 400 submovements ($q = 400$). Then, the dimension of the original input $\mathbb{X} \in \mathbb{R}^{q \times n \cdot T}$ is 400×2000. Using RP with $m = 60$, the dimension of the projected input $\mathbb{Y} \in \mathbb{R}^{q \times m \cdot T}$ can be reduced to 400×1200.

In [49], Dasgupta showed that even if the original distribution of data samples is highly skewed (having an ellipsoidal contour of high eccentricity), its projected counterparts will be more spherical. For example, neural activity data vectors often form very eccentric clusters. As shown by the different amplitude for different neurons in Figure 6.11 (a), some neurons are highly activated (30-40 spikes/sec) but others typically have only a few spikes per second. Since it is conceptually much easier to design algorithms for spherical clusters than ellipsoidal ones, this feature of random projection can simplify the separation into the low-rank and sparse components. Therefore, using RP, we can reduce the computational complexity of



(a) (b)

Figure 6.11: RPCA input matrix. $|\mathbb{X}| \in \mathbb{R}^{q\times n\cdot T}$(a) and randomly projected input matrix $|\mathbb{Y}| \in \mathbb{R}^{q\times m\cdot T}$(b) where $q = 832$(submovements), $n = 64$(units), $m = 45, T = 28$(time points), and their averaged neural activity across all submovements $|\bar{\mathbb{X}}|, |\bar{\mathbb{Y}}|$ (red color represents submovement onset times for each $\mathcal{X}_i, \mathcal{Y}_j$ where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$).

the non-smooth convex optimization, in particular l_1 and nuclear norms minimization, used in RPCA[2].

6.5.3 Simple Examples

To illustrate the issue of identifiability and how RP can alleviate the issue, we consider a simple example: we generate a sparse low-rank input matrix $\mathbb{X} \in \mathbb{R}^{50\times 2\cdot 10}$ ($q = 50$, $n = 2$, $T = 10$) where the rank of $\mathbb{X}$ is 6. Figure 6.9(a)(upper) shows $\mathbb{X}$ and its singular value decomposition (SVD). Each row of the matrix containing the right-singular vectors $\mathbf{V}_x^*$ of $\mathbb{X}$ has many zeroes by construction. After applying RP, we obtain $\mathbb{Y}$ which is much denser; each row of $\mathbf{V}_y^*$ has fewer zeroes. By using RP, singular vectors in the low-rank component of the input matrix become reasonably spread out. Note that in this example we chose the same dimension for the input $\mathbb{X}$ and $\mathbb{Y}$ ($m = n$). This is done so that $\mathbf{\Psi} \in \mathbb{R}^{m\times n}$ in equation

[2]Many speedup methods were developed in optimization by avoiding large-scale SVD. In [107], Mu *et al.* demonstrated the power of projected matrix nuclear norm by reformulating RPCA and in [170], Zhou *et al.* demonstrated the effectiveness and the efficiency of Bilateral Random Projections. However, both methods consider a dense matrix $\mathbb{X}$ while in this study we consider the case when the input matrix $\mathbb{X}$ is sparse.

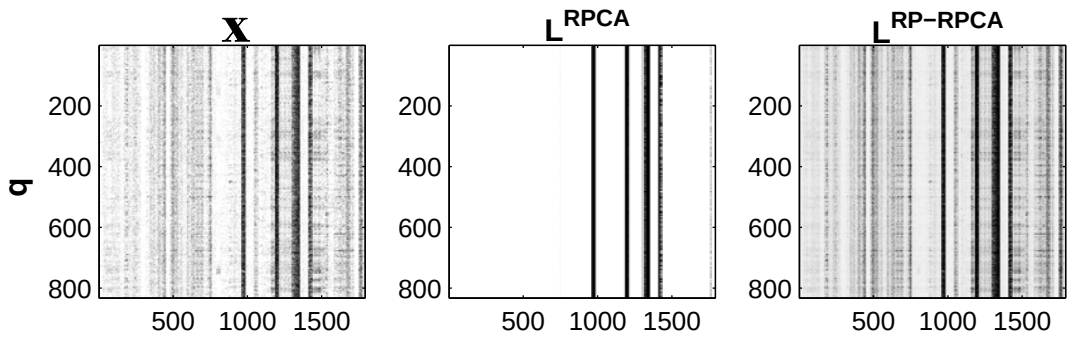


Figure 6.12: The low-rank matrices from both RPCA and RP-RPCA where $\mathbb{X}$ are input matrices from Figure 6.11 and we choose $m = n = 64$ for the comparison.

(6.12) is invertible, allowing us to compare the outputs of RPCA and RP-RPCA directly, as described below. In general, choosing $m < n$ makes $\mathbb{Y}$ much denser because information is compressed by RP.

Figure 6.9(b) shows the statistics of both RPCA and RP-RPCA (in which RPCA is applied to the matrix $\mathbb{Y}$) as a function of the tuning parameter λ, and Figure 6.10 shows the output of RPCA and RP-RPCA for two values of λ. To evaluate the performance of separation into a low-rank and a sparse component, we add sparse corruption for $\mathbb{X}$ as follows:

$$\begin{aligned} \mathbb{X}_{corruption} &= \mathbb{X} + \mathbf{S}_{corruption} \\ \mathbb{Y}_{corruption} &= \mathbb{X}_{corruption}\mathbf{R} = \mathbb{X}\mathbf{R} + \mathbf{S}_{corruption}\mathbf{R} \end{aligned}$$

where $\mathbf{R} = (\mathbf{\Psi}^T \otimes \mathbf{I}_T)$ is the projection and $\mathbb{Y}_{corruption}$ is the projected corrupted input matrix $\mathbb{X}_{corruption}$. To compare the performance of RP-RPCA with RPCA, we first decompose $\mathbb{Y}_{corruption}$ into a its low-rank and sparse components. Then, we invert the projection:

$$\begin{aligned} \mathbb{X}_{corruption} &= \mathbf{L}^{rpca} + \mathbf{S}^{rpca} \quad \text{(orginal RPCA)} \\ &= \mathbb{Y}_{corruption}\mathbf{R}^{-1} \\ &= (\mathbf{L}^{rpca}_{\mathbb{Y}} + \mathbf{S}^{rpca}_{\mathbb{Y}})\mathbf{R}^{-1} \\ &\triangleq \bar{\mathbf{L}}^{rpca} + \bar{\mathbf{S}}^{rpca} \quad \text{(RP-RPCA)} \end{aligned}$$

where we define $\bar{\mathbf{L}}^{rpca} \triangleq \mathbf{L}^{rpca}_{\mathbb{Y}}\mathbf{R}^{-1}$ and $\bar{\mathbf{S}}^{rpca} \triangleq \mathbf{S}^{rpca}_{\mathbb{Y}}\mathbf{R}^{-1}$. In this example, inverting $\mathbf{R}$ is possible because we choose $\mathbf{\Psi} \in \mathbb{R}^{m \times n}$ where $m = n$. If $m < n$, then we have to solve l_1-minimization in equation (6.16) to obtain $\bar{\mathbf{L}}^{rpca}$ and $\bar{\mathbf{S}}^{rpca}$ instead.

Figure 6.9(b) shows the result of running RPCA with different λ values in equation (6.5). In this example, $\lambda^* = 1/\sqrt{\max(q, n \cdot T)} = 1/\sqrt{50}$. Since our input is still sparse in this case, the rank of both $\mathbf{L}^{rpca}, \bar{\mathbf{L}}^{rpca}$ is 15 for $\lambda^* = 0.141$ (rank$(\mathbb{X}) = 6$). If we choose $\lambda = 0.113$ (discounting the penalty for sparse component), the ranks of $\mathbf{L}^{rpca}, \bar{\mathbf{L}}^{rpca}$ are approximately 6, which is the same as the rank of the original input $\mathbb{X}$. With this choice of λ, for RPCA we find that $||\mathbb{X} - \mathbf{L}^{rpca}||$ is much bigger than the original corruption signal $||\mathbb{X} - \mathbb{X}_{corruption}|| =$

$||\mathbf{S}_{corruption}||$. On the other hand, for the RP-RPCA, we have $||\mathbb{X} - \bar{\mathbf{L}}^{rpca}|| \approx ||\mathbf{S}_{corruption}||$. Therefore, for RP-RPCA, the separation of the low-rank component and sparse component is close to the true solution but for original RPCA, we have misidentification in both the low-rank and sparse components. Figure 6.10 shows the output of RPCA and RP-RPCA with two different λ values. We can easily see that $\mathbf{S}^{rpca}$ shows characteristics of the low-rank component in Figure 6.10 (middle columns of each panel).

6.6 Application to Neural Data

6.6.1 Experimental Setup

The experiment involved an adult male rhesus macaque instructed to make visually-guided planar reaches with its right hand. To elicit additional submovements, the task was made more difficult by imposing "target jumps", i.e., changes in the position of the target midway through the reach. Neural units from primary motor and dorsal premotor cortex were recorded from chronically-implanted arrays. Spikes were sorted and binned every 50ms.

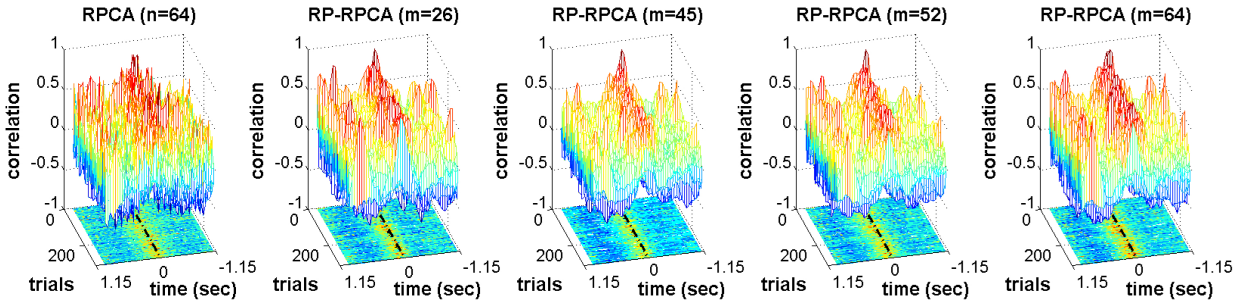
Hand velocity data (sampled at 100 Hz) were decomposed into a sum of minimum-jerk basis functions. Figure 6.11 (a) shows the actual neural activities aligned with movement onset. The aligned neural activity shows that the ratios between units' mean firing rates are fairly constant from the salient vertical striations in the plots and temporal patterns exists across all the submovements. Also, as mentioned previously, the neural population activities are sparsely active (white color represents 0 spikes/sec) and show eccentric behavior; for example, some neurons have a much higher spiking rate than others.

