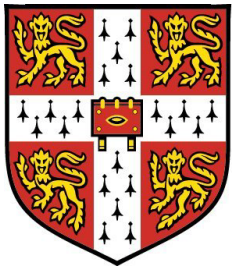
Decentralised Network Prediction and Reconstruction Algorithms



Ye Yuan

Control Group
Department of Engineering
Pembroke College

University of Cambridge

A thesis submitted for the degree of

Doctor of Philosophy

January, 2012

To My Parents.

"The fear of the LORD is the beginning of wisdom."

–Proverbs 9:10

Declaration

As required under the University's regulations, I hereby declare that this dissertation is not substantially the same as any that I have submitted or will be submitting for a degree or diploma or other qualification at this or any other university. Furthermore, this dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specified explicitly in the text. I also declare that the length of this dissertation is less than 65,000 words and that the number of figures is less than 150.

Ye Yuan
Control Group
Department of Engineering
University of Cambridge
30-1-2012

Acknowledgements

First and foremost, I wish to deeply thank my supervisor Dr. Jorge Gonçalves, for his constant encouragement and guidance on my research and also for the role model that he has set for my daily life. I sincerely appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience joyful, productive and stimulating.

I also wish to thank all the people in the Control Group Cambridge, Prof. Keith Glover, Prof. Jan Maciejowski, Prof. Malcolm Smith, Prof. Zoubin Ghahramani, Dr. Alex Webb, Dr. Neil Dalchau, Dr. Andrew Phillips, Dr. Hillel Kugler (Microsoft Research Cambridge) for their inputs and help in my research.

During the past years, I travelled weekly to Imperial College London as a visiting researcher. I wish to thank Dr. Guy-Bart Stan, Prof. Mauricio Barahona, Dr. Ze Zhang, Mr. Wei Pan, Dr. Wei Dai, Miss Neave O'Clery, Prof. Zidong Wang (Brunel) for providing valuable discussions, giving different views on my research and teaching me much. I really appreciate their constant contribution especially from Prof. Mauricio Barahona and Dr. Guy-Bart Stan.

In my second year, I was lucky enough to have the opportunity to visit Balling Lab at the Luxembourg Centre of System Biomedicine and Murray Lab at Caltech. I wish to thank Prof. Rudi Balling, Mr. Christophe Trefois and Dr. Seth Davis (Max Planck) for their collaborations.

During the seven months at Caltech, I was given the chance to learn much that I have never thought of (including doing biological experiments), thanks to discussion and help from Prof. Richard Murray, Mr. Enoch Yeung, Dr. Jun Liu, Dr. Jongmin Kim, Dr. Necmiye Ozay, Prof. Linlin Ou (ZJUT), Prof. Henrik Sandberg (KTH) and Dr. Brian Munsky (LANL). I also thank Prof. Reza Olfati-Saber (Dartmouth), Prof. John Doyle (Caltech) and Prof. Alexandre Megreski (MIT) for discussions and sharing ideas. I am also grateful for the collaborations with Miss Ni Ji and Prof. Alexander van Oudenaarder at MIT.

I wish to thank my research supervisors in my undergraduate, Prof. Yugeng Xi, Dr.

Xiaoli Li (SJTU), Prof. Yasamin Mostofi and Prof. Herbert Tanner (UNM).

I also wish to thank my long-term collaborators Prof. Ling Shi (HKUST), Prof. Sean Warnick (BYU) for always providing valuable discussion.

This research could only be conducted with the generous financial support from Dorothy-Hodgkin Trust, Microsoft Research Cambridge, Cambridge Overseas Trust, EPSRC, Henry Lester Trust, Pembroke College, IEEE Control and System Society.

I wish to thank people from Cambridge Chinese Christian Church, Glory Crown Assembly of God Pasadena for letting me know about the meaning of my life. The fear of the LORD is indeed the beginning of wisdom.

Last but not the least, my thanks would go to my great parents and lovely girlfriend Xin He for their loving considerations and great confidence in me all through the past two years.

Ye Yuan
Cambridge, England
On my 25th Birthday Eve

Contents

List of Figures

LIST OF FIGURES

List of Tables

Abstract

This study concerns the decentralised prediction and reconstruction problems in a network.

First of all, we propose a decentralised prediction algorithm in the framework of network consensus problem. It allows any individual to compute the consensus value of the whole network in finite time using only the minimal number of successive values of its own history. We further prove that the minimal number of steps can be characterised using other algebraic and graph theoretical notions: minimal external equitable partition (mEEP) that can be directly computed from the Laplacian matrix of the graph and from the underlying network structure. Later, we consider a number of possible theoretical extensions of the proposed algorithm to issues arising from practical applications, e.g., time-delays, noise, external inputs, nonlinearities in the network, and analyse how the proposed algorithm should be changed to incorporate such constraints.

For the decentralised reconstruction problem, we firstly define a new presentation: dynamical structure functions encoding structural information and explore the properties of such a representation for the purpose of solving the reconstruction problem. We have studied a number of theoretical problems: identification, realisation, reduction, etc. for dynamical structure functions and showed that how these theoretical can be used in solving decentralised network reconstruction problems. We later illustrate the results on a number of in-silico examples.

We conclude the thesis with some ideas and future perspectives to continue based on the research of decentralised prediction and reconstruction problems.

Chapter 1

Introduction

"To see a world in a grain of sand,
And a heaven in a wild flower,
Hold infinity in the palm of your hand,
And eternity in an hour."

This is the first four lines of a beautiful poem, "Auguries of Innocence" by William Blake. From the angle of a scientist, the first two lines have the illusion that every form of life is in a harmonious and designed order in nature, and even the smallest grain codes the whole of the universe. This stimulates that, even in studying the simplest object we have the chance to decode the information of the whole universe. The last two lines indicate that every hour in our life contains enough information about the past, present, and future. Sometimes, we can predict the future from observations of the past.

Inspired by this brilliant four lines, the first part of my Ph.D. research is to illustrate the idea how to predict the future using past observations (Chapter 2) and the second part is to illustrate the idea how to use local information to infer global properties (Chapter 3).

1.1 Networks

Networks are everywhere. We watch TV through the television network; we interact with each other in a closely connected social network; our body itself is a highly complicated biological network.

What is a network? It is in general hard to give a precise definition. Mathematically, a network can be thought as a collection of nodes that represent some physical quantities and edges that interconnect different nodes. As a simple example, a social network is

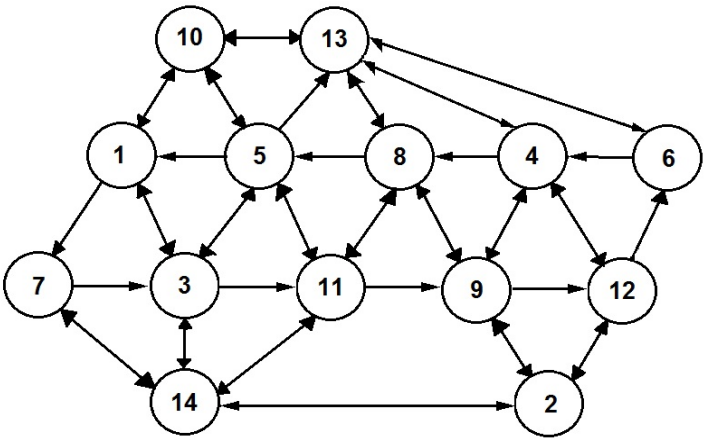


FIGURE 1.1: Underlying network structure of the considered graph consisted of 14 nodes.

a social structure made up of individuals called nodes, which are tied by one or more specific types of interdependency, such as friendship. Such ties can be also viewed as edges in the graphical representation.

We use graph-theoretical notations to model mathematical networks. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of nodes, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges, and $W = \{W[i,j]\}_{i,j=1,\ldots,n}$ is the corresponding n by n adjacency matrix, with $W[i,j] = 1$ when there is a link from j to i, and $W[i,j] = 0$ when there is no link from j to i. We can define the corresponding Laplacian matrix, $L \in \mathbb{R}^{n \times n}$ as $L[i,i] = \sum_{l \neq i}^{n} W[i,l]$, $\forall i = 1, \ldots, n$ and $L[i,j] = -W[i,j]$, $\forall i \neq j$.

For example, from the network structure in Fig. 1.1, we can see that the corresponding adjacency matrix (assuming unweighted) is

$$W = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

We can then obtain the following Laplacian matrix

$$L = \begin{bmatrix} 3 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 \\ -1 & 0 & 4 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & 0 & 4 & 0 & 0 & -1 \\ 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & 5 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 4 \end{bmatrix}.$$

From the above example, we can see that there is a strong link between the graph-theoretical notation of a network and an algebraic notation, i.e. the adjacency and Laplacian matrices. This link opens the gate for mathematical analysis of networks.

1.2 Control theory

Control theory mainly deals with the behaviour of dynamic systems. Block diagrams as in Fig 1.2 are typically used to describe the control problem as a feedback **system**. It

contains three elements: a plant, a feedback element and a control element.

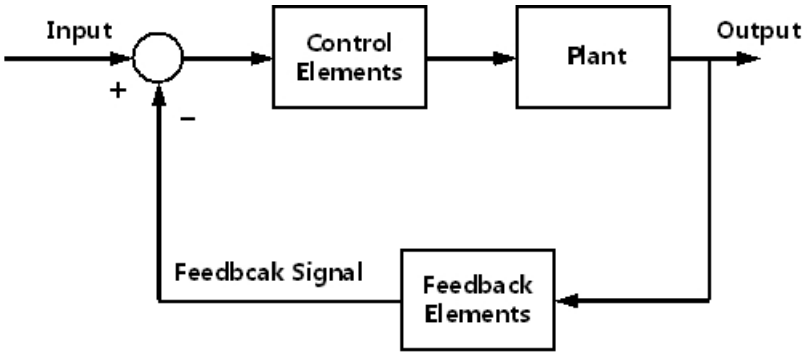


FIGURE 1.2: Block diagram to illustrate the framework of control theory.

Two of the most important problems in control theory are the following.

Firstly, **identification**[1], i.e. to understand and obtain a mathematical characterisation of the plant. System identification theory [51] uses statistical methods to build mathematical models of dynamic systems from measured data. This identification process is normally the first and the most important step, since it provides a mathematical model to characterise the behaviour of the plant in question for the later use, control.

Secondly, **control**[2], i.e. to set up the control element in Fig. 1.2 to manipulate the input(s) of a system and obtain the desired effect on the output(s) of the system. This control process aims to have the feedback system achieve specific goals or objectives, e.g., stability.

A key concept and mechanism in control is **feedback**, i.e. a causal path that leads from the initial generation of the feedback signal to the subsequent modification of the input signal. There are several advantages using feedback; among which the most important one is to reduce the effect from unknown disturbance to the output.

1.3 Networked identification/control problems

Given the network structure, one may consider the node in network as the controller/plant in Fig. 1.2 and such feedback links to be the interconnections of different nodes. We then ask how we identify and control over such network. In this thesis, most efforts have been concentrated on the identification part in the network setting, but we shall point interested readers to [76, 65] for results on networked control systems.

[1] sometimes called learning or prediction in different communities

[2] or equivalently, optimisation

In networked identification/control systems, there are mainly two types of different algorithms. One is the centralised algorithm; by which we try to monitor and control a networked system with all the information in the network. The other is the decentralised algorithm, by which we try to use local information to infer/achieve global properties. Local information is defined as of the information or measurement of a sub-set of nodes; global as the information of the whole set of nodes. Decentralisation is helpful in many ways, for instance, it helps control systems operate over a larger geographical area, it simplifies the complexity in controllers (algorithms) and leads to more efficient computations.

In real world, events take place via the interaction of individuals, organisations, and countries each pursuing their own interest. Control and optimisation can be viewed as special cases of the algorithm that we mentioned, which reflects the fundamental goal, to improve. However, most of control and optimisation can only be conducted locally due to the available information.

Unfortunately, the decentralised control problem is hard. The famous Witsenhausen problem [90] is a very good example to clearly state the difficulty in a decentralised decision and control problem [39]. In general, decentralised control problems are challenging due to

- Lack of global information. In most decentralised identification/control problems local information may not be able to infer/achieve some global properties. And the problem complicates if communication channels have limited capacity and signals are noisy, sampled and quantised. One should learn from "the blind men and elephant" fable (Fig. 1.3) and be careful about what global information can exactly be inferred from the local information and further how to carefully design algorithms to obtain some inferable information.

- Lack of global control strategy. Control law can only be implemented locally, the global stability and performance measure might not be satisfied and optimised based only on local information. There are a number of interesting attempts to design local controller to have global stability in the literature: a decentralised stability criteria for heterogeneous networks provided that the network structures are bipartite [48].

In this thesis we shall focus on decentralised algorithms for network prediction and reconstruction which can be viewed as sub-tasks in the framework of decentralised identification problem in the network setting. We shall proposed two types of decentralised algorithms that use local information in the network to infer some global properties.

FIGURE 1.3: The blind men and elephant: A group of blind men touch an elephant to learn what it is like. They conclude that the elephant is like a wall, snake, spear, tree, fan or rope, depending upon where they touch. They have a heated debate and the conflict is never resolved.

In chapter 2 we assume that only one node in the considered network is measured, and we try to compute the consensus value of the network under the assumption that this network reaches consensus. However, this does not tell us how the network is connected. In Chapter 3 we drop the consensus assumption and look at a more basic question for general networks. Here, we measure more than one node (but not all) and ask the question how do measured states relate (causally) to each other.

1.4 Contributions

Here are the main technical contributions of this thesis.

1.4.1 Chapter 2

◊ we formulate and propose the network prediction algorithm in the network consensus problem.

◊ we present a fully decentralised algorithm that allows any individual to compute the consensus value of the whole network in finite time using only the minimal number of successive values of its own history.

◊ we prove that the minimal number of steps can be characterised using other alge-

braic and graph theoretical notions: minimal external equitable partition (mEEP) that can be directly computed from the Laplacian matrix of the graph and from the underlying network structure.

◊ we consider a number of possible theoretical extensions of the proposed algorithm to issues arising from practical applications, e.g., time-delays, noise, external inputs, nonlinearities in the network, and analyse how the previous algorithm should be changed to incorporate such constraints.

1.4.2 Chapter 3

◊ for the purpose of encoding structural information into the representation of a linear time-invariant (LTI) system we define a new presentation: dynamical structure functions. We explore the properties of such representations.

◊ similar to a well-studied representation of LTI system, transfer function, we have studied a number of theoretical problems: identification, realisation, reduction, etc. for the dynamical structure functions.

◊ we apply the results to the biological network reconstruction problem that has been intensively studied in the literature and compare our reconstruction algorithm performance with the existing methods.

1.5 Notations

The thesis is self-contained and only requires some very basic preliminaries in graph theory and system theory; good references might be [33, 107]. The notation in this thesis is also fairly standard. For a matrix $A \in \mathbb{R}^{M\times N}$, $A[i,j] \in \mathbb{R}$ denotes the element in the i^{th} row and j^{th} column, $A[i,:] \in \mathbb{R}^{1\times N}$ denotes its i^{th} row, $A[:,j] \in \mathbb{R}^{M\times 1}$ denotes its j^{th} column, and $A[i_1 : i_2, j_1 : j_2]$ denotes the submatrix of A defined by the rows i_1 to i_2 and the columns j_1 to j_2. For a column vector $\alpha \in \mathbb{R}^{N\times 1}$, $\alpha[i]$ denotes its i^{th} element. We denote by $e_r^T = [0,\ldots,0,1_{r^{th}},0,\ldots,0] \in \mathbb{R}^{1\times N}$. Furthermore, I_N denotes the identity matrix of size N.

Chapter 2

Decentralised network prediction

In this chapter, we shall propose a decentralised network prediction algorithm in the network consensus problem framework to demonstrate the second two-line of Blake's poem *"Hold infinity in the palm of your hand, And eternity in an hour."*

2.1 Introduction

Generally speaking, the network consensus problem, which originates from a social decision making problem, is as follows: consider a group of individuals who act together as a team and suppose that each individual has its own subjective opinion or understanding of the unknown value of some parameter. A model is used to describe how the group might reach agreement on a common subjective opinion by pooling their individual opinions.

Well-known results give conditions to ensure that the opinion of each individual reaches the consensus value *asymptotically* [43, 62, 72]. However, it is unsatisfactory in practice since it requires an arbitrarily long time for individuals in this group to know the consensus value. A lot of efforts have been made to study and increase the convergence speed. For example, Olshevsky and Tsitsiklis [63, 64] stated the fundamental limitation on the convergence speed of such consensus-type dynamics. In Zhang *et al* [106], model predictive control is used to speed up the consensus process, while a new model for finite-time consensus is designed by Cortes [17] and Wang and Xiao [88]. Sundaram and Hadjicostis [82] proposed an algorithm that computed the asymptotic final consensus value of the network in finite-time.

Built on these state-of-the-art results, this chapter proposes a decentralised consensus computation algorithm that computes the consensus value of a network in *finite*

and minimal time using only the information observed by one individual in the group. Moreover, the information used for that purpose is solely based on the accumulation of successive values of the individual under consideration, and consequently, the corresponding algorithm is truly decentralised. Thus, the predicted future value of a network is computed by merely using observations at the node level, which coincides with Blake's vision in the second two-line.

Using networks to represent individuals and their connections, the consensus problem has broad implications beyond the analysis and design of group collective behaviour. Various applications can be cast in this framework, including swarming and flocking [83], distributed computing [5], agreement in social networks [60, 89] or synchronisation of coupled oscillators [68, 3, 81]. Regarding applications of the proposed algorithm, the consensus algorithm can be embedded as a sub-algorithm in a number of distributed algorithms such as distributed Kalman filtering, distributed computation, etc.. The consensus algorithm proposed in this Chapter is much faster than any other algorithm in the literature since it computes the final consensus value in a minimal number of steps.

The organisation of this chapter is as follows: After introducing the problem formulation of the consensus problem, we introduce an algorithm that allows any individual in a consensus-guaranteed network to compute the consensus value using a minimal number of steps. This algorithm relies on the analysis of the rank of a Hankel matrix constructed from local observations at any chosen node. Furthermore, we show that the minimal number of steps is linked to a global property of the network: the degree of a specific matrix polynomial. This provides us with an algebraic characterisation of the local convergence to consensus in terms of properties of the Laplacian matrix of the graph. Later, we show that the minimal number of steps required to compute the consensus value from local observations of any chosen node can also be characterised in terms of a combinatorial graph theoretical property: the minimal external equitable partition of the graph with respect to that node. We also consider some extensions of proposed algorithm to incorporate observations with uncertainties. Finally, we discuss some other issues raised by imperfect observations and briefly introduce some potential applications of the proposed algorithm.

2.2 Consensus dynamics: formulation and previous results

2.2.1 Formulation of the problem

Consider a directed unweighted graph denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of nodes, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges, and $W = \{W[i,j]\}_{i,j=1,\ldots,n}$ is the corresponding n by n adjacency matrix, with $W[i,j] = 1$ when there is a link from j to i, and $W[i,j] = 0$ when there is no link from j to i.

Let $x[i] \in \mathbb{R}$ denote the state of node i, which might represent the subjective opinion of individual i. The classical consensus problem on a network of continuous-time integrator individuals is defined by the following dynamics [62]:

$$\dot{x}(t) = -Lx(t),$$

where $L \in \mathbb{R}^{n \times n}$ is the Laplacian matrix induced by the topology $\mathcal{G}$. L is defined as $L[i,i] = \sum_{l \neq i}^{n} W[i,l]$, $\forall i = 1, \ldots, n$ and $L[i,j] = -W[i,j]$, $\forall i \neq j$.

Here we consider the associated discrete-time consensus dynamics on a network:

$$\begin{aligned} x_{k+1} &= (I_n - \epsilon L)\, x_k \triangleq A\, x_k \\ y_k &= e_r^T x_k = x_k[r], \end{aligned} \tag{2.1}$$

where $x_k \in \mathbb{R}^n$ and ϵ is the sampling time. Without loss of generality, we concentrate on the case where the measurable output $y_k \in \mathbb{R}$ corresponds to the local state of an arbitrarily chosen individual labelled r.

2.2.2 Global asymptotic convergence to distributed consensus (see [43, 62]):

Let $d_{\max} = \max_i L[i,i]$ denote the maximal node in-degree of the graph $\mathcal{G}$. If the network has a rooted directed spanning tree (or is connected in the case of an undirected graph) over time [43, 72], and the sampling time ϵ is such that $0 < \epsilon < 1/d_{\max}$ [1], then the discrete-time version of the classical consensus protocol given in (2.1) ensures global asymptotic convergence to consensus in the sense that

$$\lim_{k \to \infty} x_k = \left(c^T x_0\right) 1_{n \times 1}$$

[1] the choice of sampling time for each node needs global knowledge about $d_{\max}$

where $1_{n\times 1}$ is a column vector with all components equal to 1, and c^T is a constant row vector. In other words, the values of all nodes converge asymptotically to the same linear combination of the initial node values x_0.

Algebraic characterisation of distributed asymptotic consensus [91]:

When $c^T 1 = 1$, the iteration given by (2.1) achieves distributed consensus if and only if:

A.1 A has a simple eigenvalue at 1, and all other eigenvalues have a magnitude strictly less than 1.

A.2 The left and right eigenvectors of A corresponding to the eigenvalue 1 are c^T and 1, respectively.

2.2.3 Finite-time computation of the final consensus value [82]

Recent work by Sundaram and Hadjicostis [82] has shown that it is possible to obtain the final value of the consensus dynamics in a finite number of steps. Their result hinges on the use of the minimal polynomial associated with the consensus dynamics (2.1) in conjunction with the final value theorem.

Definition 2.2.1 (Minimal polynomial of a matrix)**.** *The minimal polynomial of matrix $A \in \mathbb{R}^{n\times n}$ is the monic polynomial $q(t) \triangleq t^{D+1} + \sum_{i=0}^{D} \alpha_i t^i$ with minimal degree $D+1$ that satisfies $q(A) = 0$.*

Given the explicit solution of the linear system in (2.1) with initial state x_0, it follows from the definition of the minimal polynomial that the dynamics in (2.1) satisfy the linear regression equation:

$$x_{k+D+1} + \alpha_D x_{k+D} + \ldots + \alpha_1 x_{k+1} + \alpha_0 x_k = 0, \forall k \in \mathbb{N}. \tag{2.2}$$

Similarly, the regression equation for $y_k = x_k[r]$, the measurable output at node r, is determined by the minimal polynomial of the corresponding matrix observability pair $[A, e_r^T]$.

Definition 2.2.2 (Minimal polynomial of a matrix pair)**.** *The minimal polynomial associated with the matrix pair $[A, e_r^T]$ denoted by $q_r(t) \triangleq t^{D_r+1} + \sum_{i=0}^{D_r} \alpha_i^{(r)} t^i$ is the monic polynomial of minimal degree $D_r + 1 \leq D + 1$ that satisfies $e_r^T q_r(A) = 0$.*

Remark 2.2.1. *The minimal polynomial of a matrix and the minimal polynomial of a matrix pair are unique due to the monic property.*

Again, it is straightforward to show that:

$$y_{k+D_r+1} + \alpha^{(r)}_{D_r} y_{k+D_r} + \ldots + \alpha^{(r)}_1 y_{k+1} + \alpha^{(r)}_0 y_k = 0, \forall k \in \mathbb{N}. \tag{2.3}$$

Therefore each node r will be associated with a particular length of the regression (D_r+1) which is upper bounded by the degree of the minimal polynomial of the dynamical matrix A.

Consider now the Z-transform of y_k[2]: $(\alpha^{(r)}_{D_r+1} = 1)$

$$Y(z) = \frac{\sum_{i=1}^{D_r+1} \alpha^{(r)}_i \sum_{\ell=0}^{i-1} y_\ell z^{i-\ell}}{q_r(z)} \triangleq \frac{H(z)}{q_r(z)}. \tag{2.4}$$

Parameters $\alpha^{(r)}_i$s differ with the choice of rs, we shall drop the superscript (r) for notational simplicity.

On the assumptions specified in Section 2.2.2, the minimal polynomial $q_r(t)$ does not possess any unstable root apart from one at 1. We can then define the following polynomial:

$$p_r(z) \triangleq \frac{q_r(z)}{z-1} \triangleq \sum_{i=0}^{D_r} \beta_i z^i. \tag{2.5}$$

The application of the final value theorem [32] then gives the consensus value

$$\phi = \lim_{z\to 1}(z-1)Y(z) = \frac{H(1)}{p_r(1)} = \frac{y^T_{D_r}\beta}{\mathbf{1}^T\beta} \tag{2.6}$$

where $y^T_{D_r} = \begin{bmatrix} y_0 & y_1 & \ldots & y_{D_r} \end{bmatrix}$ and $\beta_{(D_r+1)\times 1}$ is the vector of coefficients of the polynomial $p_r(z)$ defined in eq. (2.5).

Based on these results, an algorithm to obtain the consensus value was proposed in [82]. The proposed algorithm was distributed but not entirely local, in the sense that a local calculation is repeated over n independent iterations (where n is the total number of nodes of the network) and at each iteration it requires each node to store its own values for $n+1$ steps. Hence a total of $n(n+1)$ successive values of $x[r]$ is required for the calculation of ϕ.

[2]This follows from the time-shift property of the Z-transform: $\mathcal{Z}(x_{k+n}) = z^n X(z) - \sum_{l=0}^{n-1} z^{n-l} x_l$ where $X(z) = \mathcal{Z}(x_k)$.

2.2.4 Minimal-time, decentralised computation of the final consensus value

The main purpose of this chapter is to characterise the computation of the final consensus value ϕ using only the output observations $y_k = x_k[r]$ of the node r *in minimal time*. We formalise and improve here our previous results [99] and show that, for a general arbitrary initial condition, except for a set of initial conditions with Lebesgue measure zero [7], the consensus value can be obtained from local observations in a minimal number of steps that does not depend explicitly on the total size of the graph. In our framework the minimal number of steps is computed in a truly decentralised manner by checking a rank condition of a Hankel matrix constructed exclusively from local output observations. We also provide a graph theoretical characterisation of this local property in terms of the minimal external equitable partition of the graph. This characterisation provides understanding into which properties of the graph contribute to the disparity in the ability of the different nodes to compute the global consensus value from local information.

2.3 Minimal time consensus and the Jordan block decomposition of the consensus dynamics

Given the linear system in (2.1) and an initial state x_0, it follows from the above that there always exist scalars $d \triangleq d(r,x_0) \in \mathbb{N}$ and $a_0,...,a_d \in \mathbb{R}$ such that the following linear regression equation is satisfied $\forall k \in \mathbb{N}$

$$x_{k+d+1}[r] + a_d x_{k+d}[r] + \ldots + a_1 x_{k+1}[r] + a_0 x_k[r] = 0. \tag{2.7}$$

From the definitions above it is clear that $D_r + 1$ is the *minimal length of recursion*:

$$D_r + 1 = \min_{d\in\mathbb{N}} \max_{x_0\in\mathbb{R}^n} \{d(r,x_0)+1\text{: eq. (2.7) holds } \forall k\}.$$

Remark 2.3.1. *Among the many recursions of length d that are not necessarily minimal, (D_r+1) appears as a min-max over the space of (d,x_0). When $d+1 = D_r+1$, the coefficients a_i in (2.7) correspond to $\alpha_i^{(r)}$, the coefficients of the minimal polynomial of the matrix pair $[A, e_r^T]$ in (2.3).*

In this section,we give an algebraic characterisation of the minimal number of steps $D_r + 1$ based on the projection of the Jordan block decomposition of A^k on e_r^T. Our aim

is to obtain the coefficients $\alpha_i^{(r)}$ in (2.3) from data so that we can compute future outputs recursively. Consider the standard Jordan decomposition:

$$A = SJS^{-1} \quad \text{where} \tag{2.8}$$

$$S = \begin{bmatrix} s_1 & s_2 & \dots & s_n \end{bmatrix} \tag{2.9}$$

$$J = \text{diag}\{J_1(\lambda_1), J_2(\lambda_2), \dots, J_l(\lambda_l)\} \tag{2.10}$$

where

$$J_i(\lambda_i) = \begin{bmatrix} \lambda_i & 1 & & & \\ & \lambda_i & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda_i & 1 \\ & & & & \lambda_i \end{bmatrix}_{n_i \times n_i}, \tag{2.11}$$

and s_i, the columns of the non singular matrix S, are the generalised eigenvectors of A [107]. The matrix A has l eigenvalues λ_i[3], each of them associated with a Jordan block of size n_i, such that $\sum_{i=1}^{l} n_i = n$. Without loss of generality, we assume that the blocks are ordered in decreasing size: $n_1 \geq n_2 \geq \ldots \geq n_l$.

Using eq. (2.8), the linear dynamics (2.1) can be rewritten as follows:

$$x_k[r] = e_r^T A^k x_0 = \left(e_r^T S\right) J^k \left(S^{-1} x_0\right) \triangleq \sigma^T J^k \chi, \tag{2.12}$$

where the vectors

$$\sigma^T = \begin{bmatrix} \sigma_1^T & \sigma_2^T & \dots & \sigma_l^T \end{bmatrix}_{1 \times n} \tag{2.13}$$

$$\chi^T = \begin{bmatrix} \chi_1^T & \chi_2^T & \dots & \chi_l^T \end{bmatrix}_{1 \times n} \tag{2.14}$$

are partitioned according to the Jordan blocks in (2.8), e.g., $\sigma_1^T = \begin{bmatrix} \sigma_{11} \dots \sigma_{1n_i} \end{bmatrix}$ and $\chi_1^T = \begin{bmatrix} \chi_{11} \dots \chi_{1n_i} \end{bmatrix}$. Here,

$$J^k = \text{diag}\left\{J_1^k(\lambda_1), J_2^k(\lambda_2), \dots, J_l^k(\lambda_l)\right\}$$

has the well-known structure [42]:

$$J_i^k(\lambda_i) = \sum_{m=0}^{k-1} \binom{k}{m} \lambda_i^{k-m} J_i^m(0), \tag{2.15}$$

[3]For $i \neq j$, λ_i might equal to λ_j

where $J_i^m(0)$ is the m-th power of a Jordan block, as defined in (2.11).

The output dynamics (2.12) then becomes:

$$x_k[r] = \sum_{i=1}^{l} \sum_{m=0}^{k-1} \binom{k}{m} \lambda_i^{k-m} \left[\sigma_i^T J_i^m(0) \chi_i \right]. \tag{2.16}$$

Note that because of its Jordan block structure the matrix $J_i^m(0)$ induces a strict m-shift on the vector χ_i for $m \leq n_i$. Therefore if $k \geq \max_i\{n_i\}$, we have:

$$x_k[r] = \sum_{i=1}^{l} \sum_{m=0}^{n_i-1} \binom{k}{m} \lambda_i^{k-m} \left[\sum_{j=1}^{n_i-m} \sigma_{ij} \chi_{ij+m} \right] \triangleq \sum_{i=1}^{l} \sum_{m=0}^{n_i-1} \binom{k}{m} \lambda_i^{k-m} g_{im} \tag{2.17}$$

However, some of the g_{im} might be zero (we might even have situations in which all the coefficients associated with a particular eigenvalue are zero) so that the dynamics of node r can be written as:

$$x_k[r] = \sum_{i=1}^{l_r} \sum_{m=0}^{n_i^r-1} \binom{k}{m} \lambda_i^{k-m} g_{im} \tag{2.18}$$

where $n_i^r \leq n_i$ and $l_r \leq l$. Here, $\{\lambda_1, \ldots, \lambda_{l_r}\}$ is an ordered subset of *distinct* eigenvalues from the original Jordan block decomposition. In consequence, the degree of the characteristic polynomial that underlies the length of the recursion for node r is:

$$\sum_{i=1}^{l_r} n_i^r = D_r + 1.$$

Eq. (2.18) can be rewritten as a dot product:

$$x_k[r] = v_r(k)^T g_r \triangleq \begin{bmatrix} v_1^T(k) & v_2^T(k) & \ldots & v_{l_r}^T(k) \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{l_r} \end{bmatrix}$$

where

$$v_i^T(k) \triangleq \begin{bmatrix} \binom{k}{0}\lambda_i^k & \binom{k}{1}\lambda_i^{k-1} & \ldots & \binom{k}{n_i^r-1}\lambda_i^{k-n_i^r+1} \end{bmatrix}_{1 \times n_i^r}$$
$$g_i^T \triangleq \begin{bmatrix} g_{i0} & \ldots & g_{i(n_i^r-1)} \end{bmatrix}.$$

Based upon the decomposition of confluent Vandermonde matrices introduced in [10],

it is easy to see that

$$v_i^T(k) = \bar{e}_i^T \mathscr{J}_i^k(\lambda_i)$$

where $\mathscr{J}_i(\lambda_i)$ is a Jordan block of size n_i^r as defined in (2.11) and $\bar{e}_i^T = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}_{1\times n_i^r}$ is the unit vector of the same length. The dynamics (2.12) can thus be rewritten in terms of a Jordan decomposition of reduced dimensionality as follows:

$$x_k[r] = {E_r}^T J_r^k g_r, \ \forall \ k, \tag{2.19}$$

where

$$\begin{aligned} {E_r}^T &\triangleq \begin{bmatrix} \bar{e}_1^T & \dots & \bar{e}_{l_r}^T \end{bmatrix}_{1\times(D_r+1)} \text{ and} \\ J_r &\triangleq \operatorname{diag}\{\mathscr{J}_1(\lambda_1), \mathscr{J}_2(\lambda_2), \dots, \mathscr{J}_{l_r}(\lambda_{l_r})\} \end{aligned} \tag{2.20}$$

are partitioned according to the l_r blocks.

From the above analysis we have the following lemma.

Lemma 2.3.1. *Consider the discrete-time LTI system* (2.1)*. The minimal polynomial associated with $x[r]$, as given in Definition 2.2.2, is the characteristic polynomial of the matrix J_r in eq.* (2.19) *which has order $D_r + 1 = \sum_{i=1}^{l_r} n_i^r$. The final consensus value ϕ can be computed from eq.* (2.6) *based on the coefficients of the minimal polynomial of the pair $[A, e_r^T]$ and the successive values of $x[r]$.*

Proof. The Jordan matrix J_r in eq. (2.19) has the property that each of its Jordan block has distinct eigenvalues. Hence the minimal polynomial of $[A, e_r^T]$ is the same as the characteristic polynomial of $[J_r, e_r^T]$ (see [42]): $e_r^T q_r(A) = e_r^T q_r(J_r)$. Therefore the minimal polynomial possesses the following explicit form: $\det(J_r - tI) = \prod_{i=1}^{l_r}(t-\lambda_i)^{n_i^r} = t^{D_r+1} + \alpha_{D_r} t^{D_r} + \ldots + \alpha_1 t + \alpha_0$, and has degree $D_r + 1$. This latter relationship also shows that $D_r + 1 = \sum_{i=1}^{l_r} n_i^r$. □

Remark 2.3.2. *Lemma 2.3.1 states that instead of an n-dimensional Jordan block form J of $x_k[r]$, as in eq.* (2.12)*, the general expression of $x_k[r]$ can be written in terms of a smaller $D_r + 1$-dimensional Jordan matrix J_r, as in eq.* (2.19)*.*

Remark 2.3.3. *The minimal integer value $D_r + 1$ necessary for the recursion* (2.7) *to hold for* almost any *initial condition x_0 is given by the degree of the minimal polynomial of the observability pair $[A, e_r^T]$ (see [99]). In other words, eq.* (2.7) *holds for a randomly chosen initial state x_0, except for a set of initial conditions of Lebesgue measure zero [7].*

2.4 Decentralised minimal-time consensus computation algorithm

In the decentralised problem we assume that node r does not have access to any external information such as the total number of individuals n in the network, the local communication links around node r or the state values or number of its neighbours. In [99], we showed that for the general discrete-time LTI system (2.1), $2D_r+3$ successive discrete-time steps are needed by individual r to compute the final value in a fully decentralised manner (Algorithm 1). This is a general result for all linear systems, if the communication network is well-designed for consensus (i.e. Assumptions A.1 and A.2 are satisfied and asymptotic convergence to consensus is guaranteed), we can here propose an algorithm (Algorithm 2) that computes the final value using $2D_r+2$ successive discrete-time steps, i.e. one step fewer than Algorithm 1 [99].

Problem 2.4.1 (Decentralised problem)**.** *Consider the discrete-time LTI dynamics in eq.* (2.1) *where an arbitrarily chosen state $x[r]$ is observed and assume that the conditions for consensus (Assumptions A.1 and A.2) are satisfied. The decentralised problem is to compute the asymptotic value of this state $\phi = \lim_{k\to\infty} x_k[r]$ using only its own previously observed values $y_k = x_k[r]$.*

Consider the vector of successive discrete-time values at node r, $X_{0,1,\dots,2k}[r] = \{x_0[r],\ x_1[r],\dots,x_{2k}[r]\}$, and its associated Hankel matrix:

$$\Gamma\{X_{0,1,\dots,2k}[r]\} \triangleq \begin{bmatrix} x_0[r] & x_1[r] & \dots & x_k[r] \\ x_1[r] & x_2[r] & \dots & x_{k+1}[r] \\ \vdots & \vdots & \ddots & \vdots \\ x_k[r] & x_{k+1}[r] & \dots & x_{2k}[r] \end{bmatrix} \quad k \in \mathbb{Z}. \tag{2.21}$$

Based on the linear iteration in eq. (2.3), we can then propose the following algorithm using $2D_r+3$ steps to compute the final value of the r^{th} node. Due to the assumption that the network will reach consensus, then the final value any node should be the same. As a result, the computed final value is the consensus value.

Notice that, the assumption of consensus also implies the Z-transform of regression in eq.(2.3), i.e. eq. (2.5) has a root at 1. Can one use this prior information to reduce the number of steps in computing the final consensus value? Algorithm 2 then allows us to compute the final consensus value in $2D_r+2$ steps.

Algorithm 1 Decentralised final value computation

Data: Successive observations of $x_i[r],\ i = 0, 1, \ldots$.
Result: Final consensus value: ϕ.

Step 2.4.1. *Increase the dimension k of the square Hankel matrix $\Gamma\{X_{0,1,\ldots,2k}[r]\}$ until it loses rank and store the first defective Hankel matrix.*

Step 2.4.2. *The kernel $\alpha = \begin{bmatrix}\alpha_0 & \ldots & \alpha_{D_r} & 1\end{bmatrix}^T$ of the first defective Hankel matrix gives the coefficients of eq.* (2.3)*.*

Step 2.4.3. *Solve for vector $\beta \in \mathbb{R}^{D_r+1}$ from equation $[\beta^T \ \ 0] + [0 \ \ \beta^T] = \alpha^T$.*

Step 2.4.4. *Compute the final consensus value ϕ using eq.* (2.6)*.*

We also define the vector of differences between successive values of $x[r]$:

$$\bar{X}_{0,1,\ldots,2k}[r] = \{x_1[r] - x_0[r], \ldots, x_{2k+1}[r] - x_{2k}[r]\}.$$

Algorithm 2 Decentralised minimal-time consensus value computation

Data: Successive observations of $x_i[r],\ i = 0, 1, \ldots$.
Result: Final consensus value: ϕ.

Step 2.4.1. *Compute the vector of differences $\bar{X}_{0,1,\ldots,2k}$ and increase the dimension k of the square Hankel matrix $\Gamma\{\bar{X}_{0,1,\ldots,2k}[r]\}$ until it loses rank and store the first defective Hankel matrix.*

Step 2.4.2. *The kernel $\beta = \begin{bmatrix}\beta_0 & \ldots & \beta_{D_r-1} & 1\end{bmatrix}^T$ of the first defective Hankel matrix gives the coefficients of eq.* (2.6)*.*

Step 2.4.3. *Compute the final consensus value ϕ using eq.* (2.6)*.*

To understand Algorithm 2, consider a Vandermonde factorisation [10] of the Hankel matrix (2.21):

$$\Gamma\{X_{0,1,\ldots,2k}[r]\} = V(0,k)T_r V^T(0,k), \tag{2.22}$$

in which we have defined the confluent Vandermonde matrix

$$V(0,k)_{(k+1)\times(D_r+1)} = \begin{bmatrix} E_r^T \\ E_r^T J_r \\ \vdots \\ E_r^T J_r^k \end{bmatrix}, \tag{2.23}$$

in terms of the elements defined in eq. (2.20). As shown in [10], the $(D_r+1)\times(D_r+1)$ block diagonal matrix

$$T_r = \text{diag}\{T_{r,1},\ldots,T_{r,l_r}\},\ T_{r,i} \in \mathbb{R}^{n_i^r \times n_i^r},$$

has the following symmetric upper anti-diagonal form:

$$T_{r,i} = \begin{bmatrix} * & * & * & * & t_i \\ * & * & * & t_i & \\ * & * & \iddots & & \\ * & t_i & & 0 & \\ t_i & & & & \end{bmatrix},$$

where t_i and $*$ are determined from the values of $x_k[r]$.

Without loss of generality, consider $\lambda_1 = 1$ so that $T_{r,1} \in \mathbb{R}$. We then have

$$\begin{aligned}
&\Gamma\{\bar{X}_{0,1,\ldots,2k}[r]\} \\
&= \Gamma\{X_{1,2,\ldots,2k+1}[r]\} - \Gamma\{X_{0,1,\ldots,2k}[r]\} \\
&= VT_r\text{diag}\{\lambda_1,\ldots,\lambda_{l_r}\}V^T - VT_rV^T \\
&= VT_r\text{diag}\{0,\lambda_2-1,\ldots,\lambda_{l_r}-1\}V^T \\
&= V\,\text{diag}\{0,(\lambda_2-1)T_{r,2},\ldots,(\lambda_{l_r}-1)T_{r,l_r}\}V^T \\
&= V'\text{diag}\{(\lambda_2-1)T_{r,2},\ldots,(\lambda_{l_r}-1)T_{r,l_r}\}V'^T,
\end{aligned}$$

where $V' = V[2:k+1, 2:D_r+1]$. From the last equation, it is easy to see that for $\Gamma\{\bar{X}_{0,1,\ldots,2k}[r]\}$ to be defective, one must have $k \geq D_r+1$.

Theorem 2.4.1. *Consider the system in* (2.1) *and assume that the conditions for consensus (Assumptions A.1 and A.2) are satisfied. Then the minimal number of successive discrete-time values, starting from step i, for the arbitrarily chosen state* $x[r]$*, is* $2(D_r+1)-\delta_r-\min\{i,\delta_r\}$*, where* δ_r *is the number of zero roots in* $q_r(t)=0$*.*

Proof. Combining the above derivations and performing a proof similar to the one presented in [Corollary 1, [99]] (by taking $z_k \triangleq x_{k+1}[r]-x_k[r]$ as y_k in that Corollary) yields the result. □

We hereafter provide a counter-example showing that the computation of these coefficients is not possible if one only uses $2D_r+1$ consecutive values of $x_k[r]$.

We consider a general question, given a linear regression of y_i with length D_r+2

$$y_{k+D_r+1}+a_{D_r}y_{k+D_r}+\ldots+a_0y_k=0. \tag{2.24}$$

Then, we shall show that to fully reconstruct this regression using successive outputs of $y_i, i=1,\ldots$, it requires at least $2D_r+3$ steps.

We show this by contradiction: we shall show that we can not reconstruct eq. (2.24) using $2D_r+2$ outputs. Assume there exists a matrix $\begin{bmatrix} b_0 & c_0 \\ b_1 & c_1 \\ \vdots & \vdots \\ b_{D_r-1} & c_{D_r-1} \\ b_{D_r} & c_{D_r} \end{bmatrix}$ such that

$$\Gamma\{y_0,y_1,\ldots,y_{2D_r}\}\begin{bmatrix} b_0 & c_0 \\ b_1 & c_1 \\ \vdots & \vdots \\ b_{D_r-1} & c_{D_r-1} \\ b_{D_r} & c_{D_r} \end{bmatrix}=\begin{bmatrix} y_{D_r+1} & 0 \\ y_{D_r+2} & 0 \\ \vdots & \vdots \\ y_{2D_r+1} & 0 \\ 0 & 1 \end{bmatrix} \tag{2.25}$$

Since $\Gamma\{y_0,y_1,\ldots,y_{2D_r}\}$ has full rank and thus invertible, eq. (2.25) can always be solved for b_i and c_i, $\forall i=0,\ldots,D_r$. Based on eq. (2.25), we can define two polynomials:

$$b(z)=z^{D_r+1}-b_{D_r}z^{D_r}-\ldots-b_0$$
$$c(z)=-c_{D_r}z^{D_r}-\ldots-c_0.$$

The minimal polynomial of $a(z)\triangleq= z^{D_r+1}+a_{D_r}z^{D_r}+\ldots+a_0=b(z)+y_{2D_r+2}c(z)$ [10]. If we only use the first $2D_r+2$ successive values of y, i.e. $y_0,y_1,\ldots,y_{2D_r+1}$, then we can choose different y_{2D_r+2} to obtain different $a(z)$, contradiction!

At present, we only focus on the ideal model in eq. (2.1). For simplicity of exposition, we further make the following assumption in the rest of this chapter:[4]

[4]When A has some eigenvalues at 0, the expression of the minimal number of steps for node r to compute

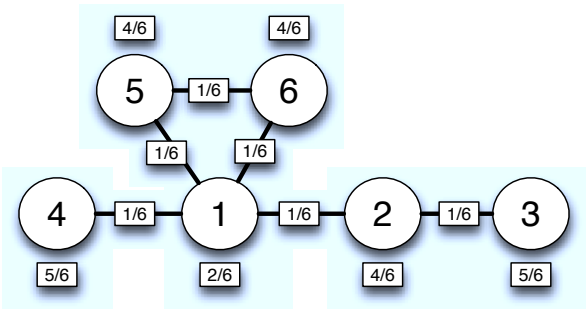


FIGURE 2.1: Underlying topology for Example 2.4.1 with sampling time $\epsilon = 1/6$.

A.3 The matrix A in eq. (2.1) does not possess any eigenvalue at 0.

Under Assumption A.3, Theorem 2.4.1 establishes that the minimal number of steps for node r to compute the final consensus value is $2D_r + 2$.

Example 2.4.1. *Consider the network topology in Fig. 2.1 under dynamics* (2.1) *with* $A \triangleq I_n - \epsilon L$ *and a sampling time* $\epsilon = 1/6$. *The topology is undirected and connected and A satisfies assumptions A.1, A.2, and A.3. Therefore the final value of each node is the average of the initial state values. For the randomly chosen initial state* $x_0 = \begin{bmatrix} 1.3389 & 2.0227 & 1.9872 & 6.0379 & 2.7219 & 1.9881 \end{bmatrix}^T$, *the final consensus value is thus* 2.6828. *We now apply Algorithm 2 to node* $r = 1$.