6.6.2 Prediction

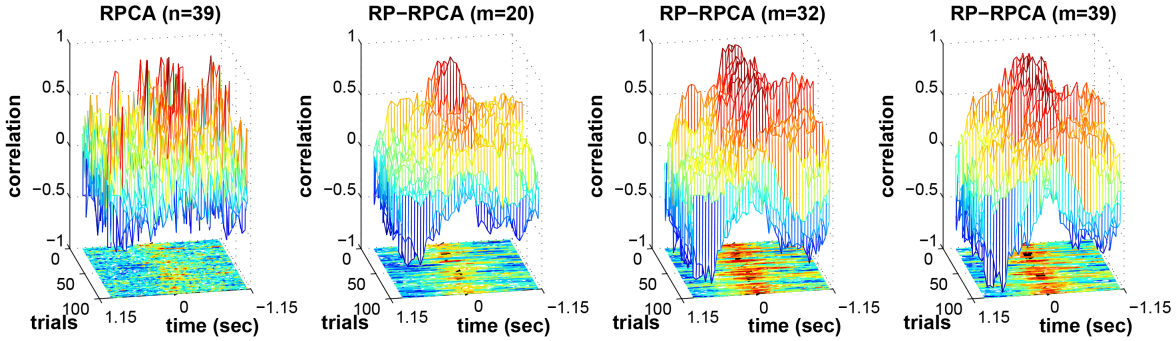
In order to extract the common features, we divide our dataset into training (70%) and test (30%) datasets. First, we run RPCA and RP-RPCA to extract the low-rank components, and then use these components as signatures or templates to detect submovement onset. Here, we simply use a correlation function as our metric:

$$\gamma_k = \frac{\langle s_k, \bar{\mathbf{T}} \rangle}{||s_k||||\bar{\mathbf{T}}||} \tag{6.17}$$

where $\bar{\mathbf{T}}$ represents the extracted temporal characteristics or template (in this case, an average of the low-rank component across submovements). The neural activity signal can be denoted as $s_k = \begin{bmatrix} x_1^k[t] & x_2^k[t] & ... & x_n^k[t] \end{bmatrix} \in \mathbb{R}^{1 \times n \cdot T}$ (for RPCA) or $s_k = \begin{bmatrix} y_1^k[t] & y_2^k[t] & ... & y_m^k[t] \end{bmatrix} \in \mathbb{R}^{1 \times m \cdot T}$ (for RP-RPCA). For practical purposes, we can choose a correlation threshold and if the correlation is over the chosen threshold, we label a submovement onset as detected.



(a)

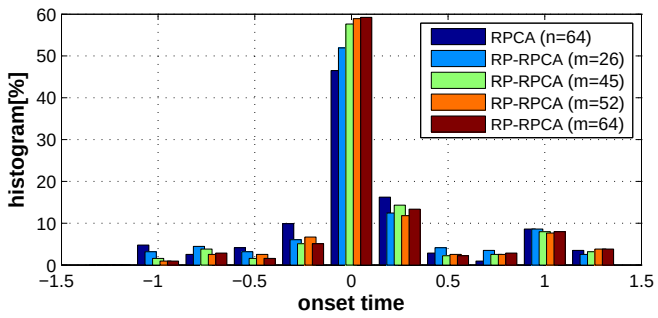


(b)

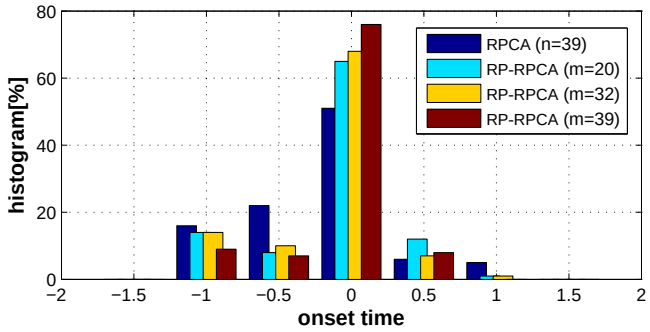
Figure 6.13: Correlation between neural activity and the common feature extracted by RPCA and RP-RPCA (a) dataset: jeev0617 (movements: $\geq 3.5cm$) (b) dataset: jeev0702 (movements: $\geq 1cm$).

6.6.3 Results

Figure 6.12 shows the low-rank matrix from both RPCA and RP-RPCA using input matrices $\mathbb{X}$ in Figure 6.11. For simple comparison, we choose $m = n$ since we do not need to solve equation (6.16). Since $\mathbb{X}$ is sparse and has an eccentric distribution, the singular vectors may not be reasonably spread out. Applying RPCA directly to $\mathbb{X}$ would result in the low-rank component being composed of only highly modulated neural activity. On the other hand, RP-RPCA can extract a low-rank component from a more distributed set of neural dimensions than RPCA alone can. Figure 6.13 shows correlations between neural activity and the extracted common feature in test datasets. To accurately predict submovement onset times found by submovement decomposition, the correlation function should peak around the movement onset time (time 0). The RPCA result (left) shows many false positives, indicated by the spurious peaks far away from time 0. For RP-RPCA, we compare the performance for different projection dimensions ($m = 26(40\%$ of the number of original dimensions, i.e., neural units), $45(70\%), 52(80\%)$ and $64(100\%)$; $n = 64$) in Figure 6.13(a). For $m = 26$, there exist inconsistent modulations across different submovements. However, for $m = 45$ and $m = 52$, the correlation functions consistently peak around the submovement onset time even though the input matrix dimension is reduced. Similarly, Figure 6.13 (b) shows that



(a)



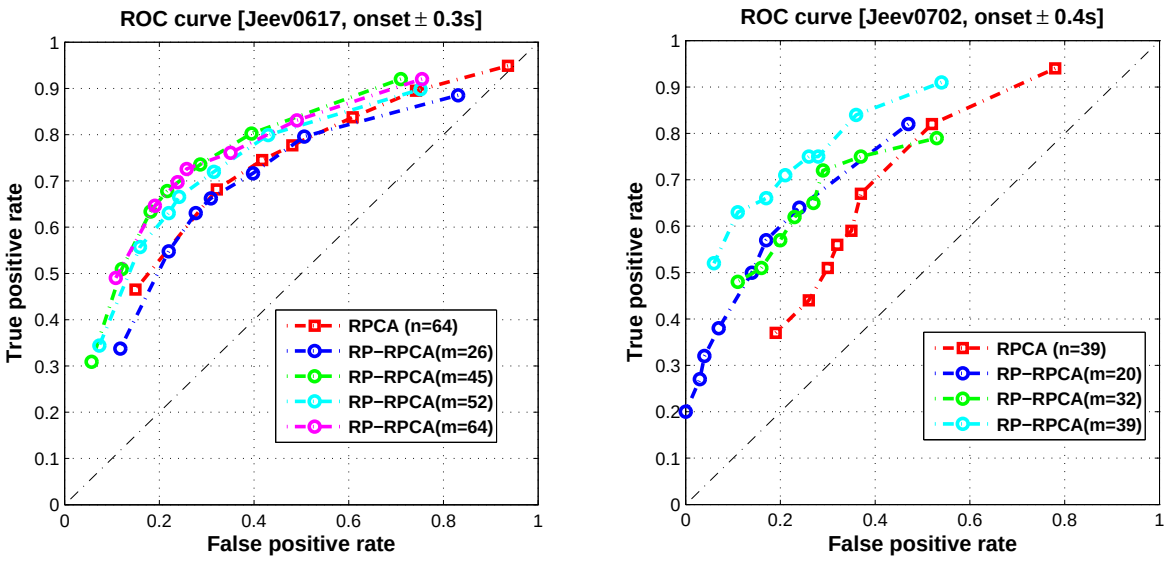
(b)

Figure 6.14: Histogram of the prediction results: for each submovement, we look for the time at which the correlation functions in Figure 6.13 is maximum (a) dataset: jeev0617 (b) dataset: jeev0702.

RP-RPCA results represent high modulation around onset while RPCA result shows many spurious peaks far way from onset time, where the task was made more difficult by imposing "target jumps".

Figure 6.14 (a) and (b) show the histograms that summarize our prediction results. Each bar counts the number of times that the submovements are predicted to lie in each time interval. For each submovement, we look for the time at which the correlation functions in Figure 6.13 is maximum. Figure 6.14 (a) RP-RPCA ($m = 45, 52, 64$) shows that 80% of the predictions are accurate within $\pm 0.25s$. Figure 6.14(b) shows the prediction results of "target jump case".