Step 2.4.1. *We increase the dimension k of the square Hankel matrix* $\Gamma\{\bar{X}_{0,1,\ldots,2k}[1]\}$ *until it loses rank. This happens for* $k = 4$. *We then store the first defective Hankel matrix:*

$$\Gamma\{\bar{X}_{0,1,\ldots,8}[1]\} = \begin{bmatrix} 1.2358 & 0.2050 & 0.0367 & 0.0047 \\ 0.2050 & 0.0367 & 0.0047 & -0.0037 \\ 0.0367 & 0.0047 & -0.0037 & -0.0067 \\ 0.0047 & -0.0037 & -0.0067 & -0.0079 \end{bmatrix}.$$

Step 2.4.2. *The normalised kernel of the first defective Hankel matrix is*

$$\beta = \begin{bmatrix} -0.0833 & 0.7778 & -1.6667 & 1 \end{bmatrix}^T.$$

This gives the coefficients of eq. (2.6).

Step 2.4.3. *We compute the final consensus value* $\phi = 2.6828$ *using eq.* (2.6).

the final consensus value takes a more complicated form, see [99].

As shown here for node $r = 1$, the value of ϕ obtained in a decentralised manner is equal to the average of the initial states.

Repeating this procedure for each of the six nodes gives the same value ϕ. However, the number of steps required by each node to compute the final consensus value ϕ differs. This is summarised in Table 2.1.

	Ref. [82]	Our result
$Node\ 1$	$6 \times 7 = 42$	$2 \times 4 = 8$
$Node\ 2$	$6 \times 7 = 42$	$2 \times 4 = 8$
$Node\ 3$	$6 \times 7 = 42$	$2 \times 4 = 8$
$Node\ 4$	$6 \times 7 = 42$	$2 \times 5 = 10$
$Node\ 5$	$6 \times 7 = 42$	$2 \times 6 = 12$
$Node\ 6$	$6 \times 7 = 42$	$2 \times 6 = 12$

TABLE 2.1: Comparison of the minimal number of successive values needed by each node to compute the final consensus value of the network in Fig. 2.1 with $n = 6$ nodes.

While the method proposed in [82] requires a total of $n(n+1)$ successive values of $x[r]$, our algorithm shows that the minimal number of successive values of $x[r]$ is just $2(D_r+1)$ for *almost all* initial conditions. Furthermore, our algorithm is completely decentralised, i.e. our result does not require that the arbitrarily chosen state $x[r]$ has any knowledge of the total number of nodes in the network, n, or any other kind of global (centralised) information about the network (contrary to what is assumed in [82, Section V]).

As can be noted in Table 2.1, some nodes need fewer successive observations of their own state to compute the final consensus value of the whole network. In what follows we call such nodes *dominant nodes*. An important question arises at this point: given a consensus-guaranteed network, can we identify the dominant nodes? Below, we answer this question using an algebraic characterisation of the minimal number of steps which we then link to a specific graph partition of the consensus network around the chosen node.

2.5 Characterisation on the minimal number of steps

We now provide an answer to the question raised at the end of the last section from two perspectives. Firstly, in Section 2.5.1 we provide an algebraic characterisation of the minimal recursion length $D_r + 1$ for node r by performing an analysis of the Laplacian of the graph. Secondly, in Section 2.5.2 we can directly compute this minimal recursion length $D_r + 1$ graph-theoretically from Mason's rule [54] and also link $D_r + 1$ to the

number of cells in a special partition of the graph called the *minimal external equitable partition with respect to node r*.

For simplicity of exposition, we only consider undirected graphs in the following sections, i.e. we assume:

A.4 The matrices W, L, A in eq. (2.1) are symmetric.

2.5.1 Algebraic characterisation

An algebraic characterisation of the degree of the minimal polynomial of $[A, e_r^T]$ can be obtained using the Jordan block decomposition described in Section 2.3. The symmetry of the Laplacian matrix in undirected graphs simplifies the analysis since the Jordan matrix in Eq. (2.12) becomes diagonal. The following Corollary provides a link between the minimal number of successive values required by a node to compute the final consensus value of the network and algebraic properties of the underlying graph. Before presenting the main result, we introduce the following notation, which will be used extensively in the remainder of the chapter.

Definition 2.5.1 (D-cardinality of a set)**.** *Let Λ be a finite set, potentially containing repeated elements, with cardinality $card\{\Lambda\}$. The d-cardinality of the set, denoted $dcard\{\Lambda\}$, is defined as the number of distinct elements in the set.*

Example 2.5.1. *Let $\Lambda = \{1,2,3,1,3,5\}$. Then $card\{\Lambda\} = 6$ and $dcard\{\Lambda\} = 4$.*

Our first algebraic characterisation of the minimal recursion length at node r relates $D_r + 1$ to the number of distinct eigenvalues of the Laplacian matrix whose eigenvectors have non-zero components for node r, as given by the following Corollary.

Corollary 2.5.1. *Consider the dynamics* (2.1) *where A is associated with an unweighted and undirected graph. Denote the eigenvalues of the symmetric matrix A by λ_i ($i = 1,\dots,n$) (allowing repeated eigenvalues) and their corresponding right eigenvector by u_i. Let $\Lambda = \{\lambda_i(A) \,|\, i = 1,\dots,n\}$ and $\Psi_r = \{\lambda_i(A) \,|\, u_i[r] = 0\}$. Then*

$$D_r + 1 = dcard\{\Lambda/\Psi_r\},$$

where Λ/Ψ_r is the relative complement of Ψ_r in Λ.

Proof. Since A is symmetric, all eigenvalues of A are real. The proof then follows from Lemma 2.3.1 and the PBH-test [107]. □

Our second algebraic characterisation relates $D_r + 1$ with the number of eigenvalues shared by the Laplacian matrix and the r-grounded Laplacian matrix.

Theorem 2.5.1. *Consider the system in Eq.* (2.1) *satisfying Assumptions A.1–A.4 and A has no repeated eigenvalues. The rank of the observability matrix for the pair $[A, e_r^T]$ is equal to $n - \mu_r$, i.e.*

$$D_r + 1 = n - \mu_r,$$

where μ_r is the number of eigenvalues shared between A and A_r, where A_r is the r-grounded matrix, i.e. the submatrix of A obtained by deleting the r^{th} row and the r^{th} column.

Proof. Without loss of generality, we let $r = 1$.Since $A_1 = [0\ I_{n-1}]A[0\ I_{n-1}]^T$, if $\lambda_j(A) = \lambda_j(A_1)$ for $j = 1,\dots,\mu_1$, we claim that if v_i (for any $i \in \{1,\dots,\mu_1\}$) is an eigenvector for A_1 corresponding to $\lambda_i(A_1)$, then $[0\ v_i^T]^T$ is an eigenvector of A associated with eigenvalue $\lambda_i(A)$ for $i = 1,\dots,\mu_1$.

To shown this claim, we shall show that if there exists v_i^1, v_i^2, v_i^3 satisfying the following equations, then $v_i^1 = 0$ (given i).

$$\begin{bmatrix} A[1,1] & A[1,2:n] \\ A[2:n,1] & A_1 \end{bmatrix} \begin{bmatrix} v_i^1 \\ v_i^2 \end{bmatrix} = \lambda_i \begin{bmatrix} v_i^1 \\ v_i^2 \end{bmatrix},$$

$$A_1 v_i^3 = \lambda_i v_i^3.$$

This is equivalent to

$$A[1,1]v_i^1 + A[1,2:n]v_i^2 = \lambda_i v_i^1 \tag{2.27}$$

$$A[2:n,1]v_i^1 + A_1 v_i^2 = \lambda_i v_i^2 \tag{2.28}$$

$$A_1 v_i^3 = \lambda_i v_i^3. \tag{2.29}$$

Left multiply eq. (2.29) with $v_i^{3,T}$, the following equality due to the symmetry of A_1.

$$v_i^{3,T} A[2:n,1] v_i^1 = 0$$

If $v_i^1 \neq 0$, then $v_i^{3,T} A[2:n,1] = 0$. Or in other words,

$$A[1,2:n]v_i^3 = 0 \tag{2.30}$$

Furthermore we have the following equation obtained from eq. (2.30)

$$\begin{bmatrix} A[1,1] & A[1,2:n] \\ A[2:n,1] & A_1 \end{bmatrix} \begin{bmatrix} 0 \\ v_i^3 \end{bmatrix} = \lambda_i \begin{bmatrix} 0 \\ v_i^3 \end{bmatrix},$$

we then find two eigenvectors of A correspond to λ_i, contradiction.

We have proven that if $\lambda_j(A) = \lambda_j(A_1)$ for $j = 1,\ldots,\mu_1$, we claim that if v_i (for any $i \in \{1,\ldots,\mu_1\}$) is an eigenvector for A_1 corresponding to $\lambda_i(A_1)$, then $[0\ v_i^T]^T$ is an eigenvector of A associated with eigenvalue $\lambda_i(A)$ for $i = 1,\ldots,\mu_1$.

In the rest of the proof, we shall show firstly that $D_r + 1 \geq n - \mu_r$ and then show that $D_r + 1 \leq n - \mu_r$ and finally we can conclude that $D_r + 1 = n - \mu_r$.

From the definition of an eigenvalue, we then have the following equation:

$$\begin{bmatrix} A[1,1] & A[1,2:n] \\ A[2:n,1] & A_1 \end{bmatrix} \begin{bmatrix} 0 \\ v_i \end{bmatrix} = \lambda_i(A_1) \begin{bmatrix} 0 \\ v_i \end{bmatrix}, \tag{2.31}$$

for $i = 1,\ldots,\mu_1$.

Now, consider the observability matrix $[A, e_1^T]$ (see [107]):

$$\Omega = \begin{bmatrix} e_1^T \\ e_1^T A \\ \cdots \\ e_1^T A^{n-1} \end{bmatrix}. \tag{2.32}$$

It is easy to see that

$$\Omega \begin{bmatrix} 0 & 0 & \ldots & 0 \\ v_1 & v_2 & \ldots & v_{\mu_1} \end{bmatrix} = 0.$$

Therefore the dimension of the kernel of Ω is at least μ_1.

We now prove the necessity. If a vector v satisfies eq. (2.33):

$$\Omega v = \begin{bmatrix} e_1^T \\ e_1^T A \\ \cdots \\ e_1^T A^{n-1} \end{bmatrix} v = 0, \tag{2.33}$$

then we know that $[A, e_1^T]$ is unobservable. Therefore there exists a $\lambda \in \mathbb{C}$ such that $Av = \lambda v$ and $e_1^T v = v[1] = 0$ from the PBH rank test [107]. Now, let $v = [0, u]^T$. We then

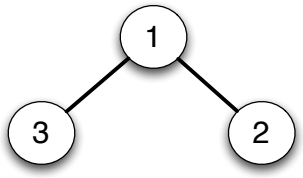


FIGURE 2.2: Example illustrating algebraic interpretation of minimal number of steps.

have:

$$\begin{bmatrix} A[1,1] & A[1,2:n] \\ A[2:n,1] & A_1 \end{bmatrix} \begin{bmatrix} 0 \\ u \end{bmatrix} = \lambda \begin{bmatrix} 0 \\ u \end{bmatrix},$$

or equivalently, $A_1 u = \lambda u$. This shows that λ is a common eigenvalue of A and A_1.

We can finally conclude that $\text{rank}(\Omega) = n - \dim(\text{Kernel}(\Omega)) = n - \mu_1$.

□

Example 2.5.2. *Consider the dynamics governed by the matrix in Fig. 2.2*

$$L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, A = I - \frac{1}{3}L = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} \end{bmatrix}.$$

The observability pair relative to node $r = 1$ is $[A, [1\ 0\ 0]]$. *Therefore the corresponding minimal polynomial is $q_1(t) = t^2 - t$ and $D_1 + 1 = 2$ since the order of this polynomial is* 2.

On the other hand, the observability matrix Ω is such that

$$\text{rank}(\Omega) = \text{rank}\left(\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}\right) = 2.$$

Since $\Lambda = \{1, \frac{2}{3}, 0\}$ and $\Psi_1 = \{\frac{2}{3}\}$, we have from Corollary 2.5.1 that

$$dcard\{\Lambda/\Psi_1\} = 2.$$

To illustrate Theorem 2.5.1, note that A and A_1 share an eigenvalue at $\frac{2}{3}$, therefore $D_1 + 1 = 2$.

2.5.2 Graph-theoretical characterisation

Mason's rule

Consider the system in eq. (2.1), and a virtually built graph [33] $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A/z)$, where $\mathcal{V}, \mathcal{E}$ denotes the vertex/edge set, A is the state-space matrix of the considered discrete-time LTI system in eq. (2.1) and z is the Z-Transform operator. In the following, we show that the degree of the minimal polynomial of $[A, e_r^T]$ can be determined using Mason's rule [54].

Let $\Phi = (I - A/z)^{-1}$. If we build the signal-flow network for A/z , then from Mason's rule we can obtain the gain from node i to node j directly from the graph as follows:

$$\Phi[i,j] = \frac{1}{\Delta} \sum_{\text{path } p \in \mathbb{G}} T_p \Delta_p, \tag{2.34}$$

where Δ is the determinant of the graph, which can be computed by

$$\Delta = 1 - \sum L_i + \sum L_i L_j + \ldots + (-1)^m \sum \ldots$$

T_p is the gain of the p^{th} forward path from node i to node j, L_i is the loop gain of each closed loop in the graph, and $L_i L_j$ is the product of the loop gains of any two non-touching loops (i.e. loops with no common nodes). Δ_p is the cofactor value of Δ for the p^{th} forward path, with the loops touching the p^{th} forward path removed (i.e. the remaining graph when you have removed those parts of the graph that form loops while retaining the parts on the forward path).

The McMillan degree of $e_r^T \Phi$ can be directly computed from the network using Mason's rule in eq. (2.34). Furthermore, it can be seen that the McMillan degree of $e_r^T \Phi$ [107], i.e. the number of poles, is the same as degree of the minimal polynomial of $[A, e_r^T]$ obtained from method in Section 2.2 .

Sometimes the graph is rather complicated and therefore it is hard to compute the formula (2.34) from Mason's rule. In this case one might resort to some basic graphical information [33], e.g., the diameter of the graph, the number of nodes in the graph, etc., to obtain a rough estimate of the minimal number of steps. Based on this idea, we propose the following upper and lower bounds.

Proposition 2.5.1. *Consider the system in eq.* (2.1)*. The degree of the minimal polynomial of* $[A, e_r^T]$*, namely* $D_r + 1$*, is lower bounded by* $d_r + 1$*, where* d_r *is the longest path from node r to all other nodes, and upper bounded by n.*

Proof. The upper bound can be obtained directly using the Cayley-Hamilton theorem,

i.e. $D_r + 1 \leq n$. Therefore we only need to show the lower bound.

Suppose the minimal polynomial for $[A, e_r^T]$ is $q_r(t) = t^{D_r+1} + \alpha_{D_r} t^{D_r} + \ldots + \alpha_1 t + \alpha_0$. Since $e_r^T q_r(A) = 0$, we have:

$$\begin{bmatrix} \alpha_0 & \alpha_1 & \ldots & \alpha_{D_r} & 1 \end{bmatrix} \begin{bmatrix} e_r^T \\ e_r^T A \\ \vdots \\ e_r^T A^{D_r+1} \end{bmatrix} = 0. \tag{2.35}$$

From a graph-theoretical perspective, element $A^k[i,j]$ being 0 means there is no path from i to j with length k [33]. Meanwhile, note that the consensus is guaranteed if and only if the digraph is strongly connected (see [72]). Strong connectedness implies that there always exists an edge-following path from node r to any other node in the graph. Therefore we can pick the longest path, say from node r to node s with length d_r. Given these two facts, if $A^{d_r}[r,s]$ is nonzero, then eq. (2.35) implies that $D_r + 1 \geq d_r + 1$. □

Remark 2.5.1. *Proposition 2.5.1 proposes a fundamental limitation on the minimal number of steps (successive values of a node) needed to compute the final value based on the graphical definition of a network; though if the graph contains quite a number of feedback loops, the Mason's formula is hard to compute.*

External equitable partition

In this section, we shall further consider the following question: given an undirected network, can we directly identify the dominant node(s) from the graph without any algebraic computation?

We adopt definitions and notations from [23]. A partition of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as a mapping from vertices to subsets of vertices called *cells*: $\pi : \mathcal{V} \to \{C_1, \ldots, C_K\}$ where $C_i \subseteq \mathcal{V}$, $\forall i$. Let $Im(\pi)$ denote the image of π, i.e. $Im(\pi) = \{C_1, \ldots, C_K\}$ and $\deg_\pi(i, C_j)$ denote the node-to-cell degree. $\deg_\pi(i, C_j)$ characterises the number of nodes in cell C_j that share an edge with node v_i under partition π:

$$\deg_\pi(i, C_j) = \text{card}\left\{k \in \mathcal{V} | \pi(k) = C_j \text{ and } (i,k) \in \mathcal{E}\right\}.$$

We define $\pi^{-1}(C_i) = \{j \in \mathcal{V} | \pi(j) = C_i\}$, i.e. the set of nodes that are mapped to cell C_i.[5]

[5]Note that π is not a one-to-one mapping but a one-to-many mapping. However, we can still define a new function to map back from C_j to $\mathcal{V}$. We adopt this notation from [23].

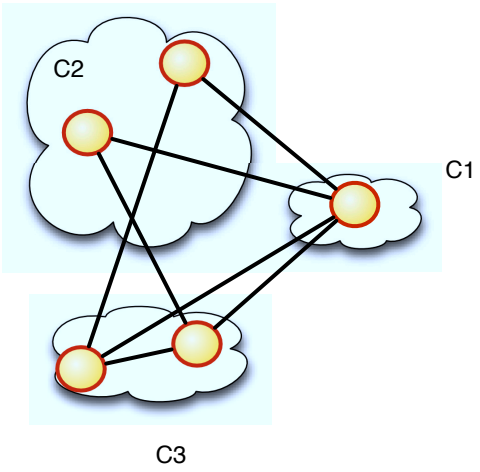


FIGURE 2.3: An example to illustrate EEP.

In what follows we use the concept of external equitable partition (EEP) [23]. As we shall show below, EEPs partition the graph into cells while neglecting the internal interconnection structure inside a cell. We shall show that the EEP with respect to a node is directly related to the minimal number of steps necessary for this node to calculate the final consensus value.

Definition 2.5.2 (External equitable partition (EEP) [23])**.** *A partition π^* of the set of nodes $\mathcal{V}$ consisting of $s > 1$ cells $\{C_1, \ldots, C_s\}$ is external equitable if the number of neighbours in C_j of a vertex $v \in C_i$ depends only on the choice of C_i and C_j $(i \neq j)$, i.e.*

$$deg_{\pi^*}(l, C_j) = deg_{\pi^*}(k, C_j), \forall k, l \in {\pi^*}^{-1}(C_i).$$

Definition 2.5.3 (Minimal EEP with respect to a node)**.** *A partition π_r of $\mathcal{V}$ consisting of cells $\{C_1, \ldots, C_s\}$ is external equitable with respect to node r if the partition is external equitable and the node r is in a cell alone, i.e. $\pi(v_r) = v_r$. The minimal EEP of a graph with respect to node r, π_r^*, is such that $card\{Im(\pi_r^*))\}$ is minimal.*

Example 2.5.3. *We here use a simple example to illustrate the above definitions. The partition on the graph in Fig. 2.3 is external equitable since different nodes in the same cell have the same degree to other cells. The partition is also an external partition with respect to the node in $C1$.*

Theorem 2.5.2. *Consider the system in* (2.1)*. Solely based on observations of node r, the minimal length of recursion necessary to obtain the final consensus value is equal to the*

number of cells s_r in π_r^*, the minimal external equitable partition with respect to node r, i.e.

$$D_r + 1 = card\left\{Im\left(\pi_r^*\right)\right\} \triangleq s_r. \tag{2.36}$$

Proof. Without loss of generality, let $r = 1$. We use a Breadth-First-Search (BFS) algorithm to label the cells, as follows. We start from node 1 (i.e. cell 1) and explore all the neighbouring cells. For each of those nearest cells, we consider their own neighbouring cells and so on, until we have labelled all the cells in the minimal EEP with respect to cell 1 (see the Appendix for an illustrative example of this BFS algorithm).

Consider now the block matrix obtained by permuting and partitioning A according to π_1^*, a minimal EEP with respect to node 1:

$$A_{\pi_1^*} = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1s_1} \\ A_{21} & A_{22} & \dots & A_{2s_1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{s_11} & A_{s_12} & \dots & A_{s_1s_1} \end{bmatrix}.$$

Here, $A_{ii} \in \mathbb{R}^{l_i \times l_i}$ contains the interconnections between any two nodes in cell C_i^* and l_i denotes the number of nodes in cell C_i^*. Hence, $l_1 = 1$ and $\sum_{i=1}^{s} l_i = n$. The off-diagonal submatrices $A_{ij} \in \mathbb{R}^{l_i \times l_j}$ contain the interconnections between nodes in C_i^* and C_j^*. In particular, we shall consider the following submatrices:

$$A_1 \triangleq A_{\pi_1^*}[2:n, 2:n]$$
$$f_1^T \triangleq A_{\pi_1^*}[1, 2:n] = \begin{bmatrix} A_{12} & \dots & A_{1j} & 0 & \dots & 0 \end{bmatrix}.$$

Note that there are only j neighbouring cells to cell 1, i.e. $A_{1(j+1)}, \dots, A_{1s_1} = 0$ for some $j > 1$.

The observability matrix (2.32) associated with the pair $[A_{\pi_1^*}, e_1^T]$ is:

$$\Omega = \begin{bmatrix} 1 & 0 & \dots & 0 \\ A_{11} & A_{12} & \dots & A_{1s_1} \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \dots & * \end{bmatrix}, \tag{2.37}$$

where $*$ is a placeholder representing a real value.

Let Ξ be the observability matrix associated with the pair $[A_1,\ f_1^T]$. According to [53, 23], the rank of the observability matrix is equal to the dimension of the following

span

$$\text{rank}(\Xi) = \text{dim-span}\left\{\begin{bmatrix}1_{r_2}\\0\\0\\\vdots\\0\end{bmatrix}, \begin{bmatrix}0\\1_{r_3}\\0\\\vdots\\0\end{bmatrix}, \ldots, \begin{bmatrix}0\\0\\0\\\vdots\\1_{r_{s_1}}\end{bmatrix}\right\}, \tag{2.38}$$

with $r_i = \text{card}\left\{C_i^*\right\}$. Hence,

$$\text{rank}(\Xi) = s_1 - 1,$$

from where it follows that

$$D_1 + 1 = \text{rank}(\Omega) = \text{rank}\begin{bmatrix}1 & 0 & \ldots & 0\\*\\\vdots & & \Xi & \\*\end{bmatrix}$$
$$= \text{rank}(\Xi) + 1 = \text{card}\left\{\text{Im}\left(\pi_1^*\right)\right\}.$$

□

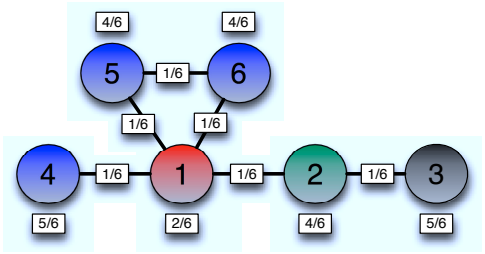
Remark 2.5.2. *Definition 2.5.3 implies that the number of cells in π_r^*, s_r, is greater or equal than the longest distance from node r to all other nodes in the graph $\mathcal{G}$, $d_r + 1$. Therefore*

$$D_r + 1 \geq d_r + 1.$$

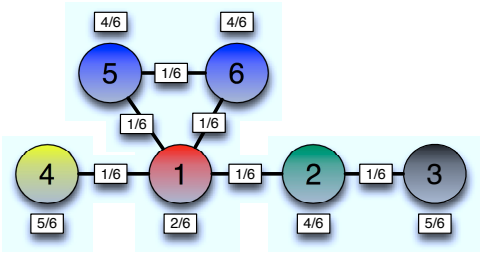
Remark 2.5.3. *Theorem 2.5.2 provides a link between local observations, i.e. the minimal number of successive values that a node r needs to accumulate to compute the final consensus value of the network) and a global property, i.e. the underlying minimal EEP of the network with respect to node r. Using this theorem, one can directly identify the dominant nodes in the network without resorting to algebraic numerical manipulations.*

Example 2.5.4. *As shown numerically in Example 2.4.1, nodes 1, 2 and 3 are the dominant nodes since they only require 8 steps, i.e. $D_r + 1 = 4$ for $r = 1, 2, 3$. It is easy to check in Fig. 2.4(a) that the minimal external equitable partition with respect to these nodes has 4 cells. Similarly, Figs. 2.4(b) and 2.4(c) show the minimal EEPs for node 4 and for nodes 5 and 6, respectively. The number of cells in the corresponding minimal EEPs is consistent with the numerical results in Example 2.4.1 which consistently indicates that these nodes*

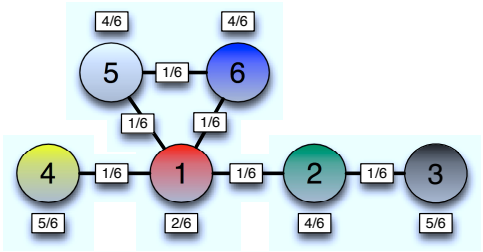
require respectively 10 *and* 12 *successive values of their own state to compute the final consensus value of the network according to Algorithm 2.*



(a) 4-cell based minimal external equitable partition with respect to nodes $1,2,3$. As illustrated in Example 2.4.1, nodes $1,2,3$ require $2\times 4=8$ steps to compute the final consensus value.