In practical terms, for the application in a brain-machine interface, one would threshold the correlation signal online to generate real-time submovement onset detection. Figure 6.15 represents receiver operating characteristic (ROC) curve based on Figure 6.13. Here, we vary thresholds for correlation score and show ROC result. We define True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) as follows:



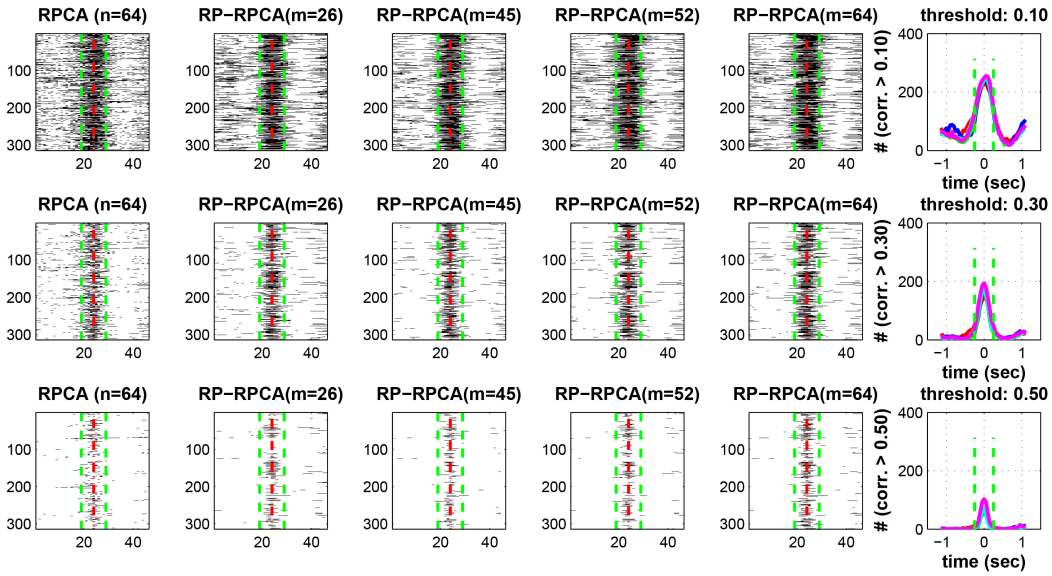
(a) (b)

Figure 6.15: Receiver Operating Characteristic (ROC) curve of different thresholds: (a) jeev0617 (b) jeev0702.

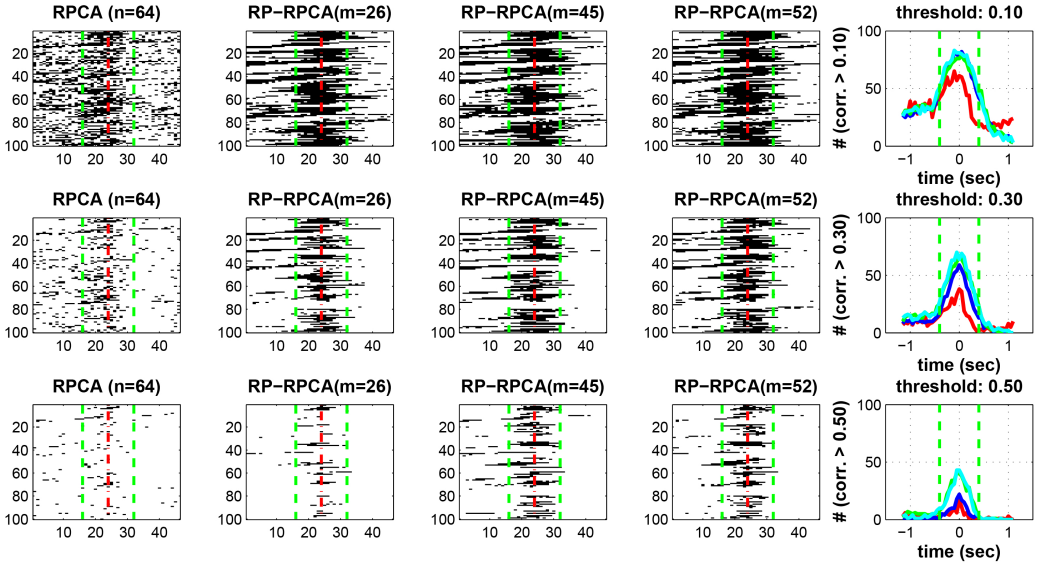
Algorithm 2: TP, FP, FN and TN can be formulated as follows;
if $\gamma_k \geq \Theta$ $(k \in \mathcal{S})$ then
$TP \leftarrow TP + 1$
else
$FN \leftarrow FN + 1$
end if
if $\gamma_k \geq \Theta$ $(k \in \mathcal{S}^c)$ then
$FP \leftarrow FP + 1$
else
$TN \leftarrow TN + 1$
end if

True Positive Rate (TPR) = TP/(TP+FN)
False Positive Rate (FPR) = FP/(FP+TN)
Accuracy (ACC) = (TP+TN)/(TP+FN+FP+TN)
Specificity (SPC) = 1-FPR

where γ_k represent a correlation function defined by equation (6.17) and set $\mathcal{S}$ is defined by the region which is close to movement onset time, for example, $\pm 0.2sec.$ around movement onset time. In order to predict movement onset correctly, $\gamma_k \geq \Theta$ when $k \in \mathcal{S}$ and $\gamma_k < \Theta$ when $k \in \mathcal{S}^C$. In Figure 6.15, we can see that the overall prediction performance based on RP-RPCA is better than the performance based on RPCA. In order to choose proper



(a)



(b)

Figure 6.16: Trade-off for choosing threshold level: as we choose large threshold, we can reduce false positive rate but we lose the true positive rate as well; (a) jeev0617 (b) jeev0712.

threshold, trade-offs have to be made for reducing false positive rate while having reasonable true positive rate. Figure 6.16 shows a trade-off example of different thresholds, for example, if we increase threshold level, we can reduce false positive rate but we lose true positive rate as well.

Figure 6.17 shows the ROC curves for different monkeys or tasks where paco0716 rep-

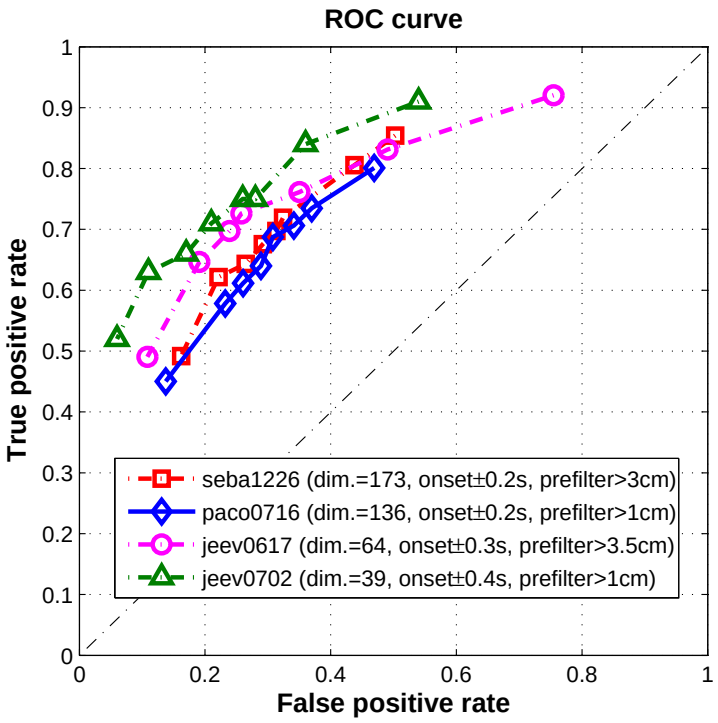


Figure 6.17: Receiver Operating Characteristic (ROC) curve for different monkeys or tasks where we prefiltered certain submovements with small amplitude in order to avoid artifacts of overfitting.

resents "center-out" task and seba1226 represents "random-pursuit" task which requires animal to reach to a (stationary) target shown anywhere in workspace (except near cursor's currently location) and then, hold it there for a target hold time.

Also, for the BMI application, once RP-RPCA had been run on the data, real-time submovement onset detection decisions would need to be based on matches between online data and the portion of the RP-RPCA template prior to submovement onset.

6.7 Conclusion

In this study, we analyze a neural activity dataset with different spatiotemporal perspectives in order to understand the basic neural response properties and extract neural signatures. First we identify discrete modes and demonstrate that the monkey brain operates in various discrete modes when controlling arm movements, using unsupervised segmentation of neural activity via sparse subspace clustering.

Second, we develop a new method, the combination of random projection and RPCA, for the neural data which is naturally sparse. This is the first application known to us of

RPCA to neuronal data. We apply the proposed method to neural spike data sets in order to validate the method and extract a low-rank elements of motor cortical activity. Such features represent common dynamic structures in the neural activity and can be used as a signature to detect motor primitives such as submovements. We explore the benefits of using Random Projection (RP) and show that using RP can improve the overall prediction performance.

Chapter 7

Conclusion

Even though many new techniques for identifying biological systems have been developed, they need to be able to incorporate different types of experimental data and varying levels of prior knowledge about the biological system. In order to reverse engineer biological systems based on prior knowledge and datasets, we discuss how to develop and choose a proper mathematical model, and identify an optimal type of experiment, and requirements on the data set in terms of quality and size as well as underlying assumptions. This thesis presents new mathematical tools which fill a gap between existing systems modeling tools in order to overcome the limitation of present mathematical models in systems biology.

The key novel ideas in this thesis are:

- a Hybrid Boolean Framework for modeling response of biological signal pathways is developed by taking advantage of existing mathematical tools and compensating for their limitation or disadvantages (Chapter 3);
- an optimization-based inference scheme is developed which is able to not only identify a model for temporally evolving biological networks but also capture how the gene regulatory network evolves over time, by combining dynamical system approach with a graph model (Chapter 4);
- a data-driven algorithm for identifying gene regulatory network is proposed which is able to reconstruct the graph structure exactly and suggest new experimental direction as well as reveal deficiencies in the model, based on compressive sensing (Chapter 5);
- new spatio-temporal perspectives in analyzing neural activity datasets are developed to understand the neural activities and extract neural signatures (Chapter 6).

In order to deal with applying these techniques to real biological systems, the construction of systematic data sets is needed since often, currently available data size and data quality make the application of these mathematical tools a challenge. Thus, we anticipate that the construction of systematic datasets can accelerate systems biology, which involve innovations that include the development of new experimental methods, the establishment of

degree of belief associated with individual measurements, and the protocol of self-consistent experiment.

The tools presented in this thesis have applications to engineering systems. For example, the Hybrid Boolean framework and optimization-based inference scheme have applications as we presented. On the other hand, the proposed data-driven approaches can be applied to diverse areas such as another biological system, social network, communication and engineering systems.