(b) 5-cell based minimal external equitable partition with respect to node 4. As illustrated in Example 2.4.1, node 4 requires $2\times 5=10$ steps to compute the final consensus value.



(c) 6-cell based minimal external equitable partition with respect to nodes $5,6$. As shown in Example 2.4.1, nodes $5,6$ require $2\times 6=12$ steps to compute the final consensus value.

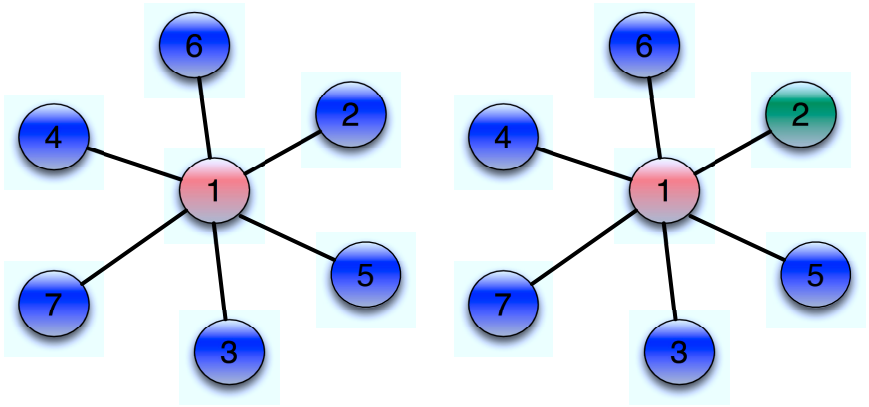
FIGURE 2.4: Minimal EEP with respect to the different nodes in Example 2.4.1. Different colours correspond to different cells (colour online).

2.6 Minimal-time consensus on special networks

In this section we apply the results of the previous section to characterise the mEEP of different types of node in special network topologies, e.g., star, line, and cycles, wheels, Mobius ladders, small world networks, regular graphs, scale-free networks in order to uncover what properties of the nodes make them "optimal" in terms of the minimal number of steps needed locally to compute the consensus value. This can serve as an indication of the design principles that need to be considered if one wants optimally to build a minimal-time consensus network.

2.6.1 Star and line graphs

We shall firstly show how the results can be applied to a star graph with n nodes. First of all, we see that the number of cells in the minimal EEP with respect to the centre node is 2, while it is 3 for the other nodes. Therefore the node in the centre of the star only requires 4 steps to compute the consensus value while all the other nodes at the periphery of the network require precisely 6 steps. It is worth mentioning that this is true irrespective of the number of nodes in the star-configured network. The reason for a generally small minimal number of steps is that the diameter of a star graph is small, 3.



(a) minimal EEP with respect to the centre node

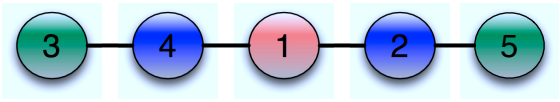
(b) minimal EEP with respect to a node at the boundary

FIGURE 2.5: minimal EEP with respect to different nodes in the star configuration, different colours are used to represent different cells.

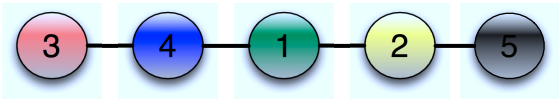
In the case of a line graph with n nodes, $D_r+1 = \lceil \frac{n-1}{2} \rceil + 1$ for the centre node $r = \lceil \frac{n-1}{2} \rceil$ and $D_r + 1 = n - 1$ when $r = 1, n$, i.e. a non-centre node. Again, the reason for a generally higher number of steps than for the star graph is that the diameter of a line graph is typically larger than that of a star graph. The nodes at the boundary need to wait more steps for the information to reach them than the centre node.

Remark 2.6.1. *Star graph seems to be an optimal topology in the sense that every node in such a network needs a small number of steps to compute the consensus value. However, it is not a robust network, because, any communication edge failure will lead to some node's failure to compute the consensus value.*

This raises a question, how to give a mathematical definition and solution of designing a robust consensus network. This will be done in the near future. We can define a



(a) minimal EEP with with respect to the centre node



(b) minimal EEP with respect to a node at the boundary

FIGURE 2.6: minimal EEP with respect to different nodes in the line configuration, different colours are used to represent different cells.

p-robust network, i.e. the failure of any p edges will not lead the isolation of any node and the analysis follows by this definition.

2.6.2 Cycle graphs

For cycle graphs every node needs the same number of steps of compute the final value due to graphical symmetry. The minimal EEP with respect to any nodes in the cycle graphs $D_r + 1 = \lceil \frac{n-1}{2} \rceil + 1$. Comparing with the line graph, with one additional link, it has shortened the number of steps requiring for boundary nodes to compute the final consensus value.

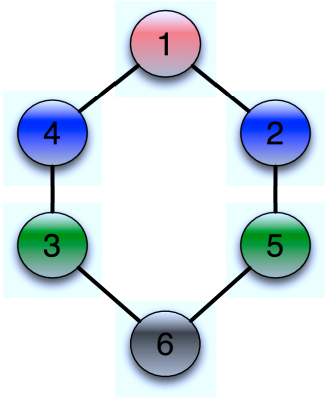
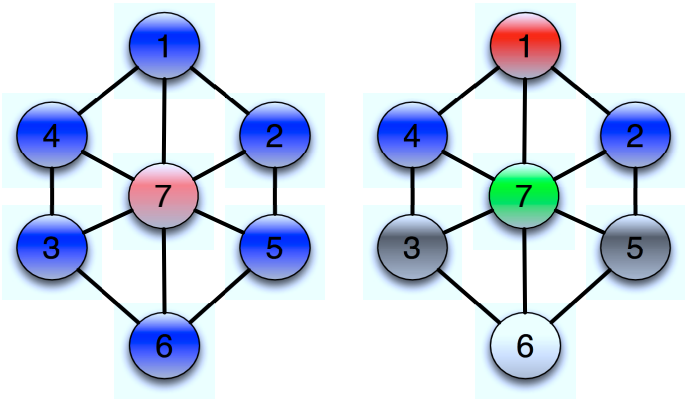


FIGURE 2.7: Minimal EEP with respect to different nodes in the cycle configuration, different colours are used to represent different cells.

2.6.3 Wheel graphs

For wheel graphs, which are a combination of star graphs and cycle graphs, there are two types of node. The mEEP with respect to the centre node is the same as in the star graph $D_r + 1 = 2$. The minimal EEP with respect to any nodes in the wheel graph $D_r + 1 = \lceil \frac{n-1}{2} \rceil + 1$.



(a) minimal EEP with with respect to the centre node

(b) minimal EEP with respect to a node at the boundary

FIGURE 2.8: minimal EEP with respect to different nodes in the wheel configuration, different colours are used to distinguish and represent different cells.

2.6.4 Möbius ladder

For Möbius ladder every node needs the same number of steps of compute the final value due to symmetry. The minimal EEP with respect to any nodes in the graph in $D_r + 1 = \lceil \frac{n-1}{2} \rceil + 1$. This problem can be cast as a special case of regular graph in the next section.

2.6.5 Small-world networks

Random graph theory was pioneered by Erdos-Renyi (ER) [24] and Gilbert [29]. Later, Watts and Strogatz introduced small-world networks [89] as a model that allows to interpolate between a regular lattice and a random graph based on only one parameter, the rewiring probability p [3, 60]. Almost at the same time, Barabasi and Albert proposed another model for complex networks called the scale-free network model [2]. Consensus within a network requires the underlying graph to be connected. Thus general ER

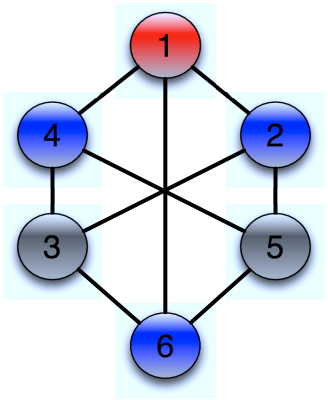


FIGURE 2.9: minimal EEP with respect to different nodes in the ladder configuration, different colours are used to represent different cells.

graphs and Gilbert graphs are not suitable for our analysis. We therefore focus on small-world networks and scale-free networks.

Recall that the minimal number of steps for node r to compute the final value is equal to 2 times the degree of the minimal polynomial of $[A, e_r^T]$: D_r+1, so we shall focus on the quantity D_r+1. A detailed simulation study of random graphs, small-world and scale-free networks [2] will be addressed in a future paper. Here we consider the following question for small-world networks: in a small-world network, can we characterise how the minimal number of steps varies as a function of the rewiring probability p? In particular, we focus on the worst-case scenario and consider the node with the maximal number of steps.

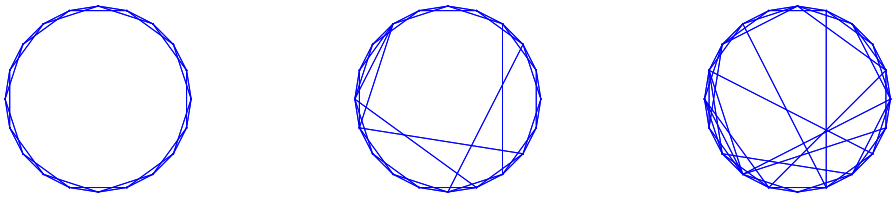
Definition 2.6.1. *We define the maximal value of the minimal number of steps needed in a given network over all nodes as* $2\hat{D}+2 = 2\max_r\{D_r\}+2$.

Remark 2.6.2. *Numerically, from Corollary 2.5.1, we can approximate* $\hat{D}+1 \approx dcard\{\Lambda\} = dcard\{\lambda_i(L) : i = 1:n\}$ *(the Laplacian spectrum). This approximation links the eigenspectrum of the Laplacian matrix and the decentralised characterisation in terms of overall minimal number of steps in a considered network.*

The following simulation study was conducted using MATLAB 2009b. The small-world network code was generated on [84]. Small-world networks [89] can usually be parameterised by $\mathscr{G}^{sw}(n,k=2d,p)$, where n is the number of nodes in the network, k is the degree of each node in the initial graph, and p is the probability of rewiring an edge. Random rewiring procedure is as follows [89] we start with a ring of n vertices, each connected to its k nearest neighbours by undirected edges. We choose a vertex and

the edge that connects it to its nearest neighbour in a clockwise sense. With probability p, we reconnect this edge to a vertex chosen uniformly at random over the entire ring, with duplicate edges forbidden; otherwise we leave the edge in place. We repeat this process by moving clockwise around the ring, considering each vertex in turn until one lap is completed. Next, we consider the edges that connect vertices to their second-nearest neighbours clockwise. As before, we randomly rewire each of these edges with probability p, and continue this process, circulating around the ring and proceeding outward to more distant neighbours after each lap, until each edge in the original lattice has been considered once.

One effect of increasing p is an increase of the algebraic connectivity of the Laplacian, as shown in the synchronisation literature [3] and in the consensus literature [60]. However, other dynamical properties of network dynamics are more involved [3].



(a) p=0 (b) p=0.5 (c) p=1

FIGURE 2.10: Small world $G^{sw}(20,4,p)$ for $p = 0,\ 0.5,\ 1$. For $p = 0$, $G^{sw}(20,2,0)$ is a regular network; for $p = 1$, $G^{sw}(20,2,1)$ becomes a random network.

Watts and Strogatz showed that the characteristic length $l(p)$ of a small-world network considerably reduces over the range $0.0001 \leq p \leq 0.01$ and remains almost unchanged for $p > 0.01$. This indicates that random rewiring with a small value of p creates a small-world network out of a regular network that originally has a large diameter.

We now present some numerical simulation results. For these simulations we consider regular networks ($p = 0$) small-world networks ($0 < p < 1$) and random networks ($p = 1$). In particular, we are interested in the following question: how does the maximal minimal number of steps $2\hat{D} + 2$ (we consider the worst case scenario) vary with respect to the probability p of rewiring an edge?

1. We fix $d = 2$ and let $n = 100, 110, 120, 150, 200, 250$. For each n, we compute the average worst case $\hat{D} + 1$ at different p over 10000 instances of the small-world model to see how $\hat{D} + 1$ varies with p (Fig. 2.11)

2. We fix $n = 100$ and let $d = 1, 2, 3, 4, 5, 6$. For each d, we compute the average worst

case $\hat{D}+1$ over 10000 random realisations for

$$p = [0, 0.0001, 0.0002, 0.001, 0.002, 0.01, 0.02, 0.1, 0.2, 0.5, 0.9, 1].$$

(Fig. 2.12)

Our simulation results show that for fixed n in the network and $k = O(log(n))$ the average maximal (worst-case) minimal number of steps of $\hat{D}+1$ is a monotonically increasing function of p. Fig. 2.11 and Fig. 2.12 also show that the more regular the graph, the lower the minimal number of steps needed for a randomly picked node.

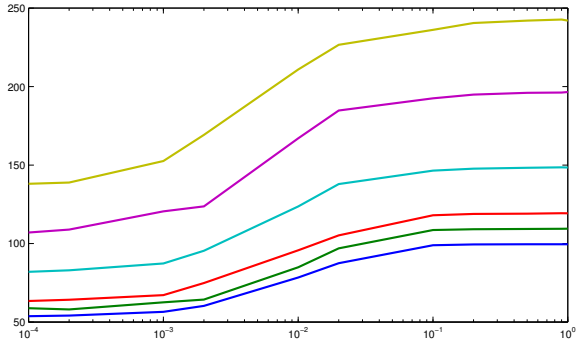


FIGURE 2.11: Small-world network. Average $\hat{D}+1$ from 10000 runs (y-axis) for $n = 100, 110, 120, 150, 200, 250$ versus the probability of rewiring, p (x-axis). Different curves correspond to different n.

More detailed analysis and simulations are currently under investigation.

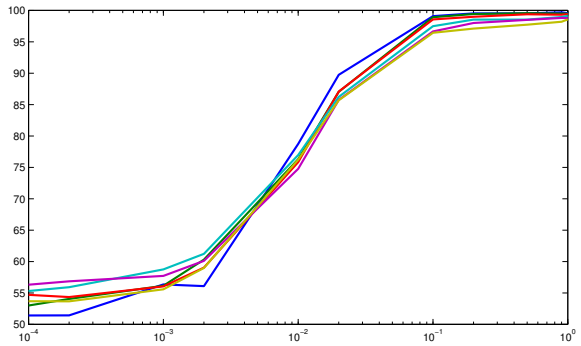


FIGURE 2.12: Small-world network. Average $\hat{D}+1$ from 10000 runs (y-axis) for $n = 100$ versus the probability of rewiring, p (x-axis). Different curves correspond to different k.

2.6.6 Regular graphs

In this section we explicitly characterise the minimal number of steps of a randomly picked node in a regular graph. A $\mathcal{G}^{sw}(n, k = 2d, p = 0)$ regular graph is defined as a n-node graph within which each node is connected to $k = 2d$ neighbours according to the adjacency matrix

$$W = \begin{bmatrix} c_0 & c_1 & \dots & c_{n-1} \\ c_{n-1} & c_0 & \dots & c_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \dots & c_0 \end{bmatrix}. \tag{2.39}$$

Similarly to [61], we define $c_{-l} = c_{n-l}$ for $l > 0$; $c_{\pm l} = 1,\ \forall 0 < l \leq d$; $c_l = 0,\ \forall\ d < l < n - d$; and $c_0 = 0$. Since for this topology the Laplacian is $L = kI_n - W$, it is easy to show that $\forall i,\ \lambda_i(L) = k - \lambda_i(W)$ and furthermore, L and W share the same eigenvectors for $\lambda_i(L)$ and $k - \lambda_i(W)$. Given this property, instead of considering the rank of the observability matrix of the pair $[A \triangleq I - \epsilon L, e_r^T]$, we can consider that of $[W, e_r^T]$. It is worth mentioning that for a regular graph, due to graphical symmetry, the minimal number of steps for all nodes is the same.

Theorem 2.6.1 (Minimal steps in k-regular graph)**.** *Consider the system in eq.* (2.1) *where the adjacency matrix W is given in* (2.39)*. The minimal number of steps for any node in the network is $2dcard\{\Phi\} + 2$, i.e. $D_r + 1 = dcard\{\Phi\} + 1,\ \forall r$ where*

$$\Phi = \left\{ \left. \frac{\sin\left(\frac{(k+1)\pi m}{n}\right)}{\sin(\frac{\pi m}{n})} \right| m = 1, \dots, n-1 \right\}.$$

Proof. Firstly we characterise the rank of observability matrix $\Omega = [e_r,\ We_r,\ ,\dots,\ W^{n-1}e_r]^T$. Since W is symmetric, W is diagonalisable and has a Jordan canonical from $W = SJS^{-1}$. In consequence, the observability matrix has the following decomposition

$$\Omega = \begin{bmatrix} S[r,:] \\ S[r,:]J \\ \ddots \\ S[r,:]J^{n-1} \end{bmatrix} S^{-1}. \tag{2.40}$$

Recall that S^{-1} is full rank and from [35]

$$\text{rank}(\Omega) = \text{rank}\begin{bmatrix} S[r,:] \\ S[r,:]J \\ \ddots \\ S[r,:]J^{n-1} \end{bmatrix} = \text{rank}\begin{bmatrix} 1 & 1 & \dots & 1 \\ J[1,1]=\phi_0 & J[2,2]=\phi_1 & \dots & J[n,n]=\phi_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ J[1,1]^{n-1} & J[2,2]^{n-1} & \dots & J[n,n]^{n-1} \end{bmatrix}.$$

It is easy to verify that $\phi_0 = d$. For $m \neq 0$,

$$\phi_m = \frac{\sin(\frac{\pi m}{n})}{\sin(\frac{\pi m}{n})}\sum_{t=1}^{d}\cos(\frac{2\pi t m}{n}) = \frac{\sin\left(\frac{(k+1)\pi m}{n}\right)}{\sin(\frac{\pi m}{n})} - 1 \triangleq \lambda_m - 1,$$

where second equality results from $\sin(\alpha+\beta)-\sin(\alpha-\beta) = 2\sin\alpha\cos\beta$. We therefore obtain

$$\text{rank}\begin{bmatrix} 1 & 1 & \dots & 1 \\ d & \lambda_1 - 1 & \dots & \lambda_{n-1}-1 \\ \vdots & \vdots & \ddots & \vdots \\ d^{n-1} & (\lambda_1-1)^{n-1} & \dots & (\lambda_{n-1}-1)^{n-1} \end{bmatrix} = \text{rank}\begin{bmatrix} 1 & 1 & \dots & 1 \\ \lambda_1 & \lambda_2 & \dots & \lambda_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \dots & \lambda_{n-1}^{n-1} \end{bmatrix} + 1.$$

The rank of this Vandermonde matrix is equal to the number of different values of λ_i for $i = 1,\dots,n-1$, which, in turns, is equal to the d-cardinality of Φ. Therefore $D_r + 1 = \text{dcard}\{\Phi\} + 1$. □

It is easy to see that for all m, the following equality holds

$$\frac{\sin\left(\frac{(k+1)\pi m}{n}\right)}{\sin(\frac{\pi m}{n})} = \frac{\sin\left(\frac{(k+1)\pi(n-m)}{n}\right)}{\sin(\frac{\pi(n-m)}{n})}.$$

Which implies that $2\text{dcard}\{\Phi\} + 2 \leq 2\lceil\frac{n-1}{2}\rceil + 2$.

If $n, k+1$ are not coprime, i.e. $(n,k+1) = a > 1$, then when $m = sn/a$, $\frac{\sin\left(\frac{(k+1)\pi m}{n}\right)}{\sin(\frac{\pi m}{n})} = \frac{\sin\left(\frac{(k+1)\pi s}{a}\right)}{\sin(\frac{\pi s}{a})} = 0$ for $s = 1,2,\dots,d$. This implies that

$$2\text{dcard}\{\Phi\} + 2 \leq 2\left\lceil\frac{n-d}{2}\right\rceil + 2. \tag{2.41}$$

Remark 2.6.3. *We failed to show that* $2dcard\{\Phi\} + 2 = 2\left\lceil\frac{n-d}{2}\right\rceil + 2$ *but conjectured the equality held based on a large number of examples. We are still trying to show this*

eq. (2.41) *using Chebyshev polynomials.*

For the cycle graphs and Möbius ladder in the previous sections, we can apply the theorem we had in this Section by letting $d = 1$ for the cycle case and $d = 2k$ for the Möbius ladder case.

2.6.7 Scale-free network

In scale-free networks [2], a mechanism called "preferential attachment" is proposed to explain the appearance of the power-law distribution. We are starting from the following seed matrix and at each step a new node is added to the network with connections to d nodes in the network. The new node links to an existing node with a probability that is proportional to the current degree of that node. We choose $d = 2$ and

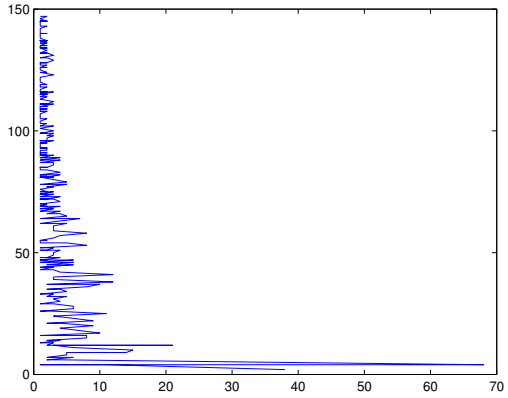
$$\text{Seed} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

We denote the final graph $\mathscr{G}^{sf}(n,d,seed)$.