Moreover, the developed tools can also be used in conjunction with existing tools. As an example, RP-RPCA, which is developed for analyzing neural activities, can be applied to cluster and classify high dimensional microarray datasets of biological systems. In this context, the low-rank component represents the integrated regulatory signals through the common gene regulatory network across all different cell lines, and the sparse component might represent the signals corresponding to heterogeneity of mutant cell lines. Then, clustered datasets by using RP-RPCA might be useful for identifying graph structure, and the identified structure is applicable to building either a hybrid model or a mechanistic model.

These are other avenues for our future direction. Recently, the area of big data is particularly interesting to many communities, especially systems and control engineers. In order to make big data useful, we need to understand these data with a systems point of view and investigate how to handle them. These steps are necessary for being able to build a model and to understand underlying systems. Extending system approaches to large datasets will be challenging because of high dimension, complexity and the nature of the limitation of available knowledge. However, it seems promising because of the similarity in organization of dataset and systems with respect to biological systems.

In order to go beyond this, innovative ideas and open discussion will be required. Especially, a tight collaboration between biologists, medical doctors, neuroscientists, and engineers will be the key driver for fully understanding our systems.

Bibliography

[1] W. Abend, E. Bizzi, and P. Morasso. "Human arm trajectory formation". In: *Brain* 105.Pt 2 (1982), pp. 331–48.

[2] Lilia Alberghina and Hans V. Westerhoff. *Systems Biology: Definitions and Perspectives (Topics in Current Genetics)*. Springer, 2005.

[3] Bree B. Aldridge et al. "Physiochemical modelling of cell signalling pathways". In: *Nature Reviews Cell Biology* 8 (2006), pp. 1195–1203.

[4] Uri Alon. "Network motifs: theory and experimental approaches". In: *Nature Review Genetics* 8 (2007), pp. 450–461.

[5] Dhara N. Amin et al. "Resiliency and vulnerability in the HER2-HER3 tumorigenic driver". In: *Science Translational Medicine* 2.16 (2010).

[6] Jim Anderson. *http://www.nigms.nih.gov/Research/FeaturedPrograms/SysBio/*. (2003).

[7] Ery Arias-Castro, Emmanuel J. Candès, and Mark. A. Davenport. "On the fundamental limits of adaptive sensing". In: *IEEE Transactions on Information Theory* 59.1 (2013).

[8] Anil Aswani, Nicholas Boyd, and Claire J. Tomlin. "Graph-Theoretic topological control of biological genetic networks". In: *Proceedings of the American Control Conference (ACC)* (2009), pp. 1700–1705.

[9] S. Baasner et al. "Reversible tumorigenesis in mice by conditional expression of the HER2/c-erbB2 receptor tyrosine kinase". In: *Oncogene* 13.5 (1996), pp. 901–911.

[10] Marc Baily-Bechet et al. "Inference of sparse combinatorial-control networks from gene-expression data: a message passing approach". In: *BMC Bioinformatics* 11 (2010), p. 355.

[11] M. Balasubramanian et al. "The Isomap algorithm and topological stability". In: *Science* 290.5500 (2000), pp. 2319–2323.

[12] Mukesh Bansal, Giusy Della Gatta, and Diego di Bernardo. "Inference of gene regulatory networks and compound mode of action from time course gene expression profiles". In: *Bioinformatics* 22.7 (2006).

[13] Richard G. Baraniuk and Michael B. Wakin. "Random projections of smooth manifolds". In: *Journal of Foundations of Computational Mathematics* 9.1 (2009), pp. 51–77.

[14] J. Baselga et al. "Phase II study of weekly intravenous trastuzumab (Herceptin) in patients with HER2/neu-overexpressing metastatic breast cancer". In: *Journal of Clinical Oncology* 14 (1996), pp. 737–744.

[15] Gregory Batt et al. "Efficient parameter search for qualitative models of regulatory networks using symbolic model checking". In: *Bioinformatics* 26.18 (2010), pp. 603–610.

[16] D. Di Bernardo, T.S. Gardner, and J.J. Collins. "Robust identification of large genetic networks". In: *Pacific Symposium on Biocomputing* 9 (2004), pp. 486–497.

[17] Ella Bingham and Heikki Mannila. "Random projection in dimensionality reduction: applications to image and text data". In: *Proceedings of the KDD '01 Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (2001), pp. 245–250.

[18] Stefan Bornholdt. "Less is more in modeling large genetic networks". In: *Science* 310.5747 (2005), pp. 449–451.

[19] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[20] John Brognard and Alexandra C. Newton. "PHLiPPing the switch on Akt and protein kinase C signaling". In: *Trends in Endocrinology and Metabolism* 19.6 (2008), pp. 223–30.

[21] Vernon B. Brooks. "Motor programs revisited". In: *Posture and movement (ed. R.E. Talbott and D.R. Humphrey)* (1979), pp. 13–49.

[22] Harold J. Burstein, Anna Maria Storniolo, and Sandra X Franco. "A phase II, open-label, multicenter study of lapatinib in two cohorts of patients with advanced or metastatic breast cancer who have progressed while receiving trastuzumab-containing regimens". In: *Annals of Oncology* 15.27 (2004).

[23] Emmanuel Candès and Justin Romberg. "Sparsity and incoherence in compressive sampling". In: *Inverse Problems* 23 (2006), pp. 969–985.

[24] Emmanuel Candès and M. Wakin. "An introduction to compressive sampling". In: *IEEE Signal Processing Magazine* (2008), pp. 21–30.

[25] Emmanuel J. Candès and Paige A. Randall. "Highly robust error correction by convex programming". In: *IEEE Transaction on Information Theory* 54.7 (2008), pp. 2829–2840.

[26] Emmanuel J. Candès, Justin K. Romberg, and T. Tao. "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information". In: *IEEE Transaction on Information Theory* 52 (2004), pp. 489–509.

[27] Emmanuel J. Candès and Terence Tao. "Near-optimal signal recovery from random projections: universal encoding strategies?" In: *IEEE Transaction on Information Theory* 52 (2006), pp. 5406–5425.

[28] Emmanuel J. Candès et al. "Robust Principal Component Analysis?" In: *Journal of the ACM* 58.1 (2011), pp. 1–37.

[29] Jose Carmena, Claire Tomlin, and Masayoshi Tomizuka. *NSF EFRI-M3C Grant (Project Proposal, EFRI.* (2011).

[30] Jose M Carmena et al. "Learning to control a brain-machine interface for reaching and grasping by primates". In: *PLoS Biology* 1.2 (2003), pp. 193–208.

[31] Young Hwan Chang, Joe Gray, and Claire J. Tomlin. "Optimization-based inference for temporally evolving Boolean networks with applications in biology". In: *Proceedings of the American Control Conference (ACC)* (2011), pp. 4129–4134.

[32] Young Hwan Chang, Joe Gray, and Claire J. Tomlin. "Optimization-based inference for temporally evolving networks with applications in biology". In: *Journal of Computational Biology* 19.12 (2012), pp. 1307–1323.

[33] Young Hwan Chang and Claire Tomlin. *Exact reconstruction of gene regulatory networks using Compressive Sensing (in preparation).* (2013).

[34] Young Hwan Chang and Claire Tomlin. *Reconstruction of gene regulatory networks with hidden nodes (submitted).* (2013).

[35] Young Hwan Chang and Claire J. Tomlin. "Data-driven graph reconstruction using compressive sensing". In: *IEEE Conference on Decision and Control (CDC)* (2012), pp. 1035–1040.

[36] Young Hwan Chang and Claire J. Tomlin. "Inference of temporally evolving network dynamics with applications in biological systems". In: *Proceedings of the IEEE Conference on Decision and Control and European Control Conference (CDC)* (2011), pp. 3706–3711.

[37] Young Hwan Chang and Claire J. Tomlin. "Modeling response of biological signal pathways using a Hybrid Boolean Framework". In: *Proceedings of the ASME Dynamic Systems and Control Conference* (2012).

[38] Young Hwan Chang et al. *Low-rank representation of neural activity and detection of submovements (submitted).* (2013).

[39] William W. Chen et al. "Input-output behavior of ErbB signaling pathways as revealed by a mass action model trained against dynamic data". In: *Molecular Systems Biology* 5.239 (2009), pp. 1–19.

[40] Aniruddha Choudhury et al. "Small interfering RNA (siRNA) inhibits the expression of the Her2/neu gene, upregulates HLA class I and induces apoptosis of Her2/neu positive tumor cell lines". In: *International Journal of Cancer* 108.1 (2004), pp. 71–77.

[41] Mark M. Churchland and Krishna V. Shenoy. "Temporal complexity and heterogeneity of single-neuron Activity in premotor and motor cortex". In: *Journal of Neurophysiology* 97.6 (2007), pp. 4235–4257.

[42] Mark M. Churchland et al. "Neural population dynamics during reaching". In: *Nature* 487.7405 (2012), pp. 51–56.

[43] Mark M Churchland et al. "Stimulus onset quenches neural variability: a widespread cortical phenomenon". In: *Nature Neuroscience* 13 (2010), pp. 369–78.

[44] Mark M Churchland et al. "Techniques for extracting single-trial activity patterns from large-scale neural recordings". In: *Current Opinion in Neurobiology* 17.5 (2007), pp. 609–18.

[45] Raphael A. Clynes et al. "Inhibitory Fc receptors modulate in vivo cytoxicity against tumor targets". In: *Nature Medicine* 6 (2000), pp. 443–446.

[46] MIT Computational and Systems Biology Initiative. *CSBI Overview, http://csbi.mit.edu/overview/.* (2011).

[47] Nicholas G. Cooper, Calin A. Belta, and Agung Agung Julius. "Genetic regulatory network identification using multivariate monotone functions". In: *Proceedings of the IEEE conference on Decision and Control and European Control Conference (CDC-ECC)* (2011), pp. 2208–2213.

[48] Siddharth Dangi et al. "Adaptive Kalman filtering for closed-loop brain-machine interface systems". In: *Proceedings of the 5th International IEEE EMBS Conference on Neural Engineering, Cancun (Mexico)* (2011), pp. 609–612.

[49] Sanjoy Dasgupta. "Experiments with Random Projection". In: *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence* (2000), pp. 143–151.

[50] Andrea dÀvella and Emilio Bizzi. "Shared and specific muscle synergies in natural motor behaviors". In: *Proceedings of the National Academy of Science of the United States of America* 102.8 (2005), pp. 3076–3081.