One observation from Fig. 2.13(a) is that it has much smaller D_r+1 than the number of nodes in the network, i.e. n. And it is natural to ask whether the hub nodes would need fewer steps. From the above simulations we have Conjecture 2.6.1. In the meantime, we also have some observations for constructing counter-examples. From Fig. 2.14 and Fig. 2.15, we can easily see that the node (circled in yellow) that needs the smallest number of steps to compute the consensus value is not the hub node. A further interesting finding is that the number of cells in mEEP with respect to node r is independent of the number of connections of node r but somehow links to the social "role" of the node r. We the have the situation represented in Fig. 2.16!

Conjecture 2.6.1. *Given a scale-free network $\mathscr{G}^{sf}(n,d,seed)$, for a hub node i and a non-hub node j, i.e. the degree of node i is much larger than that of node j, then*

$$D_i + 1 \leq D_j + 1 \textit{ with high probability}.$$



(a) Scale-free network with 600 nodes

FIGURE 2.13: Scale-free network, degree of minimal polynomial for $[A, e_i^T]$ (y-axis) versus the degree of node i(x-axis).

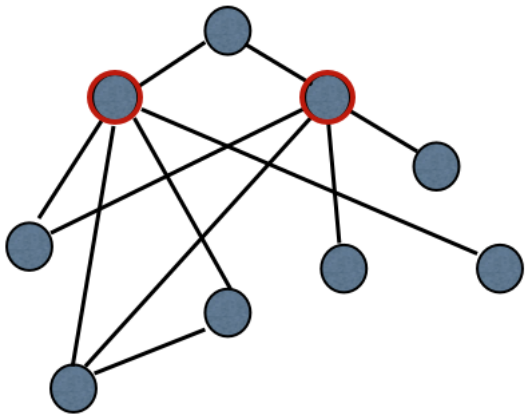


FIGURE 2.14: A example of scale-free network, the red circled node is the hub node in such network.

2.7 Application to dynamic consensus

From this section on, we shall consider extensions of proposed algorithm to deal with issues arising in practical applications.

Motivated by applications such as mobile networks and distributed Kalman filtering, we consider the scenario when the system has input signals instead of the standard static stimulus x_0 as in the previous sections. In this case, every node could receive a signal, e.g. ramp, sinusoid with possibly different amplitudes and frequencies. The goal of this consensus network is to reach a weighted average of these input signals for all

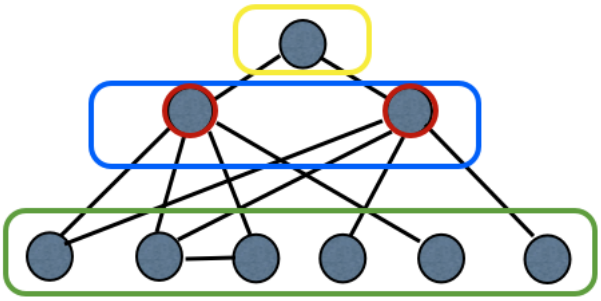


FIGURE 2.15: A example of scale-free network, the red circled node is the hub node in such network. However, by using mEEP with respect to different nodes, we found that the one that needs the smallest number of steps to compute the consensus value is not the hub nodes but the one circled in yellow.

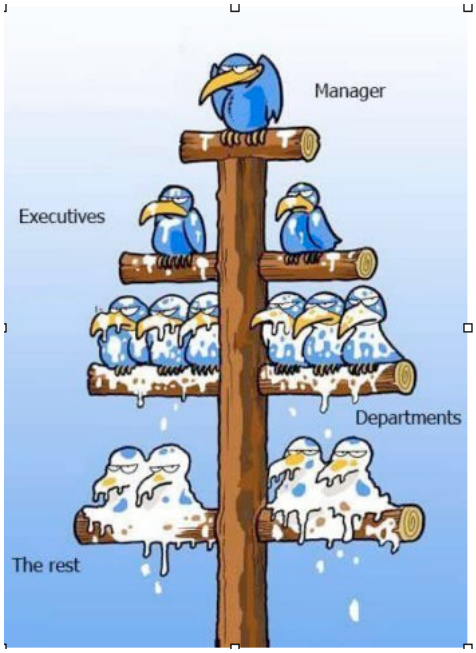


FIGURE 2.16: A joke to illustrate how the social role determines the time that one needs to know the consensus value of the network. When top level guys look down, they see only shit heads; When bottom level guys look up, they see only assholes...

nodes [80]. Hence, in contrast to the static consensus problem, consensus is reached in a dynamic sense. More specifically, assume that each individual i has an associated signal $S[i]$ with value $s_k[i]$ at different discrete-time step k. We define the vector s_k, which contains the individual $s_k[i]$ as its components. Dynamic consensus can be viewed as a situation in which all agents asymptotically track the evolution of some aggregate network quantity. In addition to the input signals $S[i]$, each agent maintains a local variable $x_k[i]$, which is a time-varying estimate of the instantaneous average value for node i at time k.

Definition 2.7.1 (Asymptotic dynamic consensus)**.** *System* (2.1) *is said to asymptotically achieve dynamic consensus with input signal* $S = [S[1],\ S[2],\ \ldots, S[n]]^T$ *if for any* $i,\ j$

$$\lim_{k\to\infty} |x_k[i] - x_k[j]| = 0. \tag{2.42}$$

Under the consensus protocol that ensures tracking of the consensus signal (see next section), we focus on an arbitrarily chosen node r and compute the consensus signal using a minimal number of its own observations. Note that additional assumptions on the input signals are needed to guarantee discrete-time consensus [108] since, for certain input signals, there may exist steady-state errors due to the poles introduced by the input signals. Fig. 1 in [80] shows a non-zero steady-state error for ramp inputs. We shall then derive such conditions in the next section.

2.7.1 System model for dynamic consensus

Consider the discrete time LTI dynamics in eq. (2.1) where an arbitrarily chosen state $x[r]$ is observed. The decentralised problem is to compute the dynamic consensus signal of the network ϕ_k at time step k using only its own previously observed values $y_k = x_k[r]$. We consider the extended model of eq. (2.1) instead of the standard static consensus problem. The dynamic consensus will track the weighted average signal Φ asymptotically.

This section proposes a dynamic version of decentralised consensus computation algorithm, i.e. it computes the final consensus signal using a minimal number of successive discrete-time observations of any node, say node r, in the network. In particular, we assume that node r in the network does not have access to any other external information about the input signals and the network, such as, the type of signals, total number of agents n in the network (2.1), its local communication links or even the state values/number of its neighbours.

Similar to static consensus protocol, consider the following decentralised protocol

$$\begin{aligned} x_{k+1} &= Ax_k + u_k, \\ y_k &= e_r^T x_k = x_k[r], \end{aligned} \tag{2.43}$$

where $x_0 = 0$ and u_k is the unknown input/disturbance. In most cases, $u_k = s_k$; for certain applications, we let $u_k = s_k - s_{k-1}$ to guarantee the consensus of some signals with pole at 1, for example, ramp.

$$\begin{aligned} x_{k+1} &= Ax_k + s_k - s_{k-1}, \\ y_k &= e_r^T x_k = x_k[r]. \end{aligned} \tag{2.44}$$

Let $E_{k+1} \triangleq x_{k+1} - \frac{1}{n}11^T s_k$, we have

$$E_{k+1} = AE_k + (I - \frac{1}{n}11^T)(s_k - s_{k-1}).$$

Take the Z-transform,

$$E(z) = (zI - A)^{-1}(I - \frac{1}{n}11^T)(1 - z^{-1})S(z). \tag{2.45}$$

Let $E(z) \triangleq X(z) - \frac{11^T}{n}(z)$ and assume the matrix A can be decomposed as

$$A = \sum_{i=1}^{n} \lambda_i(A) v_i v_i^T.$$

From the eigenvalue relation that $\lambda_i(A) = 1 - \epsilon\lambda_i(L)$ and the eigenvector corresponding to 0 is 1.

Let $H(z) = (zI - A)^{-1}(I - \frac{1}{n}11^T)$, and following a similar analysis as in [80], we know that the pole 1 of $(zI - A)^{-1}$ will be cancelled by multiplying $(I - \frac{1}{n}11^T)$. Then we use final value theorem to find out the final error

$$e_\infty = \lim_{z \to 1}(z - 1)E(z). \tag{2.46}$$

$e_\infty = 0$ if $E(z)$ has no pole at 1.

This system is reaching consensus asymptotically under various conditions on sampling time, underlying topology which are developed in [108]. In the following part of this section, we are assuming that these conditions for guaranteeing consensus are all satisfied. The goal of this section is not to develop new theory or condition for consensus,

but to focus on developing an algorithm that computes such dynamic consensus signal using minimal number of successive outputs.

Remark 2.7.1. *The information used in the proposed algorithm was solely based on the accumulation of successive state values of the agent under consideration. No further information about the network and signal is used.*

Remark 2.7.2. *More importantly, there does not exist criteria for any node in the network to check whether dynamic consensus is reached or not using merely its own observations. In other word, unlike static consensus problems, the node does not know whether dynamic consensus is reached or not. Hence, the proposed algorithm also provides a purely decentralised way to check dynamic consensus.*

For the purpose of main algorithm in Section 2.7.2, we need to impose the following assumption for the input signals, we will later consider when the assumption does not hold in [97].

Assumption 2.7.1. *The Z-transform of such input signal at each node must have a finite number of poles.*

Remark 2.7.3. *In engineered systems, there is a set of signals, e.g., step, ramp, sinusoid, that are commonly used. They all satisfy the above assumption. Alternatively, one may think of the scenario that these signals are the estimations of the same linear process by different nodes. Again, the assumption is still satisfied.*

Taking the Z-transform on both sides of eq. (2.44), $zX(z) = AX(z) + (1-z^{-1})S(z)$ leads to

$$\begin{aligned} Y(z) &= e_r^T X(z) = e_r^T (zI - A)^{-1}(1 - z^{-1})S(z) \\ &\triangleq y_0 + y_1 z^{-1} + \cdots . \end{aligned}$$

Let $\Phi(z) \triangleq \phi_0 + \phi_1 z^{-1} + \cdots$, where $\phi_k = \frac{1}{n} 1^T x_k$. $Y(z)$ is usually different from the consensus signal, but it has the property for consensus in that

$$\lim_{K \to \infty} \| y_K - \phi_K \| = 0 \tag{2.47}$$

2.7.2 Main algorithm for minimal-time dynamic consensus

From eq. (2.44) and [100], we have the following regression for the observations.

Proposition 2.7.1. *Given a linear system* (2.1) *and an input signal vector S satisfying Assumption* 1*, there exist a $d \in \mathbb{N}$ and scalars $\alpha_0, ..., \alpha_d$ such that the following linear regression equation must be satisfied $\forall k \in \mathbb{N} \geq 0$,*

$$y_{k+d+1} + \alpha_d y_{k+d} + \ldots + \alpha_1 y_{k+1} + \alpha_0 y_k = 0. \tag{2.48}$$

Proof. Taking the Z-transform on both sides of equation $zX(z) = AX(z) + (1 - z^{-1})S(z)$ leads to $Y(z) = e_r^T X(z) = e_r^T (zI - A)^{-1}(1 - z^{-1})S(z)$. By assuming that the number of poles of $S(z)$ is finite and noticing that $e_r^T (zI - A)^{-1}$ has finite poles [99], then the multiplication has finite poles and therefore it can be written in the form of eq. (2.48). The rest of proof is similar to the one to Theorem 2.4.1. □

Remark 2.7.4. *An algebraic characterisation of d for static consensus is given in [99] derived from the Jordan block decomposition. In the dynamic case, d is a function of input signal S and the matrix A.*

Remark 2.7.5. *If we can obtain the unknown coefficients in eq.* (2.48) *from data, then we can compute future outputs recursively from eq.* (2.48) *and past outputs. If we let $D_{r,s} + 1$ be the length of the regression in eq.* (2.48) *then Proposition 2.7.1 also indicates that some scalars $\alpha_0, ..., \alpha_{D_{r,s}}$, the following equation always holds:*

$$y_{k+D_{r,s}+1} + \alpha_{D_{r,s}} y_{k+D_{r,s}} + \ldots + \alpha_1 y_{k+1} + \alpha_0 y_k = 0. \tag{2.49}$$

To obtain such coefficients $\alpha_0, ..., \alpha_{D_{r,s}}$, we again resort to the Hankel matrix using successive outputs. A nice property of such a Hankel matrix is the following Kronecker Theorem.

Theorem 2.7.1. *[Kronecker Theorem][67] The Hankel matrix $\Gamma\{Y_{0,1,\ldots}\}$ has finite rank if and only if $f(z) \triangleq x_0[r] + x_1[r]/z + \cdots$ is a rational function with respect to z. The rank of the Hankel matrix $\Gamma\{Y_{0,1,\ldots}\}$ is equal to the number of poles of $f(z)$.*

Remark 2.7.6. *This theorem links the rank of a specially constructed matrix to the number of poles of a rational transfer function.*

We now present the main procedure for reconstructing the coefficients. Without loss of generality, assume that the outputs start from discrete-time step 0. It is easy to remove these assumptions, e.g. see [99]. From Proposition 2.7.1 and Kronecker Theorem, when increasing the dimension of this Hankel matrix it will eventually lose rank. When it does, at discrete-time step $k = 2D_{r,s} + 2$, where $D_{r,s}$ is defined in eq. (2.7), compute its

normalised kernel:

$$\Gamma(y_0,\ y_1,\ ,\cdots,\ y_{2D_{r,s}+2})\begin{bmatrix}\alpha_0 & \alpha_1 & \dots & \alpha_{D_{r,s}} & 1\end{bmatrix}^T = 0. \tag{2.50}$$

It can be shown that the normalised kernel obtained from eq. (2.50) corresponds to the coefficients in eq. (2.49) [99].

Define $d(z) = z^{D_{r,s}+1} + \sum_{i=1}^{D_{r,s}} \alpha_i z^i$ and let

$$n(z) \triangleq d(z)(y_0 + y_1/z + y_2/z^2 + \cdots).$$

Then by multiplication, we have $n(z) \triangleq \sum_{i=1}^{D_{r,s}} n_i z^i$.

From eq. (2.49), the coefficients from eq. (2.50) and past outputs we can predict future values of y_k, for all $k \geq 2D_{r,s}+3$. In addition, the explicit expression of $Y(z) = e_r^T(zI - A)^{-1}(1-z^{-1})S(z)$ can be obtained by taking the Z-transform of eq. (2.48).

To predict the future outputs of the observation at time K (it is actually the averaged signal under the assumption that the network reached consensus), we can use the expansion of a SISO transfer function and check the coefficient of z^{-K}.

We first define a reversion map $\mathscr{R}$ which takes z^{-m} to z^m, i.e., if $Y(z) = y_0 + y_1 z^{-1} + \cdots$, then $\mathscr{R}\{Y(z)\} = y_0 + y_1 z + \cdots$. It is not hard to show that

$$\mathscr{R}\left\{\frac{n(z)}{d(z)}\right\} = \frac{n(z^{-1})}{d(z^{-1})}.$$

Then y_K can be computed by the following equality

$$y_K = \frac{1}{K!}\left\{\frac{d^K \mathscr{R}\left\{\frac{n(z)}{d(z)}\right\}}{dz^K}\right\}_{z=0}, \tag{2.51}$$

when $Y(z)$ has a complicated form to take the K^{th} derivative, we could first use partial fraction expansion to decompose it to a summation of simple expressions, i.e., $Y = Y^1 + \cdots + Y^l$, where Y^is have only one or two poles. We can apply eq. (2.51) to Y^i to get y_K^i and then sum them up to get y_K

$$y_K = y_K^1 + \cdots + y_K^l.$$

Or alternatively, y_K can be computed iteratively using eq. (2.49).

The whole algorithm can be written as follows:

Algorithm 3 Decentralised minimal-time dynamic consensus value computation with input signal constraints

Data: Successive observations of $y_i = x_i[r],\ i = 0, 1, \cdots$.
Result: Final consensus signal at time K: ϕ_K.

Step 2.7.1. *Increase the dimension k of the square Hankel matrix $\Gamma\{Y_{0,1,\cdots,2k}\}$ until it loses rank and store the first defective Hankel matrix.*

Step 2.7.2. *The kernel $S = \begin{bmatrix}\alpha_0 & \dots & \alpha_{D_{r,s}} & 1\end{bmatrix}^T$ of the first defective Hankel matrix gives the coefficients of eq.* (2.48).

Step 2.7.3. *Compute $Y(z) = y_0 + y_1/z + \cdots$ and from then, we can compute $\phi_K \approx y_K$ (when K is large) using eq.* (2.51).

Example 2.7.1. *Let $Y_{0,1,2,3,\cdots} = (1,\ p,\ p^2,\ p^3,\ \cdots)$, then*

$$\Gamma(Y_{0,1,\cdots,2k}) = \begin{bmatrix} 1 & p & \cdots \\ p & p^2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

The Hankel matrix loses rank when $k = 1$. The normalised kernel of this Hankel matrix is $[-p,\ 1]^T$. Let $Y(z) = 1 + p/z + \cdots$ and $d(z) = z - p$, then $n(z) = Y(z)d(z) = z$, so $Y(z)$ writes

$$Y(z) = \frac{n(z)}{d(z)} = \frac{z}{z-p}.$$

For example, let $K = 3$, then

$$y_K = \frac{1}{K!}\left\{\frac{d^K \mathscr{R}\left\{\frac{n(z)}{d(z)}\right\}}{dz^K}\right\}_{z=0} = p^3,$$

which is consistent.

2.7.3 Examples

In this section we use examples to illustrate the results stated in the previous sections.

Example 2.7.2. *First we consider a simply connected 4-node network in [80] in Fig. 2.17(a). We then stimulate the network with ramp inputs of different magnitude. Consensus is reached in a dynamic and unstable sense in Fig. 2.17(c). Next we apply more complicated input signals and illustrate the above algorithm. Without loss of generality we assume*

we can access the observations of a node. Let $\beta_0 = \begin{bmatrix} 3.5784 & 2.7694 & -1.3499 & 3.0349 \end{bmatrix}^T$ *and* $\alpha_0 = \begin{bmatrix} 1/2 & -1/2 & 1/2 & -1/2 \end{bmatrix}^T$*, then for any input signal* i*,* $s_k[i] = k\beta_0[i]\alpha_0[i]^k$*. We stimulate the system with signals and use the local update protocol in eq.* (2.44)*. The trajectory of every nodes are plotted in blue whereas the consensus signal is plotted in red in Fig. 2.17(d). By applying the algorithm listed above step by step, we can obtain the consensus signal in eq.* (2.52) *using* 19 *successive observations.*

Step 2.7.1. *We start to increase the dimension* k *of the square Hankel matrix* $\Gamma\{Y_{0,1,\ldots,2k}\}$ *until it loses rank and store the first defective Hankel matrix.*

Step 2.7.2. *We found that when* $k = 9$ *the Hankel matrix loses its rank and then we can compute its kernel*

$$S = \begin{bmatrix} 0 & 0.0069 & -0.0069 & -0.1181 & 0.1181 & 0.6111 & -0.6111 & -1.0000 & 1.0000 \end{bmatrix}^T.$$

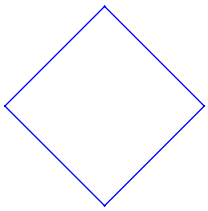
Step 2.7.3. *Compute* $Y(z) = y_0 + y_1/z + \ldots$ *and from then we can compute* y_K *using eq.* (2.51) *and finally approximate the final consensus signal at large time step* K*, i.e.* ϕ_K *as* y_K*.*

$$Y(z) = \frac{-2.952z^6 + 1.257z^5 + 1.647z^4 + 0.2067z^3 - 0.7575z^2 - 0.01911z + 0.01492}{z^7 - z^6 - 0.6111z^5 + 0.6111z^4 + 0.1181z^3 - 0.1181z^2 - 0.006944z + 0.006944}. \tag{2.52}$$

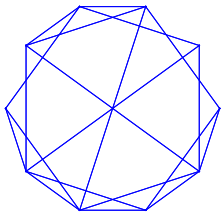
Example 2.7.3. *We then consider a more complicated 10-node random directed network (small-world network) [89] with* $\mathcal{G}^{sw}(n, 2d, p)$*, where* n *is the number of nodes in the network,* $2d$ *is the degree of each node in the initial graph, and* p *is the probability of rewiring an edge. We choose here* $n = 10, d = 1, p = 0.1$*, see Fig. 2.17(b). For input signals chosen as* $z_k[i] = \alpha_0[i] * k^{\frac{1}{5}}$*, with randomly chosen* x_0*, we can obtain the consensus signal using* 23 *successive observations.*

2.8 Consensus value computation with imperfect observations

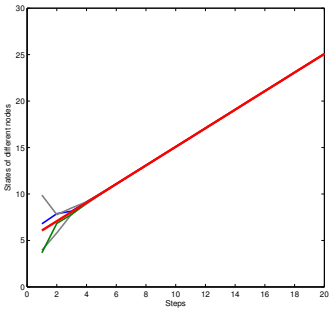
In the previous sections we proposed decentralised static and dynamic consensus computation algorithms on the assumption that local observations are perfect, i.e. without noise or quantisation or numerical error. In this section, we focus on generating a robust algorithm that incorporates uncertain observations. More specifically, assuming that



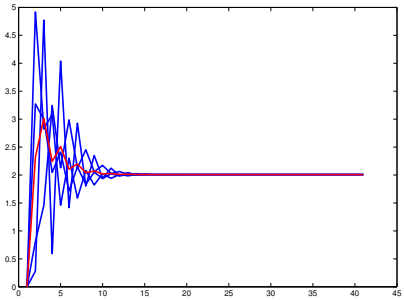
(a) 4-node ring.



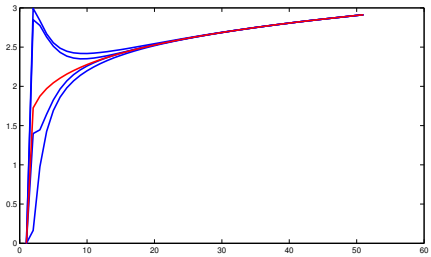
(b) 10-node small-world network.



(c) Trajectory of each node in Example 2.7.2, the red one is the consensus signal.



(d) Trajectory of each node in Example 2.7.2, the red one is the consensus signal.



(e) Trajectory of each node in Example 2.7.3, whereas the red one is the consensus signal.

FIGURE 2.17: Networks and node trajectories in Example 2.7.2 and Example 2.7.3 with 4-node ring and 10-node small world configuration respectively.

the conditions for consensus are satisfied, an agent observes its own state with uncertainty and tries, on successive uncertain observations of its own state, to compute the final consensus value of the network. At each time step, an agent may either propose an estimated final value computed from its own algorithm or wait to get more data from the next time-step. Generally, when uncertainties, i.e. unmodelled dynamics, noise or quantisation errors on observations, are taken into account the minimal number of steps cannot be found with certainty. This is due to the output measurement, being corrupted by uncertainties.

Consider a discrete-time LTI system with uncertainty

$$\begin{aligned} x_{k+1} &= Ax_k, \\ h_k &= e_r^T x_k + n_k = y_k + n_k. \end{aligned} \tag{2.53}$$

Here, h_k represents the observation with uncertainty at discrete-time k of an arbitrarily chosen particular state $x_k[r] \in \mathbb{R}$ and n_k denotes a zero-mean white noise for the randomly chosen state $x_k[r]$ at time k. Uncertainty might also lead to instability of the identified regression model (2.7), and therefore the estimated final value based on this identified model might be infinite. In this latter case no information about the final value can be inferred.

Consider system (2.53). At each discrete-time step the only information available is the output h_k which is the sum of the observed state $x_k[r]$ and of a zero-mean but statistically unknown Gaussian uncertainty. How can one quantify the error on the final value based on the data set with uncertainty and the data set without uncertainty?

Standard system identification techniques [51] require an infinite number of observations to have asymptotic performance, i.e. when the number of observations is infinite, it can guarantee a solution of the above problem. However, when the number of observations is limited and finite, there is no guarantee on the optimality of the identified model using a classic system identification procedure, e.g., subspace system identification.

The way we tackle this problem differs from the literature. A decentralised algorithm will be proposed in which at each time-step k, the successive observations $h_0,\ h_1,\ \ldots, h_k$ are used to compute an estimated consensus value of the whole network if a criterion on the quality of the data set is satisfied. On such criterion, some hard bounds on such an estimate can also be computed to show the confidence of such estimation of such predicted final consensus value.

2.8.1 Final consensus value estimation

The idea is to recover the length of the linear regression $D_r + 1$ and corresponding coefficients in eq. (2.3) from uncertain observations.

$$y_{k+D_r+1} + \alpha_{D_r} y_{k+D_r} + \ldots + \alpha_1 y_{k+1} + \alpha_0 y_k = 0, \forall k \in \mathbb{N}.$$

Technically, due to the effect of uncertainty, the identified model might have different lengths of recursion (usually much longer), different coefficients in the recursion, might not process a root at 1 exactly and even be unstable. Generally, the idea would be to put all the requirements in the constraints. This however would make optimisation hard. The challenging part of the problem is that the underlying regression has a number of constraints, e.g., the corresponding polynomial has a root at 1 and all other roots within the unit disk. The first idea comes to reduce the number of constraints and to make as optimisation problem solvable.

2.8.2 Averaging

In general, if $X_i \sim N(\mu_i, \sigma_i^2)$ for $i = 1, \ldots, n$ and they are independent, then

$$\sum_{i=1}^{n} a_i X_i \sim N\left(\sum_{i=1}^{n} a_i \mu_i, \sum_{i=1}^{n} (a_i \sigma_i)^2\right).$$

If all σ_is are equal, it is easy to show that variance of a weighted sum of independent variables is less than any of them. If we introduce a new variable as a function of outputs in eq. (2.53), let $q_i = 1/k \sum_{j=i}^{i+k} h_j$, then q_i also satisfies the same regression in eq. (2.3). This also means that the consensus value for this q_i is exactly the same as ϕ. By changing variables we now have a series of observations having the same consensus value and satisfying the same regression but with much smaller variances. With the above properties that we derived, we now use the successive observations of a new data set $\{q_i\}$ to predict the final consensus which ought to be the same.

2.8.3 Root 1 constraint

One of the constraints is that the z-transform of the following linear regression has a root at 1. This constraint is hard to incorporate in the identification process. In this section, we shall use mathematical manipulation to avoid this constraint.

$$q_{k+D_r+1} + \alpha_{D_r} q_{k+D_r} + \ldots + \alpha_1 q_{k+1} + \alpha_0 q_k = 0, \forall k \in \mathbb{N}.$$

The idea is simple: we can factorize the $(t-1)$ term out of the minimal polynomial and then the remaining part will not have a root at 1, i.e. $(t^{D_r}+\beta_{D_r-1}t^{D_r-1}+\ldots+\beta_1 t+\beta_0)(t-1) = t^{D_r+1}+\alpha_{D_r}t^{D_r}+\ldots+\alpha_1 t+\alpha_0$. To reconstruct the coefficients β_is, we introduce a new variable z and reduce the number of constraints by identifying the new data set $Z_{2k} = \{z_0 \triangleq q_1[r]-q_0[r],\ z_1,\ \ldots,\ z_{2k}\}$. The only constraint is that the identified recursion must be stable. Then

$$z_{k+D_r}+\beta_{D_r-1}z_{k+D_r-1}+\ldots+\beta_1 z_{k+1}+\beta_0 z_k = 0. \tag{2.54}$$

So far we have successfully relaxed the root 1 constraint.

2.8.4 Rank relaxation

Similarly, we want to construct a Hankel matrix using z_is. Such a Hankel matrix however will not lose rank at any finite dimension because of the uncertainties. Let $Z_{0,1,\ldots,2k} \triangleq \{z_0 = q_1-q_0,\ z_1 = q_2-q_1,\ldots,z_{2k} = q_{2k+1}-q_{2k}\}(k\in\mathbb{Z})$,

$$\Gamma(Z_{0,1,\ldots,2k}) = \begin{bmatrix} z_0 & z_1 & z_2 & \ldots \\ z_1 & z_2 & z_3 & \ldots \\ z_2 & z_3 & \ddots & \\ \vdots & \vdots & & z_{2k} \end{bmatrix}.$$

In this case, the idea is to find a Hankel matrix $\Gamma(\hat{Z}_{0,1,\ldots,2k})$ to approximate (close enough in some measure) $\Gamma(Z_{0,1,\ldots,2k})$. $\Gamma(\hat{Z}_{0,1,\ldots,2k})$ has finite rank and can therefore be used to estimate the final consensus signal. We formulate it as the following optimisation problem.

$$\begin{aligned} &\Gamma(\hat{Z}_{0,1,\ldots,2k}) = \operatorname{argmin}\|\Gamma(Z_{0,1,\ldots,2k})-\Gamma(\hat{Z}_{0,1,\ldots,2k})\|, \\ &\text{s.t.:}\det\Gamma(\hat{Z}_{0,1,\ldots,2k}) = 0, \Gamma(\hat{Z}_{0,1,\ldots,2k}) \text{ is Hankel} \end{aligned} \tag{2.55}$$

here $\|\cdot\|$ can be any norm, from the fact that

$$\begin{aligned} &\mathbb{E}\{(\Gamma(Z_{0,1,\ldots,2k})-\Gamma(\hat{Z}_{0,1,\ldots,2k}))^T(\Gamma(Z_{0,1,\ldots,2k})-\Gamma(\hat{Z}_{0,1,\ldots,2k}))\} \\ &= \Gamma(Z_{0,1,\ldots,2k})^T\Gamma(Z_{0,1,\ldots,2k}) + \text{Noise Cov. Matrix}, \end{aligned}$$

where $\mathbb{E}\{\cdot\}$ is the expected value, this means that the matrix 2-norm can be a good candidate measure for solving the problem. To solve the above problem we resort the following

lemma.

Lemma 2.8.1. *[74] Let $x \in \mathbb{R}^n$, then there exists a Hankel matrix $D \in \mathbb{R}^{n\times n}$, such that*

$$Dx = x \text{ and } \|D\|_2 \leq 1.$$

Proposition 2.8.1. *[38] Assume that the Hankel matrix $\Gamma(Z_{0,1,\ldots,2k})$ has full rank, then*

$$\begin{aligned} &\min \|\Gamma(Z_{0,1,\ldots,2k}) - H(k,k)\|_2 = \underline{\sigma}(\Gamma(Z_{0,1,\ldots,2k})) \\ &\textit{s.t.: } \det H(k,k) = 0, H(k,k) \textit{ is Hankel}. \end{aligned} \tag{2.56}$$

where $H(k,k)$ can be obtained by the following Algorithm 4.

Proof. Before referring to the algorithm we first define the hvec operator mapping from square Hankel matrix $\mathbb{R}^{n\times n}$ to a vector $\mathbb{R}^{(2n+1)\times 1}$. For example, $\text{hvec}(\Gamma(Z_{0,1,\ldots,2k})) = \begin{bmatrix} z_0 & z_1 & \ldots & z_{2k} \end{bmatrix}^T$. We now propose the algorithm for computing the nearest defective Hankel matrix with respect to $\Gamma(Z_{0,1,\ldots,2k})$. From Algorithm 4, we can see that $H(k,k)$ satisfies the constraints

Algorithm 4 Computing the nearest defective Hankel matrix

Step 2.8.1. *Form the observations as a square Hankel matrix, take a singular value decomposition of $\Gamma(Z_{0,1,\ldots,2k})$ and find the smallest singular value $\underline{\sigma}(\Gamma(Z_{0,1,\ldots,2k}))$ and corresponding singular vector $\underline{v}(\Gamma(Z_{0,1,\ldots,2k}))$;*

Step 2.8.2. *Compute the Hankel vector*

$$hvec(D) = C_x^+ C_x^T e_1,$$

where C_x^+ is the Moore-Pensore pseudoinverse of C_x, $e_1 = [1,\ 0,\ \ldots,\ 0]^T$ has length of $2k+1$ and

$$C_x = \begin{bmatrix} v[1] & \ldots & v[k] & v[k+1] & & \\ & \ddots & & \ddots & \ddots & \\ & & v[1] & \ldots & v[k] & v[k+1] \\ v[k+1] & & & v[1] & \ldots & v[k] \\ \vdots & \ddots & & & \ddots & \vdots \\ v[2] & \ldots & v[k+1] & & & v[1] \end{bmatrix}.$$

Step 2.8.3. *Let $\Gamma(\hat{Z}_{0,1,\ldots,2k}) = \Gamma(Z_{0,1,\ldots,2k}) - \underline{\sigma}(\Gamma(Z_{0,1,\ldots,2k}))D$.*

in the optimisation (2.55), because

1. by construction, $H(k,k)$ is Hankel;

2. it is easy to verify that the constructed Hankel matrix D satisfying $D\underline{v}(\Gamma(Z_{0,1,\ldots,2k}) = \underline{v}(\Gamma(Z_{0,1,\ldots,2k}))$, then

$$\begin{aligned} H(k,k)\underline{v}(\Gamma(Z_{0,1,\ldots,2k})) &= \Gamma(Z_{0,1,\ldots,2k})\underline{v}(\Gamma(Z_{0,1,\ldots,2k})) \\ &\quad - \underline{\sigma}(\Gamma(Z_{0,1,\ldots,2k}))D\underline{v}(\Gamma(Z_{0,1,\ldots,2k})) \\ &= 0. \end{aligned}$$

   In consequence, $H(k,k)$ does not have full rank;

3. since $H(k,k) - \Gamma(Z_{0,1,\ldots,2k}) = -\underline{\sigma}(\Gamma(Z_{0,1,\ldots,2k}))D$ and $\|D\|_2 \leq 1$, then

$$\|H(k,k) - \Gamma(Z_{0,1,\ldots,2k})\|_2 \leq \underline{\sigma}(\Gamma(Z_{0,1,\ldots,2k})).$$

Therefore we can choose $\Gamma(\hat{Z}_{0,1,\ldots,2k}) = H(k,k)$ as the solution of optimisation (2.55). □

Remark 2.8.1. *$\underline{\sigma}(\Gamma(Y_{0,1,\ldots,2k}))$ measures how good the approximation is. If it is large, then we probably need more observations to increase the dimension of the Hankel matrix for the purpose of a better approximation.*

Remark 2.8.2. *Due to the noise effect we can not show that $\underline{\sigma}(\Gamma(Z_{0,1,\ldots,2k}))$ is a monotonic function with respect to the number of the observations.*

2.8.5 Stability constraint

In the above section we found a close-to-defective Hankel. We can then follow similar steps in Algorithm 2 to compute the corresponding kernel of this Hankel matrix in question. However, the computed kernel (coefficients in regression (2.54)) might not map to a stable polynomial due to the noise effect. This leads to the invalidity of final value theorem.

In consequence it remains to consider this stability constraint for the computed kernel. In the literature, there are a number of attempts to impose the stability condition in the identification, [56] uses the Small-Gain Theorem [107] and $\mathscr{L}_1$ measure of distance [20] while a recent paper [15] imposes the Jury's stability criteria for the identified parameters. However, these conditions are hard to impose or computationally intractable. Here we shall solve the problem using Nehari's theorem[107].

Assume that $\Gamma(\hat{Z}_{0,1,\ldots,2k})$ satisfies the condition that its minimal singular value is less than a pre-determined small number at discrete time step $k = d$. By approximating $\Gamma(Z_{0,1,\ldots,2d})$ by $\Gamma(\hat{Z}_{0,1,\ldots,2d})$, we can then compute the kernel $[\gamma_0,\ \gamma_1,\ \ldots,\ \gamma_d,\ 1]^T$ of the latter matrix and map the coefficients to a polynomial. Furthermore, we compute the

roots of obtained polynomial mapping from the kernel $t^{d+1}+\gamma_d t^d+\ldots+\gamma_0=0$, namely, $\lambda_1,\ldots,\lambda_{d+1}$. From the obtained roots we construct a Vandermonde matrix (assuming the roots are distinct, otherwise we can construct a confluent Vandermonde matrix)

$$V(0,d)=\begin{bmatrix}1 & 1 & \ldots & 1\\ \lambda_1 & \lambda_2 & \ldots & \lambda_{d+1}\\ \vdots & \ddots & \ddots & \vdots\\ \lambda_1^d & \lambda_2^d & \ldots & \lambda_{d+1}^d\end{bmatrix}.$$

Without loss of generality, we rearrange the columns of V such that $\{|\lambda_i|\leq 1 : i=2,\ldots,k\}$ and $\{|\lambda_i|>1 : i=k+1,\ldots,d+1\}$.

The next step is to compute the diagonal matrix T, the defective property guarantees the diagonal property of T.

$$T\triangleq \operatorname{diag}\left\{(V(0,d))^{-1}\Gamma(\hat{Z}_{0,\ldots,2d})(V(0,d)^T)^{-1}\right\}=\begin{bmatrix}T_1 & 0\\ 0 & T_2\end{bmatrix},$$

where $T_1\in\mathbb{R}^{k\times k}$ and $T_2\in\mathbb{R}^{(d+1-k)\times(d+1-k)}$. We denote $\Gamma(\hat{Z}_{0,\ldots,2d})=V(0,d)TV(0,d)^T$.

We then need to find a stable approximation of $\Gamma(\hat{Z}_{0,\ldots,2d})$, i.e. another Hankel matrix $H'(d+1,d+1)=V'(0,d)T'V'(0,d)^T$ such that the Vandermonde matrix V' does not contain any λ_is greater or equal to 1 and also $\|\Gamma(\hat{Z}_{0,\ldots,2d})-H'(d+1,d+1)\|$ is minimal, where $\|\cdot\|$ is some norm.

Assuming $H'(d+1,d+1)=V_1'T_1'V_1'^T+V_2'T_2'V_2'^T$, after some manipulation we have

$$\|\Gamma(\hat{Z}_{0,\ldots,2d})-H'(d+1,d+1)\|=\|V_1T_1V_1^T+V_2T_2V_2^T-V_1'T_1'V_1'^T-V_2'T_2'V_2'^T\|.$$

For the stable part, we choose $V_1'=V_1$ and $T_1'=T_1$.

For the unstable part, we shall find a "stable" Hankel matrix[6] to approximate $V_2T_2V_2^T$. If we choose the norm to be the Hankel norm [67], we can map the Hankel matrix to a transfer function and use well-established Nehari's theorem (see for example Section 8 in [107]) to find the closest stable approximation of this unstable transfer function in terms of the $\mathscr{L}_\infty$ norm.

Given a Hankel matrix $H(d+1,d+1)\triangleq V_2T_2V_2^T$, we consider a one-to-one mapping from this Hankel matrix to a time-series data $\{g_0,\ g_1,\ldots,\ g_{2d},\ldots\}$. From Kronecker's Theorem 2.7.1 [67], this Hankel matrix has finite rank $d-k+1$[7]; let $G(z)\triangleq g_0+g_1z+$

[6] the polynomial mapped from its kernel is stable.

[7] since it contains $d-k+1$ unstable roots

$\ldots + g_{2d} z^{2d}$, then $G(z)$ is a rational function of z with degree $d-k+1$.

Since $G(z)$ is unstable, we are interested in finding a stable polynomial $L(z)$ such that $L(z) = \text{argmin}_{L \in \mathscr{RH}_\infty} \|L(z) - G(z)\|_\infty$. We turn the original problem into a standard model-matching problem in [22].

The algorithm to obtain $L(z)$ can be done using Nerahi's extension theorem [107], the procedure to obtain the coefficients of L: l_i is done iteratively.

Theorem 2.8.1. *[107] The minimal value of the above infinite norm can be computed by the largest singular value of*

$$\min_{L \in \mathscr{RH}_\infty} \|L(z) - G(z)\|_\infty = \|\hat{\Gamma} G(z)\|_\infty = \bar{\sigma} \left\{ \begin{bmatrix} g_{2d} & g_{2d-1} & \cdots & g_0 \\ g_{2d-1} & g_{2d-2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_0 & 0 & \ldots & 0 \end{bmatrix} \right\}. \tag{2.57}$$

The next question is how to obtain the coefficients in L. First choose l_0 to minimise

$$l_o = \text{argmin} \left\| \begin{bmatrix} * & * & * & ** \\ * & * & * & ** \\ g_3 & g_4 & g_5 & ** \\ g_2 & g_3 & g_4 & ** \\ g_1 & g_2 & g_3 & ** \\ g_0 - l_0 & g_1 & g_2 & ** \end{bmatrix} \right\|$$

This can be viewed as a matrix dilation problem solved by Parrott's Theorem [107]. More specifically, we refer to the following lemma.

Lemma 2.8.2. *[Parrott's Theorem] The solution to the following problem:*

$$\gamma_0 = \min_X \left\| \begin{bmatrix} X & B \\ C & A \end{bmatrix} \right\|$$

is given by

$$\gamma_0 = \max \left\{ \left\| \begin{bmatrix} C & A \end{bmatrix} \right\|, \left\| \begin{bmatrix} B \\ A \end{bmatrix} \right\| \right\}.$$

From above Lemma we can further write

$$\min \left\| \begin{bmatrix} * & * & * & ** \\ * & * & * & ** \\ g_3 & g_4 & g_5 & ** \\ g_2 & g_3 & g_4 & ** \\ g_1 & g_2 & g_3 & ** \\ g_0 - l_0 & g_1 & g_2 & ** \end{bmatrix} \right\| = \left\| \begin{bmatrix} * & * & * & ** \\ * & * & * & ** \\ g_3 & g_4 & g_5 & ** \\ g_2 & g_3 & g_4 & ** \\ g_1 & g_2 & g_3 & ** \end{bmatrix} \right\|$$

by properly choosing l_0 from a constructive proof of Parrotts' Theorem.

Once l_0 has been solved, we then solve l_1 similarly

$$l_1 = \text{argmin} \begin{bmatrix} * & * & * & ** \\ g_3 & g_4 & g_5 & ** \\ g_2 & g_3 & g_4 & ** \\ g_1 & g_2 & g_3 & ** \\ g_0 - l_0 & g_1 & g_2 & ** \\ -l_1 & g_0 - l_0 & g_1 & ** \end{bmatrix}.$$

Once l_1 is solved, we can substitute it in and solve for l_2 and so on. In this way, we can solve for all l_is iteratively, and we also put all the computed l_is as a Hankel matrix

$$\Gamma(l_0,\, l_1,\, \ldots,\, l_{2k}) = \begin{bmatrix} l_0 & l_1 & l_2 & \ldots \\ l_1 & l_2 & l_3 & \ldots \\ l_2 & l_3 & \ddots & \\ \vdots & \vdots & & l_{2k} \end{bmatrix}. \tag{2.58}$$

From Proposition 2.7.1, when increasing the dimension of this Hankel matrix it will eventually lose rank. When it does, we compute its normalised kernel to get the regression for l_is. In addition, the explicit expression of $L(z)$ can be obtained by taking the Z-transform of the obtained regression.

The explain of the above analysis is as follows: Let $V(z) = v_0 + v_1 z + \ldots$ represent the stable part, we actually approximate for all discrete-time step k, we decompose the noisy signal $y_k = v_k + g_k + O(\epsilon) \approx \hat{y}_k = v_k + l_k$ where the approximation holds because

$$\|G(z) - L(z)\|_{\mathscr{L}_\infty} \leq \epsilon. \tag{2.59}$$

This means that, where we allow a ϵ neighbourhood of every observation, we can find

another sequence of observations (data set) that satisfies all the conditions. We can thus compute the final value of this obtained data set of $\hat{y}$ (since it satisfies every conditions) $\hat{\phi}$ from eq. (2.6) and take this value as an approximation for $\phi = \lim_k x_k[r] \approx \hat{\phi}$.

Remark 2.8.3. *We are trying to meet the requirements step by step under the constraint that we only make small changes to the data set guided by the measurement of some norm. The distance from the original data set to the estimated stable data set is bounded by $O(\epsilon)$.*

2.8.6 Compute bounds for final consensus value estimation

In Section 2.8.4 and 2.8.5, we proposed an algorithm that firstly computes a defective Hankel matrix $\Gamma(\hat{Z}_{0,1,\ldots,2k})$ close to $\Gamma(Z_{0,1,\ldots,2k})$. And it then computes a stable Hankel matrix $\Gamma(\bar{Z}_{0,1,\ldots,2k})$ close to $\Gamma(\hat{Z}_{0,1,\ldots,2k})$. Finally it uses $\Gamma(\bar{Z}_{0,1,\ldots,2k})$ to compute the estimated final consensus value. By applying Theorem 2 in [98] we can show that when there is some change (by noise) in the root of obtained polynomial (e.g. from λ_j to λ'_j), the final consensus value estimation error is at most $O(\lambda_j - \lambda'_j)$. This implies that a small error in the root will lead to a small error in the estimation. However, it remains to show the guarantee of estimated final consensus value. More specifically, how far it is away for the true consensus value.

In this section, we shall briefly list the idea of computing optimal bounds for the final consensus value estimation. Here the word "optimal" means that given the identified model and the available noisy observations h_is, the bounds for the final consensus value is the tightest using the algorithm for linear system in [105]. From the identified regression model identified from Section 2.8.1 to Section 2.8.5, we then determine how precise the identified consensus value is. After obtaining $\alpha_0, \ldots, \alpha_d \in \mathbb{R}$ such that the following linear regression equation is satisfied $\forall k \in \mathbb{N}_{\geq 0}$

$$\hat{y}_{k+d+1} + c_d \hat{y}_{k+d} + \ldots + c_1 \hat{y}_{k+1} + c_0 \hat{y}_k = 0. \tag{2.60}$$

Write this in a matrix form

$$\begin{aligned} \kappa_{k+1} &= M\kappa_k, \\ \hat{y}_k &= \kappa_k[r], \\ h_k &= \hat{y}_k + d_k \end{aligned} \tag{2.61}$$

where $\kappa_k = \begin{bmatrix} \hat{y}_{k+d} & \hat{y}_{k+d-1} & \dots & \hat{y}_k \end{bmatrix}^T$ and M has the following canonical form

$$M = \begin{bmatrix} -c_d & -c_{d-1} & c_{d-2} & \dots & -c_0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}.$$

We shall apply the algorithms in [105] to compute the optimal bounds for the final value of this iteration (2.60). As a final remark of this section, we are working on an expanded version of the result in this section in which we shall illustrate every step in the proposed algorithm [104].

2.9 Discussion

We recently found that some steps in Algorithm 3 in this Chapter have some links to the realisation theory of SISO transfer functions in the 1960s, namely, the Kalman-Ho's algorithm [40] and the Silverman's algorithm [77]. However, the differences are subtle and will be explained as follows:

Kalman-Ho's algorithm, Silverman's algorithm Some steps in our proposed decentralised consensus value computation Algorithm 2 is very similar to the Kalman's work in [45][8]. This is due to the fact that fundamentally, we are solving for the parameters in a set of linear equations. We are interested in finding out what is the minimal number of successive impulse responses that suffices to compute the corresponding transfer function and therefore the future impulse responses. However Kalman-Ho's and Silverman's algorithms in [40, 77] focused on obtaining the minimal realisation (a realisation that is both controllable and observable) using infinite number of impulse responses and Hankel matrix decomposition.

However, Kalman-Ho's algorithm is hard to implement in practice due to numerical error. More specifically, when trying to factor a large Hankel matrix using singular value decomposition, it is hard to select a threshold to do truncation on the singular values. Our algorithm starts from low dimension matrices and of course, standard rank computation algorithm performs.