[51] Andrea dÀvella, Philippe Saltiel, and Emilio Bizzi. "Combinations of muscle synergies in the construction of a natural motor behavior". In: *Nature Neuroscience* 6.3 (2003), pp. 300–308.

[52] Andrea dÀvella et al. "Control of fast-reaching movements by muscle synergy combinations". In: *The Journal of Neuroscience* 26.30 (2006), pp. 7791–7810.

[53] Andrea dÀvella et al. "Modulation of phasic and tonic muscle synergies with reaching direction and speed". In: *Journal of Neurophysiology* 100.3 (2008), pp. 1433–1454.

[54] Sampath Deegalla and Henrik Bostrom. "Reducing high-dimensional data by Principal Component Analysis vs. Random Projection for nearest neighbor classification". In: *Proceedings of the 5th International Conference on Machine Learning and Applications (ICMLA)* (2006), pp. 245–250.

[55] Micheal B. Eisen et al. "Cluster analysis and display of genome-wide expression patterns". In: *Proceedings of the National Academy of Science of the United States of America* 95.25 (1998), pp. 14863–14868.

[56] Ehsan Elhamifar and Reneè Vidal. "Sparse subspace clustering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 2790–2797.

[57] L.C. Evans. *An Introduction to Mathematical Optimal Control Theory*. UC Berkeley, http://math.berkeley.edu/∼evans/control.course.pdf.

[58] Timo Faltus et al. "Silencing of the HER2/neu gene by siRNA inhibits proliferation and induces apoptosis in HER2/neu-overexpressing breast cancer cells". In: *Neoplasia* 6.6 (2004), pp. 786–795.

[59] Alon Fishbach et al. "Kinematic properties of on-line error corrections in the monkey". In: *Experimental Brain Research* 164.4 (2005), pp. 442–57.

[60] Tamar Flash and Neville Hogan. "The coordination of arm movements: an experimentally confirmed mathematical model". In: *The Journal of Neuroscience* 5.7 (1985), pp. 1688–703.

[61] Nir Friedman and Daphne Koller. "Being Bayesian about Bayesian network structure: a Bayesian approach to structure discovery in Bayesian networks". In: *Machine Learning* 50.1-2 (2000), pp. 95–125.

[62] Tianyan Gao, Frank Furnari, and Alexandra C. Newton. "PHLPP: a phosphatase that directly dephosphorylates Akt, promotes apoptosis, and suppresses tumor growth". In: *Molecular Cell* 18.1 (2005), pp. 13–24.

[63] A.P. Georgopoulos et al. "Interruption of motor cortical discharge subserving aimed arm movements". In: *Experimental Brain Research* 49.3 (1983), pp. 327–340.

[64] Apostolos P. Georgopoulos, Ronald E. kettner, and Andrew B. Schwartz. "Primate motor cortex and free arm movements to visual targets in three-dimensional space. I. Relations between single cell discharge and direction of movement". In: *The Journal of Neuroscience* 8.8 (1988), pp. 2913–2927.

[65] Apostolos P. Georgopoulous et al. "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex". In: *The Journal of Neuroscience* 2.11 (1982), pp. 1527–1537.

[66] Wulfram Gerstner et al. "A neuronal learning rule for sub-millisecond temporal coding". In: *Nature* 383 (1996), pp. 76–78.

[67] Charles E. Geyer et al. "Lapatinib plus capecitabine for HER2-positive advanced breast cancer". In: *The New England Journal of Medicine* 355.26 (2007), pp. 2733–2743.

[68] Ronojoy Ghosh and Claire J. Tomlin. "Lateral inhibition through Delta-Notch signaling: A piecewise affine hybrid model". In: *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science* (2001), pp. 232–246.

[69] Ronojoy Ghosh and Claire J. Tomlin. "Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-Notch protein signalling". In: *IEE Systems Biology* 1.1 (2004), pp. 170–183.

[70] Jorge M. Gonçalves and Sean C. Warnick. "Necessary and sufficient conditions for dynamical structure reconstruction of LTI Networks". In: *IEEE Transactions on Automatic Control* 53.7 (2008), pp. 1670–1674.

[71] James Gordon, Maria Felice Ghilardi, and Claude Ghez. "Accuracy of planar reaching movements. I. Independence of direction and extent variability". In: *Experimental Brain Research* 99 (1994), pp. 97–111.

[72] Joe Gray et al. *Modeling resistance to HER2 targeted therapies (Project Proposal, ICBP.* (2009).

[73] Michael S. A. Graziano, Tyson N. S. Aflalo, and Dylan F. Cooke. "Arm movements evoked by electrical stimulation in the motor cortex of monkeys". In: *Journal of Neurophysiology* 94.6 (2005), pp. 4209–4223.

[74] Michael S.A Graziano, Charlotte S.R Taylor, and Tirin Moore. "Complex movements evoked by microstimulation of precentral cortex". In: *Neuron* 34.5 (2002), pp. 841–851.

[75] Iwata H et al. "Phase II clinical study of lapatinib (GW572016) in patients with advanced or metastatic breast cancer". In: *Breast Cancer Research and Treatment* (2006), 100, abstract 1091.

[76] Adam M. Halasz et al. "Analysis of Lactose metabolism in E.coli using reachability analysis of hybrid systems". In: *IET Systems Biology* 1.2 (2007), pp. 130–148.

[77] Soohee Han, Yeoin Yoon, and Kwang-Hyun Cho. "Inferring biomolecular interaction networks based on convex optimization". In: *Computational Biology and Chemistry* 31.5-6 (2007), pp. 347–354.

[78] Daniel Harari and Yosef Yarden. "Molecular mechanisms underlying ErbB2/HER2 action in breast cancer". In: *Oncogene* 19.53 (2000), pp. 6102–6114.

[79] Y. Hashambhoy and R. Vidal. "Recursive identification of switched ARX models with unknown number of models and unknown orders". In: *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference* (2005), pp. 6115–6121.

[80] Nicholas G. Hatsopoulos, Qingqing Xu, and Yali Amit. "Encoding of movement fragments in the motor cortex". In: *The Journal of Neuroscience* 27.19 (2007), pp. 5105–5114.

[81] Bryan T. Hennessy et al. "A technical assessment of the utility of Reverse Phase Protein Arrays for the study of the functional proteome in non-microdissected human breast cancers". In: *Clinical Proteomics* 6.4 (), pp. 129–151.

[82] A.L. Hodgkin. "The Croonian Lecture, ionic movements and electrical activity in giant nerve fibres". In: *Proceedings of the Royal Society of London. Series B, Biological Sciences* 148 (1958), pp. 1–37.

[83] Trey Ideker, Timothy Galitski, and Leroy Hood. "A new approach to decoding life: Systems Biology." In: *Annual Review of Genomics and Human Genetics* 2 (2001), pp. 343–372.

[84] Trey Ideker, L. Raimond Winslow, and A. Douglas Lauffenburger. "Bioengineering and systems biology". In: *Annals of Biomedical Engineering* 34.7 (2006), pp. 257–264.

[85] Trey Ideker et al. "Discovering regulatory and signaling circuits in molecular interaction networks". In: *Bioinfomatics* 18 (2002), pp. 233–240.

[86] Brian Ingalls. *Mathematical Modelling in Systems Biology: An Introduction*. (2012).

[87] Solomon Itani, Joe Gray, and Claire J Tomlin. "An ODE Model for the HER2/3-AKT Signaling pathway in cancers that overexpress HER2". In: *Proceedings of the American Control Conference (ACC)* (2010), pp. 1235–1241.

[88] Francois Jacob. *The logic of living systems*. 1st edition. London: Allen Lane, 1974.

[89] Kevin A. Janes and Michael B. Yaffe. "Data-driven modelling of signal-transduction networks". In: *Nature Reviews Molecular Cell Biology* 7 (2006), pp. 820–828.

[90] Hidde De Jong et al. "Qualitative simulation of genetic regulatory networks using piecewise-linear models". In: *Bulletin of Mathematical Biology* 66.2 (2001), pp. 301–340.

[91] Maryam Kamgarpour and Claire J. Tomlin. "Modeling and analysis of cell differentiation using hybrid systems". In: *Proceedings of the American Control Conference* (2010), pp. 4355–4360.

[92] William J. kargo and Douglas A. Nitz. "Early skill learning is expressed through selection and tuning of cortically represented muscle synergies". In: *The Journal of Neuroscience* 23.35 (2003), pp. 11255–11269.

[93] Steven W. Keele. "Movement control in skilled motor performance". In: *Psychological Bulletin* 70 (1968), pp. 387–403.

[94] Paul Kersey and Rolf Apweiler. "Linking publication, gene and protein data". In: *Nature Cell Biology* 8 (2006), pp. 1183–1189.

[95] Dongsan Kim, Yung-Keun Kwon, and Kwang-Hyun Cho. "Coupled positive and negative feedback circuits form an essential building block of cellular signaling pathway". In: *BioEssays* 29.1 (2007), pp. 85–90.

[96] Man-Sun Kim, Jeong-Rae Kim, and Kwang-Hyun Cho. "Dynamic network rewiring determines temporal regulatory functions in Drosophila melanogaster development processes". In: *Bioessays* 32.6 (2010), pp. 505–513.

[97] Hiroaki Kitano. "Computational systems biology". In: *Nature* 420 (2002), pp. 206–210.

[98] Daniel D. Lee and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401 (1999), pp. 788–791.

[99] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-Laszlo Barabasi. "Controllability of complex networks". In: *Nature* 473 (2011), pp. 167–173.

[100] Jacques Loeb. *The mechanistic concept of life.* reprinted 1964. Cambridge MA: Belknap Press, 1912.

[101] Michael A. Long, Dezhe Z. Jin, and Michale S. Fee. "Support for a synaptic chain model of neuronal sequence generation". In: *Nature* 468 (2010), pp. 394–399.

[102] Lan Ma and Pablo A Iglesias. "Quantifying robustness of biochemical network models". In: *BMC Bioinformatics* 3.38 (2002), pp. 1–13.