Later, Silverman continued pursuing this idea and in a paper with Glover [31] they

[8] I declare that the results in Algorithm 2 is my original work (preliminary results and more detailed derivations can be found in [95]) and therefore independent of previous algorithm as we mentioned here

studied the structural controllability (observability) problem, i.e. given the network structure and the structure of inputs, whether the structured controllability Grammian has full rank over almost all possible nonzero real parameters. This can be checked using graph-theoretical notions without resorting to complicated algebraic computations.

We have shown that

$$D_r + 1 = \text{rank} \begin{bmatrix} e_r^T \\ e_r^T A \\ \vdots \\ e_r^T A^{n-1} \end{bmatrix} \leq n,$$

assuming that n is the dimension of A. Basically, the structural controllability can be re-interpreted as given the network structure of A and the structure of observation, i.e. e_r^T and use the result in [31], we can check whether the following equality holds graph-theoretically

$$\begin{bmatrix} e_r^T \\ e_r^T A \\ \vdots \\ e_r^T A^{n-1} \end{bmatrix} = n.$$

In this thesis, we also related the minimal number of steps that needed to compute the transfer function to the rank of Grammian and further a graphical quantity, i.e. mEEP. In our case, we wish to minimise the rank of observability Grammian. Given the network structure of A and the structure of observation, i.e. e_r^T, there exists at least a set of parameters such that the rank of observability Grammian is equal to $D_r + 1 \leq n$. However, how to minimise $D_r + 1$ over all possible nonzero real parameters is still puzzling. This actually links a open problem in linear algebra and graph theory communities, i.e. the minimal rank problem [41].

Kung's algorithm When the impulse response is corrupted with noise, [46] proposed a solution to this problem using singular value decomposition and truncate the smallest singular value(s) if it is close to 0. The standard impulse to state-space function in Matlab $imp2ss$ uses this result. However, it is easy to know that the singular value decomposition will not preserve the Hankel matrix structure, i.e. the matrix after truncation will no longer be Hankel. Comparing with eq. (2.55), [46] basically did the following optimisation

$$\begin{aligned} &\Gamma(\hat{Z}_{0,1,\ldots,2k}) = \text{argmin} \|\Gamma(Z_{0,1,\ldots,2k}) - \Gamma(\hat{Z}_{0,1,\ldots,2k})\|, \\ &\text{s.t.:} \det \Gamma(\hat{Z}_{0,1,\ldots,2k}) = 0, \end{aligned} \tag{2.62}$$

but loosed the constraint that $\Gamma(\hat{Z}_{0,1,\ldots,2k})$ is Hankel.

Though the minimal value of the distance function in eq. (2.62) and eq. (2.55) are the same (smallest singular value of $\Gamma(Z_{0,1,\ldots,2k})$), the solutions $\Gamma(\hat{Z}_{0,1,\ldots,2k})$ in eq. (2.62) and eq. (2.55) are very different. However, the solution in Kung's algorithm could not map the truncated matrix back to the impulse response and see how much one changed at each steps.

Comparison Chen, Ohlsson and Ljung recently published a paper about how to estimate transfer function from finite number of noisy impulse responses using regularised least squares [16]. Future work lies in the comparison between proposed algorithm in Section 2.8 and [16].

2.10 Conclusion, and future works

This chapter formulates and analyses the decentralised minimal time consensus problem. Unlike other tools in the literature, our algorithm computes consensus from the history of any node in a completely decentralised, local manner. The necessary information for any node is its own history and is therefore exclusively local. The proposed algorithm does not require global knowledge, such as the total number of nodes in the system, information about the neighbourhood of the node, or specific edge weights. After characterising the minimal number of steps required for any given node to compute the final consensus value, we provide algebraic, graph-theoretical and local informative interpretations of the minimal number of steps. Furthermore, we examine the mEEP with different nodes in a variety of network structures. We also consider the extensions of the proposed algorithm to other practical problems.

Beyond the directions mentioned in the chapter and previous section, there are still a large number of interesting directions for future research.

1 **Theoretical**:

- 1.1 Optimal network structure design for minimal minimal-time consensus. It is important to mention that the EEP-based results provided here for undirected graphs can be extended to directed graphs at the price of a more elaborate exposition. Therefore, we ask: Given a constraint on the number of edges in the network, what are the network structures that minimise the d-cardinality of the Laplacian spectrum?
- 1.2 Extension to nonlinear dynamics, time-varying dynamics, in the consensus

model (2.1). Extension to nonsuccessive observations y_is in eq. (2.1) caused by packet-drop in a communication channel.

1.3 Robustness. How can we take into account robustness issues such as node or edge failures in a consensus network?

2 **Computational**:

2.1 There are a number of interesting directions for studying complex social networks. For example, based on the polynomial-time algorithm [58] to find a mEEP with respect to a node in a given network, we can now study the real networks, for example citation network [47], cellphone network [78], or social network to search for "social leaders".

2.2 We are implementing the proposed algorithm in the rendezvous problem in multi-robot systems. For example, most robots only have a single-chip microcomputer for computations and limited memory to store the observations. How to incorporate such limitations to improve numerical algorithms for Hankel matrices is currently under investigation.

3 **Application**: In general, the proposed minimal-time consensus computation algorithm can be embedded in the distributed optimisation/ Kalman filtering/ computation to improve performance, i.e. the time needed to know the global consensus value. For application and commercialisation, we are currently working on embedding our minimal-time consensus computation algorithm in the following algorithms: distributed Kalman filtering [59] in sensor networks, rendezvous in multi-robot systems, Google Pagerank algorithm[9] [11], load balancing algorithm of a huge number of computers (Amazon for example) [18] to dramatically increase the performance in the these algorithms. Preliminary results showed that our algorithm outperforms the best distributed Kalman filtering which is widely used in the sensor network by Olfati-Saber [59] and at the same time relaxing some key assumptions, e.g. the restriction on the sampling time.

[9]Preliminary result shows that variations of algorithms proposed in this Chapter improve the speed of obtaining Pagerank vector dramatically.

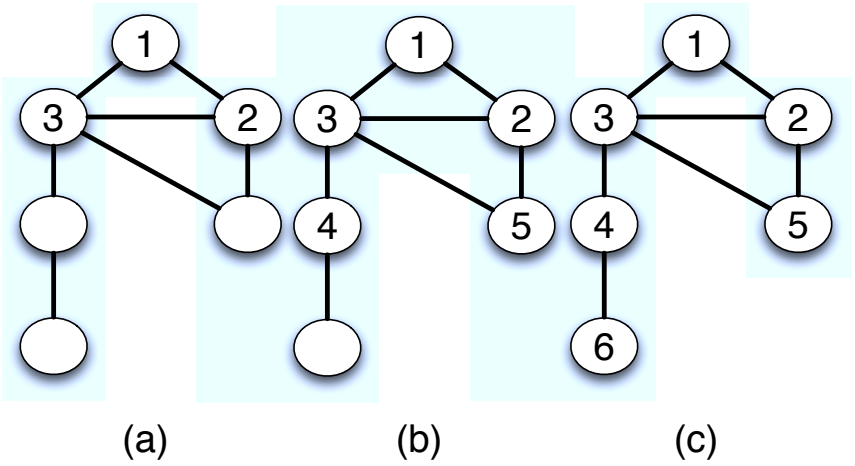


FIGURE 2.18: Step by step labelling of the cells using the BFS algorithm.

2.11 Appendix

2.11.1 BFS algorithm

We hereby give an example of the application of the Breadth-First-Search (BFS) algorithm that we mentioned in the proof of Theorem 2.5.2.

Consider the unlabelled network described in Fig. 2.18. We arbitrarily choose node 1 to be the top node on this topology. The BFS algorithm can now be used to label all the other nodes in the graph. To do so, we start to label the nodes which are direct neighbours of node 1, and therefore place labels 2 and 3 as in Fig. 2.18 (a). Then we continue to search for the unlabelled neighbouring nodes of nodes 2 and 3 and label them nodes 4 and 5 respectively (see Fig. 2.18 (b)). Finally, we add label 6 to the remaining unlabelled neighbour of node 4 (see Fig. 2.18 (c)).

Chapter 3

Network reconstruction

In the previous chapter we considered how to predict the final consensus value of a network based on the successive observations of a single node. In this chapter, assuming that the available information are the observations of a number of nodes (a subset of nodes) in the network, what can we get in addition to predicting the final consensus value [96]? An important question is how to use such observations to infer the interconnections between the measured nodes. Later we consider an even more ambitious one: Can we obtain any global information about the network (e.g., number of nodes in the network, how nodes interconnect) from such local measurements as motivated by Blake's lines *"To see a world in a grain of send, And a heaven in a wild flower"*.

3.1 Introduction

In this chapter we shall pursue these questions in the framework of network reconstruction, more specifically, biological network reconstruction. It is one of the fundamental interests in systems biology [1, 4]; since these mechanisms (unknown networks) are composed of complex networks of reactions between various chemical species. Most biologists believe that there is a strong link between the structure and the functionality of a biological system [1]. On the other hand, detailing the species and interactions forming this network can be an overwhelming task.

The biological network reconstruction problem challenges come from the necessity to deal with noisy and partial measurements (in particular, the number of hidden/unobservable nodes and their location and connections in the network are unknown) taken from a nonlinear and stochastic network. Even in the ideal situation where the underlying network is assumed to be linear time-invariant (LTI) and the measurements are assumed to be

non-noisy, it can be shown that, due to partial observability, this problem is unsolvable using classical system identification techniques [34]. In particular, identification of the system transfer function (obtained, for example, using classic system identification approaches) is useless to solve the network structure reconstruction problem since transfer functions do not contain sufficient information for that purpose.

There are a number of different representations of an LTI dynamical system. Each representation reveals different information about the system [93], and thus each also requires different amounts of data, possibly different kinds of data, as well as different computational procedures to be specified from these data. Here we discuss transfer functions, state space realisations, and dynamical structure and use them to contrast the identification, realisation, and reconstruction problems.

The transfer function of a LTI system is a description of the input-output behaviour of the system. It provides a model capable of simulation. Knowing the transfer function allows one to predict the response of the system to a new input. *System identification* [51] uses time-series data characterising stimulus and response to determine the input-output dynamics of the system. Developing reliable algorithms capable of operating on real (i.e. noisy and finite) data is a research topic in its own right. The system identification process reads time series data to produce a mathematical expression characterising the dynamic input-output behaviour of the system; in linear time-invariant systems this expression is the transfer function. For a system with m inputs and p outputs, this input-output map is a $p \times m$ matrix of transfer functions that we also call the transfer function, allowing any ambiguity to be resolved by context. In this work we denote this transfer function matrix as $G(s)$ and note that each individual element of G is a proper rational function of the Laplace variable s, a function of a complex variable.

In contrast to a transfer function, which describes only the input-output behaviour of an LTI system, a state space model of the system describes the precise network architecture used to realise a particular input-output behaviour. This representation of the system is a set of coupled ordinary differential equations of the form

$$\dot{x}(t) = Ax(t) + Bu(t)^1$$

for an $n \times n$ matrix A, an $n \times m$ matrix B, a vector of m inputs, u,and in which the dot notation indicates a derivative with respect to time, t. In this work, we assume that $p < n$ states are measured, leading to the additional output equation $y(t) = Cx$, where $C = \begin{bmatrix} I & 0 \end{bmatrix} x$ and in which I is the $p \times p$ identity matrix, and 0 is the $p \times (n-p)$ matrix of

[1]The notations in this Chapter are independent of those in the previous Chapter.

zeros. This output equation thus indicates that the first p elements of the state vector x are exactly the measured variables in the system; the remaining $(n-p)$ state variables are unmeasured “hidden" states. The zero structure of the A and B matrices exactly describe the Boolean structure of the network, and the values of these matrices encode the dynamics of the system.

In general there are many state space realisations that generate the same input-output behaviour, or transfer function. The process of finding a state space model that produces a given transfer function is called *realisation*, and a state space model derived from a given transfer function is called a realisation of the transfer function. There is a well-developed realisation theory for linear systems that answers questions such as the minimum number states, n, needed to describe a given transfer function, $G(s)$; how to find state space realisations of particular canonical forms; and how to relate (A, B, C) to $G(s)$. This theory helps us understand that generally more information, beyond the input-output data used to generate a transfer function, is needed to prefer one state space realisation over another as a description of a particular system.

Even with just one hidden state, the realisation problem becomes ill posed; a transfer function will have many state space realisations, and each of these may suggest an entirely different network structure for the system. This is true even if it is known that the true system is, in fact, a minimal realisation of the identified transfer function. As a result, failure to explicitly acknowledge the presence of hidden states, and the ambiguity in network structure that results, can lead to a deceptive and erroneous process for network discovery.

Based on this latter observation, a new representation for LTI systems, called dynamical structure functions was introduced in [34]. Dynamical structure functions capture information at an intermediate level between transfer function and state space representation (see Figure 3.1). Specifically, dynamical structure functions not only encode structural information at the measurement level, but also contain some information about hidden states. Based on the theoretical results presented in [34], we proposed an experimental guideline for the design of an experimental data-acquisition protocol which allows the collection of data containing sufficient information for the network structure reconstruction problem to become solvable. In particular, we have shown that if nothing is known about the network, then the data-collection experiments must be performed as follows:

1. for a network composed of p measured species, the same number of experiments p must be performed;

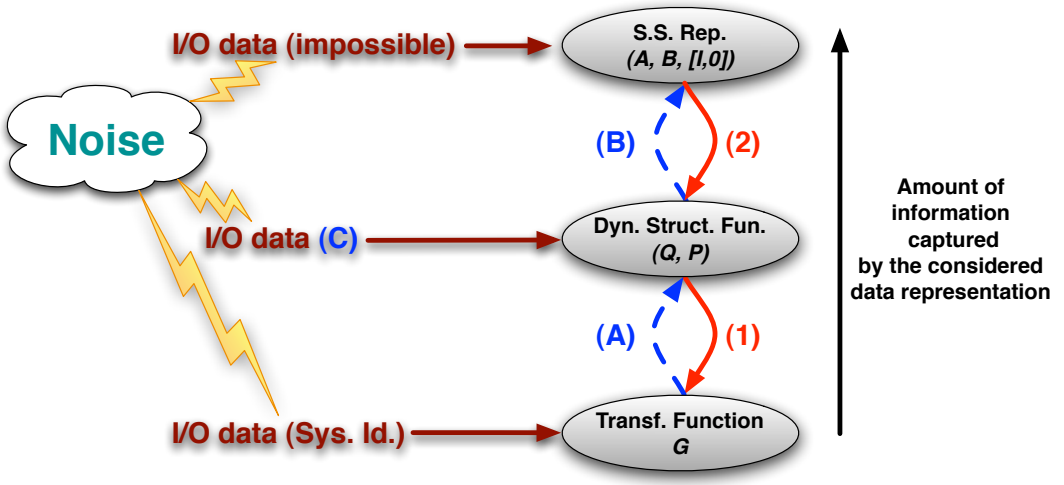


FIGURE 3.1: Mathematical structure of the network reconstruction problem using dynamical structure functions. Red arrows mean "uniquely determine", blue arrows indicate our work.

2. each experiment must independently control a measured species, i.e. control input i must first affect measured species i.

If the experiments are not performed in this way the network cannot be reconstructed, and any network structure fits the data equally well (e.g. a fully decoupled network or a fully connected network). If biologists have already some information about the network, as is usually the case, then these conditions can be relaxed as explained in [34].

Using dynamical structure functions as a mean to solve the network reconstruction problem, the following aspects need to be considered (see Figure 3.1):

Firstly (see (A) in Figure 3.1), the properties of a dynamical structure function and its relationship to the transfer function associated with the same system need to be precisely established [34]. We shall show that if experiments are performed as explained above: 1) we can not only obtain the network between the measured states but also the "self-loop" gain for each measured state; 2) we can still recover the true network structure even if the exact value of control inputs is unknown.

Secondly (see (B) in Figure 3.1), an algorithm for constructing a minimal order state-space representation consistent with an obtained dynamical structure function needs to be developed [101]. Using this last set of results, an estimate of the minimal number of hidden nodes that needs to be considered in the state space realisation can be obtained. In the context of biology, this helps understand the minimal number of unmeasured molecules in a particular pathway: a low number means that most molecules

in that pathway have been identified and measured, showing a good understanding of the system; while a large number shows that there are still many unmeasured variables, suggesting that new experiments should be carried out to better characterise that pathway.

Thirdly (see (C) in Figure 3.1), an efficient algorithm for reconstructing the dynamical structure function best fitting noisy input-output data needs to be developed. In particular, we focus in this Chapter on developing an efficient method of reconstructing networks in the presence of noise and nonlinearities. We assume that the conditions for network reconstruction presented above in (1) and (2) have been met. In our approach, we use the same information as traditional system identification methods, i.e. input-output data. However, with our method, steady-state (resp. time-series data) can be used to reconstruct the Boolean (resp. dynamical network) structure of the system.

The structure of this chapter is as follows. In Section 3.1.1, we give a simple example showing that direct system identification from input-output data does not allow the reconstruction of the network without full measurements. In Section 3.2, dynamical structure functions are defined and fundamental results concerning their usefulness in the network reconstruction problem are stated. Based on this new presentation of LTI systems, Section 3.3 introduces a number of open questions that could lead to new theories in the control society. Section 3.4 presents the main results on robust network reconstruction from input-output data in the presence of noise and nonlinearities. Further, we illustrate our algorithm on biologically-inspired network reconstruction examples in Section 3.5. We propose a minimal realisation algorithm based on state-space realisations and pole-zero analysis in Section 3.6 and model reduction algorithms in Section 3.7. Finally discussions on future work are presented in Section 3.8.

3.1.1 Motivating example

Consider a linear time-invariant system from which partial, non-noisy input-output measurements are obtained. Using system identification, a transfer function describing the input-output behaviour of this system can be obtained. However, in the partial observation case, network reconstruction is not possible without further information. To illustrate this, assume that the obtained transfer function from input-output data is given by:

$$G = \frac{1}{s+3}\begin{bmatrix} \frac{1}{s+1} \\ \frac{1}{s+2} \end{bmatrix}$$

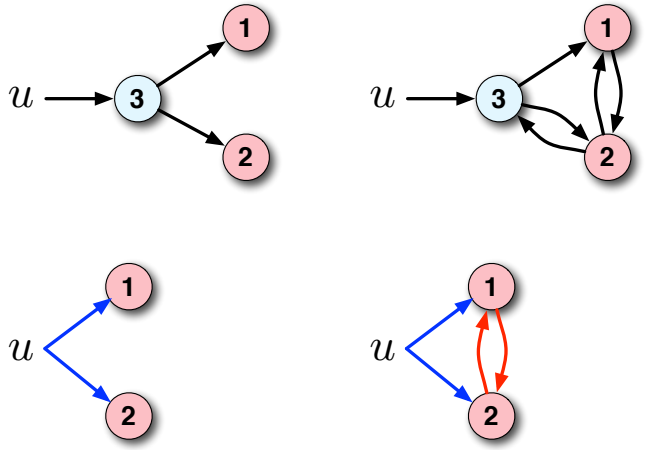


FIGURE 3.2: The same transfer function yields two minimal realisations with very different network structures: decoupled internal structure (left) and coupled internal structure (right). Measured species are shown by red circles while hidden species correspond to blue circles. The complete biochemical network, reflected in each state space realisation, is shown on top; on the bottom are the networks between measured species only. Blue and red arrows represent transfer functions and include the dynamics corresponding to hidden states.

It can be shown that this transfer function is consistent with two state-space realisations $\dot{x} = Ax + Bu$, $y = Cx$ with very different internal structures, i.e.

$$A_1 = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & -3 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -2 & -1 & 1 \\ -1 & -3 & 1 \\ 0 & -1 & -1 \end{bmatrix},$$

$B_1 = B_2 = [0 \quad 0 \quad 1]^T$, and $C_1 = C_2 = [I \quad 0] \in \mathbb{R}^{2\times 3}$ (i.e. the third state is hidden/non-observable). The networks in Figure 3.2 correspond to each of the indicated realisations of G. Note that both realisations are minimal. This demonstrates that even in the idealised setting (LTI dynamics, non noisy data), network reconstruction is not possible without additional information about the system. This also demonstrates exactly what additional information is necessary for the transfer function uniquely to determine the structure.

3.2 Dynamical structure functions and network reconstruction

Consider a nonlinear system $\dot{\bar{x}} = f(\bar{x},u,w)$, $\bar{y} = h(\bar{x},w)$ with p measured states $\bar{y}$, hidden states $\bar{z}$ (potentially a large number of them), m inputs u, and noise w. The system is linearised around an equilibrium point (a point such that $f(\bar{x}^*,0,0) = 0$), and it is assumed that inputs and noise do not move the states too far from the equilibrium point so that the linearised system is a valid approximation of the original nonlinear system. The linearised system can be written as $\dot{x} = Ax + Bu$, $y = Cx$, where $x = \bar{x} - \bar{x}^*$ and $y = h(\bar{x},0) - h(\bar{x}^*,0)$. The transfer function associated with this linearised system is given by $G(s) = C(sI - A)^{-1}B$. Typically, we can use standard system identification tools [51] to identify a transfer function $G(s)$ from input-output data.

Like system realisation, network reconstruction begins with the identification of a transfer function, but it additionally attempts to determine the network structure between measured states without imposing any additional structure on the hidden states. As we have shown in [34], this requires a new representation of linear time-invariant systems. This new representation is obtained as follows: First we transform $[A,B,C]$ to $\left[A^o,B^o,\left[I_p \quad 0\right]\right]$ (it is easy to show that this can always be done) and then partition the linear system dynamics as

$$\begin{aligned} \begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix} &= \begin{bmatrix} A^o_{11} & A^o_{12} \\ A^o_{21} & A^o_{22} \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} + \begin{bmatrix} B^o_1 \\ B^o_2 \end{bmatrix} u \\ y &= \begin{bmatrix} I_p & 0 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \end{aligned} \tag{3.1}$$

where $x = (y,z) \in \mathbb{R}^{n^o}$ is the full state vector, $y \in \mathbb{R}^p$ is a partial measurement of the state, z are the $n^o - p$ "hidden" states, and $u \in \mathbb{R}^m$ is the control input. In this work we restrict our attention to situations where output measurements constitute partial state information, i.e. $p < n^o$. We consider only systems with full rank transfer functions that do not have entire rows or columns of zeros, since such "disconnected" systems are somewhat pathological and only serve to complicate the exposition without fundamentally altering our conclusions.

Taking Laplace transforms of the signals in (3.1) yields

$$\begin{bmatrix} sY \\ sZ \end{bmatrix} = \begin{bmatrix} A^o_{11} & A^o_{12} \\ A^o_{21} & A^o_{22} \end{bmatrix} \begin{bmatrix} Y \\ Z \end{bmatrix} + \begin{bmatrix} B^o_1 \\ B^o_2 \end{bmatrix} U \tag{3.2}$$

where Y, Z, and U are the Laplace transforms of y, z, and u, respectively. Solving for Z gives

$$Z = \left(sI - A^o_{22}\right)^{-1} A^o_{21} Y + \left(sI - A^o_{22}\right)^{-1} B^o_2 U$$

Substituting this last expression of Z into (3.2) then yields

$$sY = W^o Y + V^o U \tag{3.3}$$

where $W^o = A^o_{11} + A^o_{12}\left(sI - A^o_{22}\right)^{-1} A^o_{21}$ and $V^o = A^o_{12}\left(sI - A^o_{22}\right)^{-1} B^o_2 + B^o_1$. Let D^o be a diagonal matrix with the diagonal term of W^o on its diagonal, i.e. $D^o = \text{diag}\{W^o\} = \text{diag}(W^o_{11}, W^o_{22}, ..., W^o_{pp})$. We thus obtain:

$$\left(sI - D^o\right) Y = \left(W^o - D^o\right) Y + V^o U$$

Note that $W^o - D^o$ is a matrix with zeros on its diagonal. We then have

$$Y = QY + PU \tag{3.4}$$

where

$$Q = \left(sI - D^o\right)^{-1} \left(W^o - D^o\right) \tag{3.5}$$

and

$$P = \left(sI - D^o\right)^{-1} V^o \tag{3.6}$$

Note that Q is zero on the diagonal.