[103] Wasim Q. Malik et al. "Efficient decoding with steady-state Kalman filter in neural interface systems". In: *IEEE Transaction on Neural System and Rehabilitation Engineering* 19.1 (2011), pp. 25–34.

[104] Mark M. Moasser. "The oncogene HER2: its signaling and transforming functions and its role in human cancer pathogenesis". In: *Oncogene* 26 (2007), pp. 6469–87.

[105] Susan E. Moody et al. "Conditional activation of Neu in the mammary epithelium of transgenic mice results in reversible pulmonary metastasis". In: *Cancer Cell* 2 (2002), pp. 451–461.

[106] Melody K. Morris et al. "Logic-based models for the analysis of cell signaling networks". In: *Biochemistry* 49.15 (2010), pp. 3216–3224.

[107] Yadong Mu et al. "Accelerated low-rank visual recovery by random projection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2011), pp. 2609–2616.

[108] Antonino Musolino et al. "Immunoglobulin G fragment C receptor polymorphisms and clinical efficacy of trastuzumab-based therapy in patients with HER-2/neu-positive metastatic breast cancer". In: *Journal of Clinical Oncology* 26.11 (2008), pp. 1789–1796.

[109] H. Nakada, K. Takaba, and T. Katayama. "Identification of piecewise affine systems based on statistical clustering technique". In: *Automatica* 41.5 (2005), pp. 905–913.

[110] D. Napoletani et al. "Augmented sparse reconstruction of protein signaling networks". In: *Journal of Theoretical Biology* 255.1 (2008), pp. 40–52.

[111] Domenico Napoletani and Timothy D. Sauer. "Reconstructing the topology of sparsely connected dynamical networks". In: *Physical. Review. E* 77.2 (2008), p. 026103.

[112] Amy L. Orsborn et al. "Closed-loop decoder adaptation on intermediate time-scales facilitates rapid BMI performance improvements independent of decoder initialization conditions". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 20.4 (2012), pp. 468–477.

[113] Simon A. Overduin, Andrea dÀvella, and Emilio Bizzi. "Does macaque motor cortex reflect recruitment of muscle synergies?" In: *Forum of European Neuroscience 2010 Satellite Symposium on Motor Control, Nijmegen, Holland.* (2010).

[114] Simon A. Overduin et al. *Cortical representation of motor programs in the primate brain (in preparation).*

[115] Simon A. Overduin et al. "Detection of submovement primitives for neuroprosthetic motor control". In: *Society for Neuroscience (584.07/SS6)* (2012).

[116] Simon A. Overduin et al. "Hand movements and muscle activation evoked by intracortical microstimulation". In: *Society for Neuroscience 36, San Diego CA* (2010).

[117] Simon A. Overduin et al. "Modulation of muscle synergy recruitment in primate grasping". In: *The Journal of Neuroscience* 28.4 (2008), pp. 880–892.

[118] Liam Paninski et al. "Superlinear population encoding of dynamic hand trajectory in primary motor cortex". In: *The Journal of Neuroscience* 24.39 (2004), pp. 8551–8561.

[119] Simone Paoletti et al. "Identification of hybrid systems: a tutorial". In: *European Journal of Control* 13.2 (2007), pp. 242–260.

[120] Itai Pashtan et al. "Targeting Hsp90 prevents escape of breast cancer cells from tyrosine kinase inhibition". In: *Cell Cycle* 7.18 (2008), pp. 2936–2941.

[121] A. Polit and E. Bizzi. "Characteristics of motor programs underlying arm movements in monkeys". In: *Journal of Neurophysiology* 42.1 (1979), pp. 183–194.

[122] Riccardo Porreca et al. "Structural identification of piecewise-linear models of genetic regulatory networks". In: *Journal of Computational Biology* 15.10 (2008), pp. 1365–1380.

[123] J. Ragot, G. Mourot, and D. Maquin. "Parameter estimation of switching piecewise linear systems". In: *Proceedings of the 42nd IEEE Conference on Decision and Control* (2003), pp. 5783–5788.

[124] Dhakshin Ramanathan, James M. Conner, and Mark H. Tuszynski. "A form of motor cortical plasticity that correlates with recovery of function after brain injury". In: *Proceedings of the National Academy of Science of the United States of America* 103.30 (2006), pp. 11370–11375.

[125] Brandon Rohrer and Neville Hogan. "Avoiding spurious submovement decompositions: A globally optimal algorithm". In: *Biology Cybernetics* 89.3 (2003), pp. 190–199.

[126] Brandon Rohrer and Neville Hogan. "Avoiding spurious submovement decompositions II: A scattershot algorithm". In: *Biology Cybernetics* 94.5 (2006), pp. 409–14.

[127] Karen Sachs et al. "Bayesian network approach to cell signaling pathway modeling". In: *Science Signaling The Signal Transduction Knowledge Environment* 2002.148 (2002), p. 38.

[128] Maryam Saleh, Kazutaka Takahashi, and Nocholas G. Hatsopoulos. "Encoding of coordinated grasp trajectories in primary motor cortex". In: *The Journal of Neuroscience* 32.4 (2010), pp. 1220–1232.

[129] Regina Samaga et al. "The logic of EGFR/ErbB signaling: theoretical properties and analysis of high-throughput data". In: *PLoS Comput Biol* 5.8 (Aug. 2009), e1000438.

[130] Gopal Santhanam et al. "Factor-analysis methods for higher-performance neural prostheses". In: *Journal of Neurophysiology* 102.2 (2009), pp. 1315–30.

[131] Uwe Sauer, Matthias Heinemann, and Nicola Zamboni. "Genetics: getting closer to the whole picture". In: *Science* 316.5824 (2007), pp. 550–551.

[132] Marc A Schaub, Thomas A Henzinger, and Jasmin Fisher. "Qualitative networks: a symbolic approach to analyze biological signaling networks". In: *BMC Systems Biology* 1.4 (2007).

[133] Ilka B. Schiffer et al. "Switching off HER-2/neu in a tetracycline-controlled mouse tumor model leads to apoptosis and tumor-size-dependent remission". In: *Cancer Research* 63 (2003), pp. 7221–7231.

[134] Henning Schmidt, Kwang-Hyun Cho, and Elling W. Jacobsen. "Identification of small scale biochemical networks based on general type system perturbations". In: *Federation of European Biochemical Societies (FEBS) Journal* 272.9 (2005), pp. 2141–2151.

[135] Henning Schmidt and Elling W. Jacobsen. "Linear systems approach to analysis of complex dynamic behaviors in biochemical networks". In: *Systems Biology (Stevenage)* 1.1 (2004), pp. 149–158.

[136] Natalia V. Sergina and Mark M. Moasser. "The HER family and cancer: emerging molecular mechanisms and therapeutic targets". In: *Trends in Molecular Medicine* 13.12 (2007), pp. 527–34.

[137] Natalia V. Sergina et al. "Escape from HER-family tyrosine kinase inhibitor therapy by the kinase-inactive HER3". In: *Nature* 445.25 (2007), pp. 437–441.

[138] Dennis J. Slamon et al. "Human breast cancer: correlation of relapse and survival with amplification of the HER-2/neu oncogene". In: *Science* 235 (1987), pp. 177–182.

[139] Dennis J. Slamon et al. "Use of chemotherapy plus a monoclonal antibody against HER2 for metastatic breast cancer that overexpresses HER2". In: *The New England Journal of Medicine* 344 (2001), pp. 783–792.

[140] Sen Song, Keneth d. Miller, and L. F. Abbott. "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity". In: *Nature neuroscience* 3.9 (2000), pp. 919–926.

[141] Eduardo Sontag, Anatoly Kiyatkin, and Boris N. Kholodenko. "Inferring dynamics architecture of cellular network using time series of gene expression, protein and metabolite data". In: *Bioinformatics* 20.12 (2004), pp. 1877–1886.

[142] Eduardo Sontag et al. "The effect of negative feedback loops on the dynamics of boolean networks". In: *Biophysical Journal* 95.2 (2008), pp. 518–526.

[143] Eduardo D. Sontag. "Network reconstruction based on steady-state data". In: *Essays in Biochemistry* 45 (2008), pp. 161–176.

[144] Eran Stark, Rotem Drori, and Moshe Abeles. "Partial cross-correlation analysis resolves ambiguity in the encoding of multiple movement features". In: *Journal of Neurophysiology* 95.3 (2006), pp. 1966–1975.

[145] Eran Stark et al. "Distinct movement parameters are represented by different neurons in the motor cortex". In: *European Journal of Neuroscience* 26.4 (2007), pp. 1055–1066.

[146] Jason R. Swedlow, Suzanna E. Lewis, and Ilya G. Goldberg. "Modelling data across labs, genomes, space and time". In: *Nature Cell Biology* 8 (2006), pp. 1190–1194.

[147] Terence Tao. *Compressed sensing (Or: the equation $Ax = b$, revisited).*

[148] Joseph Teran et al. "Creating and simulating skeletal muscle from the visible human data set". In: *IEEE Transactions on visualization and Computer Graphics* 11.3 (2005), pp. 317–328.

[149] Claire J. Tomlin, George J. Pappas, and Shankar S. Sastry. "Conflict resolution for air traffic management: A study in multiagent hybrid systems". In: *IEEE Transactions on Automatic Control* 43.4 (1998), pp. 509–521.

[150] Claire J. Tomlin et al. "Computational techniques for the verification of hybrid systems". In: *Proceedings of the IEEE* 91.7 (2003), pp. 986–1001.

[151] Anthony Trewavas. "A brief history of systems biology: "Every object that biology studies is a system of systems." Francois Jacob (1974)". In: *The Plant Cell* 18.10 (2006), pp. 2420–2430.

[152] Y. Uno, M. Kawato, and R. Suzuki. "Formation and control of optimal trajectory in human multijoint arm movement. Minimum torque-change model". In: *Biological Cybernetics* 61 (1989), pp. 89–101.

[153] Domitilla Del Vecchio, Richard M. Murray, and Pietro Perona. "Decomposition of human motion into dynamics based primitives with application to drawing tasks". In: *Automatica* 39.12 (2003), pp. 2085–2098.

[154] Ramana Vinjamuri et al. "Dimensionality reduction in control and coordination of the human hand". In: *Proceedings of the IEEE Transactions on Biomedical Engineering* 57.2 (2010), pp. 284–295.

[155] Ramana K. Vinjamuri et al. "Time-varying synergies in velocity profiles of finger joints of the hand during reach and grasp". In: *Proceedings of the IEEE Engineering in Medicine and Biology Society* (2007), pp. 4846–4849.

[156] James W. Ward. "The influence of posture on responses elicitable from the cortex cerebri of cats". In: *Journal of Neurophysiology* 1.5 (1938), pp. 463–475.

[157] Johan Wessberg and Miguel A.L. Nicolelis. "Optimizing a linear algorithm for real-time robotic control using chronic cortical ensemble recordings in monkeys". In: *Journal of Cognitive Neuroscience* 16.6 (2004), pp. 1022–35.

[158] Johan Wessberg et al. "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates". In: *Nature* 408.6810 (2000), pp. 361–365.

[159] Hans V. Westerhoff and Lilia Alberghina. *Systems Biology: Did we know it all along? (Topics in Current Genetics)*. Vol. 13. Springer, 2005, pp. 3–9.

[160] H Steven Wiley, Stanislav Y Shvartsman, and Douglas A Lauffenburger. "Computational modeling of the EGF-receptor system: a paradigm for systems biology". In: *Trends in Cell Biology* 13.1 (2003), pp. 43–50.

[161] Ned Wingreen and David Botstein. "Back to the future: education for systems-level biologist". In: *Nature Reviews Molecular Cell Biology* 7 (2006), pp. 829–832.

[162] Christopher R. Wren and Alex P. Pentland. "Dynamic models of human motion". In: *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition, Nara, Japan* (1998), pp. 22–27.

[163] Boyan Yordanov and Calin A. Belta. "Formal analysis of piecewise affine systems under parameter uncertainty with application to gene networks". In: *Proceedings of the American Control Conference (ACC)* (2008), pp. 2767–2772.

[164] Chang hun You, Lawrence B. Holder, and Diane J. Cook. "Learning patterns in the dynamics of biological networks". In: *Proceedings of 15th International Conference on Knowledge Discovery and Data Mining* (2009), pp. 977–986.

[165] Byron M. Yu et al. "Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity". In: *Journal of Neurophysiology* 102.1 (2009), pp. 614–35.

[166] Jing Yu et al. "Advances to Bayesian network inference for generating causal networks from observational biological data". In: *Bioinformatics* 20.18 (2004), pp. 3594–3603.

[167] Ye Yuan et al. "Robust dynamical network structure reconstruction". In: *Automatica* 47 (2011), pp. 1230–1235.

[168] Michael M. Zavlanos et al. "Identification of stable genetic networks using convex programming". In: *Proceedings of the American Control Conference (ACC)* (2008), pp. 2755–2760.

[169] Christoph Zechnera et al. "Moment-Based Inference Predicts Bimodality in Transient Gene Expression". In: *Proceedings of the National Academy of Science of the United States of America* 109.21 (2012), pp. 8340–8345.

[170] Tianyi Zhou and Dacheng Tao. *Bilateral Random Projections (arXiv:1112.5215).* (2011).

[171] Yi Ming Zou. "Modeling and analyzing complex biological networks incooperating experimental information on both network topology and stable states". In: *Bioinformatics* 26.16 (2010), pp. 2037–2041.

Appendix A

Nonlinear Dimension Reduction

A common task in the analysis of large microarray datasets is sample classification based on gene expression patterns. This process can be divided into two steps: class prediction and class discovery. For example, when gene expression arrays are used for cancer cell classification, class prediction assigns tumor samples into pre-existing groups of malignancies, while class discovery reveals previously unknown cancer subtypes. The newly discovered tumor subtypes may have different clinical patterns, respond differently to certain drugs, and require more or less aggressive surgical and radiological treatment. Class discovery may also reveal previously unknown processes in cancer biology and define more specific indications for certain drugs. For instance, specific drugs may be used to target newly discovered tumor subtypes, thus facilitating pharmacogenomic drug design and development. These goals will soon become achievable with the results from microarray studies using large sample.

A variety of methods for Nonlinear Dimension Reduction (NDR) have been developed over the years. Isometric feature mapping (Isomap), locally linear embedding and Laplacian eigenmap are typical examples. The basic principles underlying these methods are, however, very similar.

They first speculate a local neighborhood structure around each datum to form a global neighborhood graph superimposed on the whole data set and then calculate the embeddings of the data set by preserving as much as possible the invariance of all neighborhood structures in the low-dimensional representation space. The neighborhood structure of a datum can be, for example, the isometric frame of its neighbors, the local coefficients that reconstruct the datum from its neighbors, or the locality relationship between the datum and its neighbors.

The most common methods to determine the neighbors of a datum are the k-nearest neighbor (NN) and ϵ-NN methods, defining neighbors as its k nearest ones and the ones whose distances from the datum are smaller than the threshold ϵ respectively. The performance of the NDR techniques highly depends on the choice of the neighborhood size k or ϵ or method. Here, we adopt Isomap rather than other NDR methods to generate the data embeddings of each cluster. This is due to the fact that intrinsic properties of Isomap is in concordance with the requirements of classifying large microarray datasets.

The Isomap method consists of three steps:

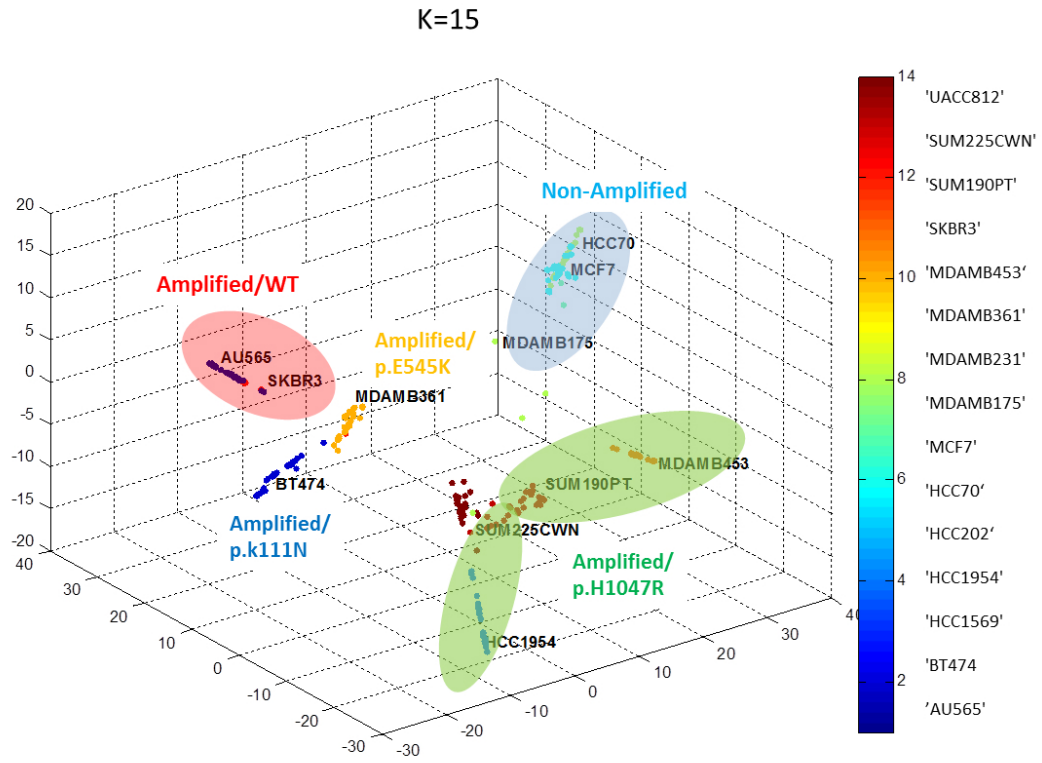


Figure A.1: 3-Dimensional representation of classification of RPPA dataset (K=15).

- constructing the neighborhood graph of the data set based on Euclidean distance;
- computing the distance of the shortest paths between data pairs in the graph;
- taking the lengths as the approximate interpoint geodesic distances constituting the geodesic distance matrix of the whole data set on which multidimensional scaling (MDS) is applied to construct the global low-dimensional embeddings.

Figure 1.3 shows Isomap result of RPPA data set. Isomap gives successful classification among different cell lines by cell line characteristics. For example, SKBR3 and AU565 cell lines are clustered together because both cell lines are nearly identical at the genetic and transcriptomic levels as well as at the level of HER family proteins. Figure A.1 and A.2 represent another view of classification results.

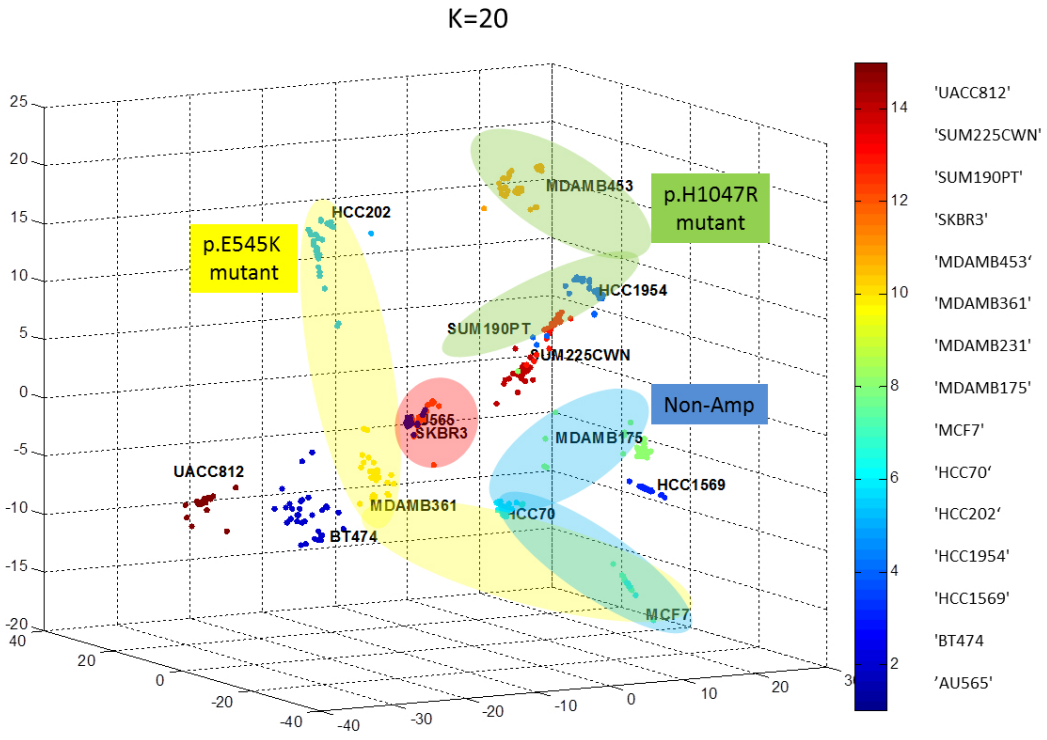


Figure A.2: 3-Dimensional representation of classification of RPPA dataset (K=20).

Appendix B

Proofs for equation (4.30)

proof :

(for convenience, we will use abbreviated notations without $(x), (t)$)

$$\begin{aligned} & \int_0^{t_f} \frac{d}{dt}\{X^T P X\} dt \\ = \; & X(t_f)^T P(t_f) X(t_f) - X(0)^T P(0) X(0) = \int_0^{t_f} \{\dot{X}^T P X + X^T \dot{P} X + X^T P \dot{X}\} dt \\ = \; & \int_0^{t_f} \{(\mathcal{A}X + W + V)^T P X + X^T \dot{P} X + X^T P(\mathcal{A}X + W + V)\} dt \end{aligned}$$

Select P which satisfies following equation:

$$\mathcal{A}^T P + \frac{dP}{dt} + P\mathcal{A} = -\mathcal{Q} + P^T \mathcal{R}^+ P, P(t_f) = \mathcal{S}$$

Using P, we can reformulate (4.30) as follows:

$$\begin{aligned} 0 \;=\; & -\frac{1}{2} X(t_f)^T \mathcal{S} X(t_f) + \frac{1}{2} X(0)^T P(0) X(0) + \frac{1}{2} \int_0^{t_f} \{X^T(-\mathcal{Q} + P^T \mathcal{R}^+ P) X \\ & + (W+V)^T P X + X^T P (W+V)\} dt \end{aligned}$$

Then, we can combine the cost function (4.26) and above equation as follows:

$$J \;=\; \frac{1}{2} X(0)^T P(0) X(0) + \int_0^{t_f} \{X^T P^T \mathcal{R}^+ P X + (W+V)^T P X + X^T P (W+V) + V^T \mathcal{R} V\} dt$$

We have the following relation because of the specific structure of $\mathcal{R}, V$ and W as follows:

$$\begin{aligned} V^T \mathcal{R} V \;=\; & \begin{bmatrix} 0_{1\times n} & v(t)^T \end{bmatrix} \begin{bmatrix} 0_{n\times n} & 0_{n\times k} \\ 0_{k\times n} & R_{k\times k} \end{bmatrix} \begin{bmatrix} 0_{n\times 1} \\ v(t) \end{bmatrix} = \begin{bmatrix} -\dot{x}_d(t)^T & v(t)^T \end{bmatrix} \begin{bmatrix} 0_{n\times n} & 0_{n\times k} \\ 0_{k\times n} & R_{k\times k} \end{bmatrix} \begin{bmatrix} -\dot{x}_d(t) \\ v(t) \end{bmatrix} \\ = \; & (W+V)^T \mathcal{R} (W+V) \end{aligned}$$

Using this relation, we can reformulate J as follows:

$$\begin{aligned} J &= \frac{1}{2}X(0)^T P(0)X(0) + \int_0^{t_f} \{X^T P^T \mathcal{R}^+ PX + (W+V)^T PX + X^T P(W+V) \\ &\quad +(W+V)^T \mathcal{R}(W+V)\}dt \\ &= \frac{1}{2}X(0)^T P(0)X(0) + \int_0^{t_f} \{(\mathcal{R}^+ PX + W + V)^T \mathcal{R}(\mathcal{R}^+ PX + W + V)\}dt \end{aligned}$$

To minimize J, $V_{opt} = -\mathcal{R}^+ PX - W$ but as we defined the structure of V and W in equation (4.27), V only satisfies the following condition: $V^* = -\mathcal{R}^+ PX$. However, when we plug in V^*, the optimal cost is $J^* = \frac{1}{2}X(0)^T P(0)X(0)$ which is the same as using V_{opt} since $W^T \mathcal{R} W = 0$.

Appendix C

Analysis of l_1 and l_2 regularization

1) $||\cdot||_{l_1}$ optimization vs. CS:

l_1 regularization has many of the beneficial properties of l_2 regularization, but yields sparse models which are more easily interpreted. Also, the model or graph structure produced under an l_1 penalty often outperform those produced with an l_2 penalty. It provides a regularized feature selection method, and can give a low variance feature selection.

$$\begin{aligned} & \min \mathbf{1}^T|A|\mathbf{1} \\ s.t. \quad & Y^k = AX^k \qquad k = 1, ..., M \end{aligned}$$

where k is a time step index, M is the number of measurements and $X^k, Y^k \in \mathbb{R}^n$. The Lagrangian associated with above equation is given by

$$\mathcal{L}(A, \Lambda) = \mathbf{1}^T|A|\mathbf{1} + \sum_{k=1}^{M} \lambda^{kT}(Y^k - AX^k)$$

where $\Lambda = \{\lambda^1, ..., \lambda^M\} \in \mathbb{R}^{n\times M}, \lambda^i \in \mathbb{R}^n$ is the Lagrange multiplier.

$$\mathcal{L}(A, \Lambda) = \sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}| + \sum_{k=1}^{M}\sum_{i=1}^{n}\lambda_i^k(y_i^k - a_i X^k) = \sum_{i=1}^{n}\sum_{j=1}^{n}|a_{ij}| + \sum_{k=1}^{M}\sum_{i=1}^{n}\lambda_i^k(y_i^k - \sum_{j=1}^{n}a_{ij}x_j^k)$$

where a_i is the ith row of matrix A, $Y^k = \left[y_1^k, y_2^k, ..., y_n^k\right]^T$ and $X^k = \left[x_1^k, x_2^k, ..., x_n^k\right]^T$. Its minimizer becomes

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial a_{pq}} &= sgn(a_{pq}) + \frac{\partial}{\partial a_{pq}}\left(\sum_{k=1}^{M}\lambda_p^k(y_p^k - \sum_{j=1}^{n}a_{pj}x_j^k)\right) \\ &= sgn(a_{pq}) + \sum_{k=1}^{M}\lambda_p^k\frac{\partial}{\partial a_{pq}}\left(-a_{pq}x_q^k\right) = sgn(a_{pq}) + \sum_{k=1}^{M}\lambda_p^k(-x_q^k) \\ \frac{\partial \mathcal{L}}{\partial \lambda_p^r} &= y_p^r - \sum_{j=1}^{n}a_{pj}x_j^r \end{aligned}$$

where $1 \leq p, q \leq n$ and $1 \leq r \leq M$. The optimality condition is given by

$$\frac{\partial \mathcal{L}}{\partial a_{pq}} = 0 \quad : \quad \sum_{k=1}^{M} x_q^k \lambda_p^{k*} = sgn(a_{pq}^*)$$
$$\frac{\partial \mathcal{L}}{\partial \lambda_p^r} = 0 \quad : \quad y_p^r = \sum_{j=1}^{n} a_{pj}^* x_j^r$$

Therefore, if the number of time points is bigger than the number of states (i.e., $M > n$, if we can measure many time points or use several data sets, this condition can be easily satisfied), we might have infeasible solution:

$$\begin{bmatrix} sgn(a_{11}^*) \\ sgn(a_{12}^*) \\ ... \\ sgn(a_{1n}^*) \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & ... & x_1^M \\ x_2^1 & x_2^2 & ... & x_2^M \\ ... & ... & ... & ... \\ x_n^1 & x_n^2 & ... & x_n^M \end{bmatrix} \begin{bmatrix} \lambda_1^1 \\ \lambda_1^2 \\ ... \\ \lambda_1^M \end{bmatrix}$$

For example, λ_1^k can be zero. Obviously, an $n \times M$ matrix has at most rank n so some of $\lambda_1^k, k = 1, ..., M$ can be zero, which means some of the constraints do not have to be satisfied (infeasible solution). This is basically l_1 optimization for overdetermined systems of equations. Thus, in order to guarantee feasibility for large $M > n$, we have to solve the following problem:

$$\min \mathbf{1}^T |A| \mathbf{1} + \gamma \sum_{k=1}^{M} ||Y^k - AX^k||_2$$

where γ can be used as a tuning parameter which handles the trade-off between sparsity and data fitting. On the other hand, CS uses the incoherent matrix for the sensing matrix which basically spreads the norm of the representation on the whole set of parameters.

2) $||\cdot||_{l_2}$ optimization vs. CS:

While l_2 regularization is an effective mean of achieving numerical stability, it does not encourage sparsity because the l_2 penalty forces the coefficients to be more similar to each other in order to minimize their joint 2-norm.

$$\begin{aligned} & \min ||A||_{l_2} \\ s.t. \quad & Y^k = AX^k \qquad k = 1, ..., M \end{aligned}$$

Without loss of generality, the above optimization problem is exactly the same as:

$$\min ||A||_{l_2}^2 \quad \text{s.t.} \quad Y^k = AX^k \quad k = 1, ..., M$$

By using similar approach for l_1 optimization problem, the optimality condition is given by

$$\frac{\partial \mathcal{L}}{\partial a_{pq}} = 0 \quad : \quad \sum_{k=1}^{M} x_q^k \lambda_p^{k*} = 2|a_{pq}^*|$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_p^r} = 0 \quad : \quad y_p^r = \sum_{j=1}^{n} a_{pj}^* x_j^r$$