Definition 3.2.1. *Given the system (3.1), we define the dynamical structure function of the system to be (Q,P), where Q and P are the internal structure and control structure, respectively, as defined in (3.5) and (3.6).*

Consider the system matrix A^o in eq. (3.1) as the original information flow graph with interconnection structure represented by a weighted directed graph (*digraph* for short). In this representation of the network, the n nodes of the digraph represent the n states of the system and a weighted edge a_{ij} from node j to node i indicates the existence of a causal dependence from from state j to state i in the network. The digraph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of nodes, $\mathcal{V} \triangleq \mathcal{V}_m \bigcup \mathcal{V}_h$, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges. $\mathcal{V}_m = \{v_1, \ldots, v_p\}$ is the set of measured nodes and $\mathcal{V}_h = \{v_{p+1}, \ldots, v_n\}$ is the set of hidden nodes. We shall give the graph theoretical interpretation of the intuition of dynamical structure function. As illustrated in Fig. 3.3, we condense the graph by deleting all the hidden states.

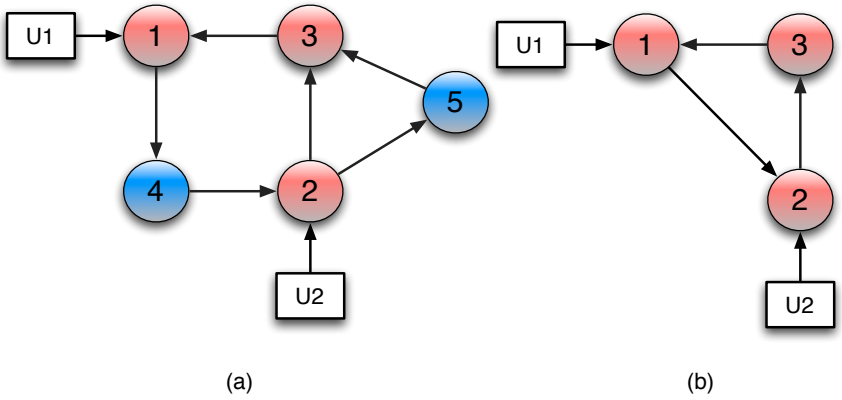


FIGURE 3.3: **(a)** An example system with two inputs, three measured states and two hidden states. **(b)** The corresponding condensed graph with measured states.

Let $\Phi = (I - A^o/s)^{-1}$, if we build the signal-flow network for A^o/s , then from Mason's rule we can obtain the gain from node i to node j directly from the graph as follows:

$$\Phi[i,j] = \frac{1}{\Delta} \sum_{\text{path } p \in \mathcal{G}} T_p \Delta_p, \tag{3.7}$$

where Δ is the determinant of the graph, which can be computed by

$$\Delta = 1 - \sum L_i + \sum L_i L_j + \ldots + (-1)^m \sum \ldots$$

T_p is the gain of the p^{th} forward path from node i to node j, L_i is the loop gain of each closed loop in the graph, and $L_i L_j$ is the product of the loop gains of any two non-touching loops (i.e. loops with no common nodes). Δ_p is the cofactor value of Δ for the p^{th} forward path, with the loops touching the p^{th} forward path removed (i.e. the remaining graph when you have removed those parts of the graph that form loops while retaining the parts on the forward path).

From the matrix inversion formula, it is easy to check that $I - \frac{W^o(s)}{s}$ lies in the upper left part of Φ representing the information flow between the measured states.

Example 3.2.1. *Consider a system with the structure depicted in Fig. 3.3. A linear*

system's representation is

$$\begin{aligned}\dot{x} &= \begin{bmatrix} a_{11} & 0 & a_{13} & 0 & 0 \\ 0 & a_{22} & 0 & a_{24} & 0 \\ 0 & a_{32} & a_{33} & 0 & a_{35} \\ a_{41} & 0 & 0 & a_{44} & 0 \\ 0 & a_{52} & 0 & 0 & a_{55} \end{bmatrix} x + \begin{bmatrix} b_{11} & 0 \\ 0 & b_{22} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} u \\ y &= \begin{bmatrix} I_3 & 0 \end{bmatrix} x \end{aligned}$$

where I_3 is the 3×3 identity matrix. Following the above definitions of $[W^o,\ V^o]$ we have

$$W^o = \begin{pmatrix} a_{11} & 0 & a_{13} \\ \frac{a_{24}a_{41}}{s-a_{44}} & a_{22} & 0 \\ 0 & a_{32} + \frac{a_{35}a_{52}}{s-a_{55}} & a_{33} \end{pmatrix}, \text{ and } V^o = \begin{pmatrix} b_{11} & 0 \\ 0 & b_{22} \\ 0 & 0 \end{pmatrix}.$$

$$Q = \begin{pmatrix} 0 & 0 & \frac{a_{13}}{s-a_{11}} \\ \frac{a_{24}a_{41}}{(s-a_{22})(s-a_{44})} & 0 & 0 \\ 0 & \frac{a_{35}a_{52}+a_{32}(s-a_{55})}{(s-a_{33})(s-a_{55})} & 0 \end{pmatrix} \tag{3.8}$$

$$P = \begin{pmatrix} \frac{b_{11}}{s-a_{11}} & 0 \\ 0 & \frac{b_{22}}{s-a_{22}} \\ 0 & 0 \end{pmatrix} \tag{3.9}$$

Note that the links between measured states and the complexity of hidden states are encoded in the expression of W^o and V^o.

$$\begin{aligned}[W^o \ \ V^o] &= [{A^o}_{11} \ \ {B^o}_1] + {A^o}_{12}(sI - {A^o}_{22})^{-1}[{A^o}_{21} \ \ {B^o}_2] \\ &= \begin{pmatrix} a_{11} & 0 & a_{13} & b_{11} & 0 \\ 0 & a_{22} & 0 & 0 & b_{22} \\ 0 & a_{32} & a_{33} & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{a_{24}a_{41}}{s-a_{44}} & 0 & 0 & 0 & 0 \\ 0 & \frac{a_{35}a_{52}}{s-a_{55}} & 0 & 0 & 0 \end{pmatrix}. \end{aligned}$$

This structure may be particularly useful in describing the relationships between measured variables, for example, in biochemical or social systems. In these more "fluid" systems, not only might it be unreasonable to assume knowledge of an explicit non-trivial partition on all system states into distinct subsystems, but such a partition might not even exist. In this setting, it is natural to describe the structure of the complex system graphically by associating each measured output and input with a node of the graph.

Edges are then identified with each non-zero entry in $[W^o,\ V^o]$.

Definition 3.2.2. *Consider a system characterised by a transfer function G. The dynamical structure of the system can be reconstructed, if there is only one admissible dynamical structure function, (Q,P), that is consistent with G. A realisation of the dynamical structure function is defined as reconstruction. Likewise, the Boolean structure of the system can be reconstructed if all admissible dynamical structure functions that are consistent with G have the same Boolean structure.*

Definition 3.2.3. *We say that a realisation is G minimal if this realisation corresponds to a minimal realisation of G [107]. We say that a realisation is (Q,P) minimal if this realisation has the smallest order and is consistent with (Q,P).*

The underlying principle to find a (Q,P) minimal realisation is to search for a realisation with the minimal number of hidden states.

It can be shown that $G = (I-Q)^{-1}P$. Thus the dynamical structure function of a system contains more information than the transfer function, and less information than the state-space representation. In [34] they conclude that, with no other information about the system, dynamical or Boolean reconstruction is not possible. Moreover, for *any* internal structure Q there is a dynamical structure function (Q,P) that is consistent with G. In particular, this shows that the use of criteria such as sparsity or decoupledness to guide our selection of a proposal network structure can be misleading. If one were to optimise for decoupledness, for example, a dynamical structure $(0,G)$ could and would always be found, regardless of the true underlying structure. Thus, if we are to use these kinds of criteria, they must be firmly justified a priori.

Proposition 3.2.1. *[34] Given a $p \times m$ transfer function G, dynamical structure reconstruction is possible from partial structure information if and only if $p-1$ elements in each column of $\begin{bmatrix} Q & P \end{bmatrix}^T$ are known that uniquely specify the component of (Q,P) in the nullspace of $\begin{bmatrix} G^T & I \end{bmatrix}$.*

The importance of this result is that it identifies exactly what information about a system's structure, beyond knowledge of its transfer function, must be obtained to be able to recover the structure without appeal to a priori assumptions, such as sparsity, or parsimony, etc. This enables the design of experiments targeting precisely the extra information needed for reconstruction. In particular when $p = m$ and G is full rank, we observe that imposing that P is diagonal, i.e. that each input controls a measured state independently, is sufficient for reconstruction.

Corollary 3.2.1. *[34] If $m = p$, G is full rank, and there is no information about the internal structure of the system, Q, then the dynamical structure can be reconstructed if each input controls a measured state independently, i.e. without loss of generality, the inputs can be numbered such that P is diagonal. Moreover, $H = G^{-1}$ characterises the dynamical structure as follows*

$$Q_{ij} = -\frac{H_{ij}}{H_{ii}} \text{ and } P_{ii} = \frac{1}{H_{ii}}. \tag{3.10}$$

For a given p, there are 2^{p^2-p} possible Boolean networks (remember that Q has zeros on the diagonal), which can be ordered $k = 1, ..., 2^{p^2-p}$.

Definition 3.2.4. *A Boolean structure $\mathscr{B}_k$ corresponding to a Boolean network k is defined as follows: if $Q(s) \in \mathscr{B}_k$ then $Q_{ij}(s) = 0$ for all (i, j) for which the Boolean network has zero entries and when $i = j$, and all other $Q_{ij}(s)$ are free variables.*

3.2.1 Extension of Corollary 3.2.1

Corollary 3.2.1 requires the following to exactly reconstruct p-node network:

a. p different inputs/experiments;

b. each input controls a measured state independently.

The second requirement is obviously not scalable and hard for biologists to implement especially when p is large. Can one reduce the number of required experiments at the expense of reconstruction accuracy? Or more specifically, can one propose a method for analysing the tradeoff between the number of correctly obtained links and number of required experiments. We remain the second requirement. We shall pursue this idea in more details in this section. We adopt the definitions of polynomial matrix, normal rank and etc. (Chapter 3.11 in [107]).

We start from the following equality

$$G^T(I - Q)^T = P^T, \tag{3.11}$$

where G^T, P^T are $m \times p$ transfer matrices and m is the number of experiments. We can decompose eq. (3.11) to the following p equations since solving eq. (3.11) is equivalent to solving the following equations (for $i = 1 : m$)

$$G^T(e_i^T - Q[i,:])^T = P[i,:]^T, \tag{3.12}$$

where $P[i,:]^T$ is the i^{th} row of P, e_i^T is the i^{th} row of identify matrix I and $Q[i,:]$ is the i^{th} row of Q.

Let $A_i \triangleq G[1:i-1\ i+1:p, 1:i-1\ i+1:m]^T$ $q_i = Q[i, 1:i-1\ i+1:p]^T$ and $g_i = G[i, 1:i-1\ i+1:m]^T$ then we have

$$A_i q_i = g_i. \tag{3.13}$$

where A_i is a known $(m-1)\times(p-1)$ transfer matrix, g_i is a known $(m-1)\times 1$ vector with entries transfer functions and q_i is a $(p-1)\times 1$ transfer matrix.

To solve eq. (3.13) for a given i, when $m = p$ and $A_i = N_i(s)/d_i(s)$ where N_i is a polynomial matrix and has full normal rank [107], then we can directly solve above eq. (3.13). However, when $m < p$, there are infinite number of solutions. We are trying to find a sparsest solution q_i and therefore formulate the following problem.

Problem 3.2.1.

$$q_i^* = argmin_{q_i \in \mathscr{S}^{p-1}} \|q_i\|_0$$
$$s.t.: A_i q_i = g_i.$$

$\|\cdot\|_0$ *is the* $\mathscr{L}_0$ *norm counting the nonzero entries.*

The cost function is optimised over a set $\mathscr{S}^{p-1}$ containing all $(p-1)\times 1$ vector with its elements strictly proper transfer functions.

There are several issues regarding the above problem formulation, for example, when does this problem have an unique solution. In this case, how does this unique solution relate to the true solution. Note that answers to above questions are related to the number of experiments m, we now have the following proposition.

Proposition 3.2.2. *For a known $m \times p$ $(m \leq p)$ transfer matrix $A \triangleq N/d$ (where N is a polynomial matrix) has full normal row rank [107] and a known $m \times 1$ transfer matrix b, when $m \geq 2S$, then any transfer matrix x with at most S non-zero transfer functions, i.e., $\|x\|_0 \leq S$ can be reconstructed uniquely from*

$$Ax = b. \tag{3.14}$$

Proof. This can be shown by contradiction.

If there exists at least two different solutions x_1 and x_2 satisfying eq. (3.14), then from $Ax_1 = b$ and $Ax_2 = b$ we can obtain $A(x_1 - x_2) = 0$. Let $z \triangleq x_1 - x_2 \neq 0$, then $\|z\|_0 \leq \|x_1\|_0 + \|x_2\|_0 = 2S$. However, from $Az = 0$ and let $z = n_z/d_z$ where n_z is a polynomial

matrix, we have $Nn_z = 0$. Since N has full normal row rank $m \geq 2S$, contradiction to the definition of normal rank! □

Remark 3.2.1. *Proposition 3.2.2 states that if the true solution is sparse then we can indeed reduce the number of required experiments. This Proposition links the number of required experiments to find the i^{th} row of Q to the number of in-degree of node i rather than the total number of measured node, i.e. p. In other words, the number of required experiments to find Q is therefore linked to the maximal number of in-degree of any node in the network.*

However, the sparsity of the true solution is unknown, how can one determine whether a solution from Problem 3.2.1 is the true solution? We now propose the following iterative algorithm to solve this. The idea is that when we increase the number of experiments, the obtained result from optimisation preserves then we claim that the obtained solution is the true solution.

Let G_i be the transfer function obtained from the i^{th} experiment, i.e. perturb the i^{th} node.

Algorithm 5 Network reconstruction for node i

$G^c = []$
for $j = 1 : p$ **do**
  $G^c = [G^c; G_j]$
  Solve q_i^j from Problem 3.2.1
  if $q_i^j = q_i^{j-1}$ **then**
    Stop and return q_i^j
  end if
end for

Remark 3.2.2. *Most of social and biological networks are scale-free networks [2], only few nodes have high degrees of connections and most of nodes only have very few connections. By using Proposition 3.2.2, we can perform a few experiments and obtain most of connections in such networks.*

Does it matter which experiments are performed? Or more specifically, how does the performance of algorithm change with respect to chosen perturbed nodes. A quick answer is no when the underlying graph is strongly connected.

Iterative algorithm also proposes a question for future research, i.e. the computational complexity issue. It is still open to give an effective algorithm to solve above Problem 3.2.1. One may try $\mathscr{L}_1$ approximation or l_1 approximation in discrete-time case

[20, 56], but it is still under investigation how the solution to this approximation to the true one.

3.3 Impact on systems theory

The introduction of dynamical structure functions introduces new problems in systems theory. In [93], it proposes a number of interesting and fundamental problems which explores the relationships between different representations of the same system. For example, classical realisation theory considers the situation where a system is specified by a given transfer function, and it explores how to construct a consistent state space description. Many important ideas emerge from the analysis:

1. State-space realisations are generally more informative than a transfer function representation of a system, as there are typically many state-space realisations consistent with the same transfer function.

2. The order of the state realisation is a sensible measure of complexity of the state representation, and there is a well-defined *minimal* order of any realisation consistent with a given transfer function; this minimal order is equal to the Smith-McMillian degree of the transfer function.

3. Ideas of controllability and observability of a state realisation characterise important properties of the realisation, and any minimal realisation is both controllable and observable.

In a similar way, introducing dynamical structure function influences a variety of concepts in systems theory, including realisation, minimality, and model reduction.

3.3.1 Realisation

The definition of dynamical structure functions enriches the kinds of realisation questions one may consider. Two classes of representation question emerge: reconstruction problems and structure realisation problems (Figure 3.4).

Reconstruction problems consider the construction of a dynamical structure function of a system given its transfer function. Because such structure representations are generally more informative than a transfer function, these problems are ill-posed. In particular, we may consider the following reconstruction problem:

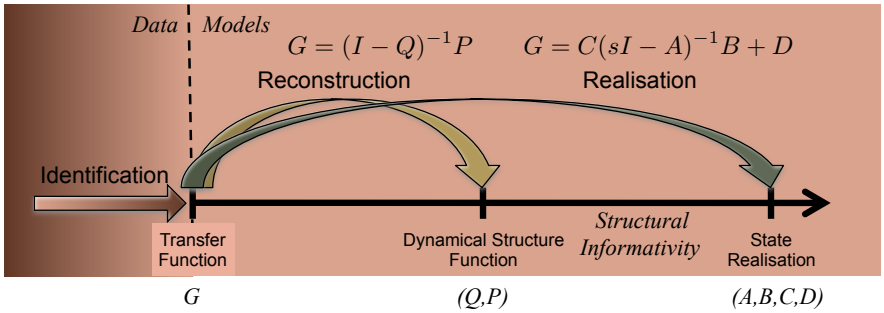


FIGURE 3.4: Dynamical structure functions introduce new classes of realisation problems: reconstruction and structure realisation. These problems are distinct from identification. [93]

Given a transfer function $G(s)$, find a dynamical structure function (Q,P) such that $G = (I-Q)^{-1}P$. Dynamical structure structure reconstruction is also called *network reconstruction*, particularly in systems biology where it plays a central role. There, the objective is to measure fluctuations of various proteins, or other chemical species, in response to particular perturbations of a biochemical system, and then infer causal dependencies among these species.

Structure realisation problems then consider the construction of a state space model, possibly generalised to include auxiliary variables as necessary, consistent with a given dynamical structure function of a system. Like the classical realisation problem or reconstruction problems, these problems are also ill-posed since there are typically many state realisations of a given partial structure representation of a system.

Dynamical structure function realisation: Given a system G associated dynamical structure function (Q,P), find a state space model (A,B,C,D) consistent with (Q,P),

Dynamical structure function realisation may sometimes be called *network realisation*, consistent with the nomenclature for dynamical structure structure reconstruction.

Note that all the reconstruction and structure realisation problems here are different from identification problems, just as classical realisation differs from identification. For the systems considered here identification refers to the use of input-output data (and no other information about a system) by which to choose a representation that best describes the data in some sense. Because input-output data only characterises the input-output map of a system, identification can at best characterise the system's transfer function; no information about structure, beyond the sparsity structure, is available in such data. In spite of this distinction, however, it is not uncommon for reconstruction problems to be called *structure identification* problems.

3.3.2 Minimality

Just as partial structure representations enrich the classical realisation problem, they also influence the way we think about minimality.

Minimal dynamical structure realisation: In this situation one needs to consider how to measure the complexity of a system's state realisation, from which signal structure is derived. The obvious choice would be to use the order of the realisation as a natural measure of complexity, and the problem would then be to find the minimal order state realisation [101, 102] consistent with its associated dynamical structure function, (Q,P). Note that this minimal order is guaranteed to be finite (for the systems considered here) and can easily be shown to be greater than or equal to the Smith-McMillian degree of the transfer function specified by the signal structure; we call this number the *structural degree* of the dynamical structure functions [92].

These various problems demand new ideas for thinking about the complexity of a system's representation, especially that of a dynamical structure function. These new ideas about complexity, in turn, introduce opportunities for characterising minimality of a representation that add insight to our understanding of the relationship between a system's behaviour and its structure, much like controllability and observability characterise classical notions of minimality in a system's state realisation. Besides suggesting the need for a characterisation of minimality, however, these ideas also impact notions of approximation and how we think about model reduction.

3.3.3 Model Reduction

Each of the reconstruction and structure realisation problems described above has associated with it not only a minimal but also an approximate-representation version of the problem. The minimal-representation versions of these problems, as described above, seek to construct a representation of minimal complexity in the targeted class that is nevertheless consistent with the system description provided. Similarly, approximate-representation versions of these problems seek to construct a representation in the targeted class that has a lower complexity than the minimal complexity necessary to deliver consistency with the system description provided. As a result, consistency with the given system description can not be achieved, so measures of approximation become necessary to sensibly discuss a “best" representation of the specified complexity.

For example, associated with the classical realisation problem is the standard model reduction problem. In this situation, a transfer function is specified, and one would like to construct a state realisation with a complexity that is lower than that which

is necessary (for such a realisation to be consistent with the given transfer function) that nevertheless "best" approximates it. Likewise, note that the appropriate notion of approximation depends on the type of system representation that is initially provided; here, a transfer function is provided, so an appropriate measure of approximation could be an induced norm, such as H_∞. Thus, one could measure the quality of an approximation by measuring the induced norm of the error between the given transfer function and that specified by the approximate state realisation. In any event, because the specified system description is a transfer function, the resulting measure of approximation is typically one that either directly or indirectly measures the difference in input-output dynamic behaviour between the approximate model and the given system description; the focus is on dynamics, not system structure, when considering notions of approximation in the standard model reduction problem.

Approximate dynamical structure functions realisation: In this situation one would like to find a state space realisation with a model complexity that is lower than the minimal complexity necessary to specify a given dynamical structure function.

The introduction of dynamical structure functions representations suggests a number of new problems in systems theory. These problems include new classes of realisation problem, called reconstruction and structural realisation problems, as well as a number of new reduction problems. The overview offered here is merely meant to give a perspective of the landscape of problems that emerges with the introduction of dynamical structure functions.

3.4 Robust network structure reconstruction

In this section we consider the problem of robustly reconstructing dynamical network structures. Data are obtained from input-output measurements of a noisy nonlinear system. From this type of data we aim to find the internal network structure Q associated with the linearised system (3.1).

For simplicity of exposition we assume that there is no *a priori* information on the internal network structure Q. The results still follow if some *a priori* information about Q is available, and such information can typically be used to relax the experimental protocol according to Proposition 3.2.1. Thus, data are collected according to the measurement protocol described in the introduction:
(1) the number of distinct data-collection experiments is the same as that of measured species. This in particular implies that $u(t), y(t) \in \mathbb{R}^p$;
(2) each input u_i controls first the measured state y_i so that P is a diagonal matrix

($p \times p$). To average out the noise, data-collection experiments are repeated N times.

In the following sections, we propose two approaches for estimating the dynamical structure function (Q,P) from measured input-output data. The first approach is indirect and involves estimating the transfer function G while the second approach relies on the solution of a direct optimisation problem. More precisely, in the first approach (see Figure 3.5 (a)), for each experiment i we first estimate $G_i(s)$ (i.e. the i^{th} column of $G(s)$) using standard system identification tools [51]. In a second step, the dynamical structure function $(Q(s),P(s))$ is computed from the estimated transfer function $G(s)$. Since information is lost in the process of estimating $G(s)$, we later consider the case where $(Q(s),P(s))$ is directly estimated from data (without estimating first $G(s)$, see Figure 3.5 (b)).

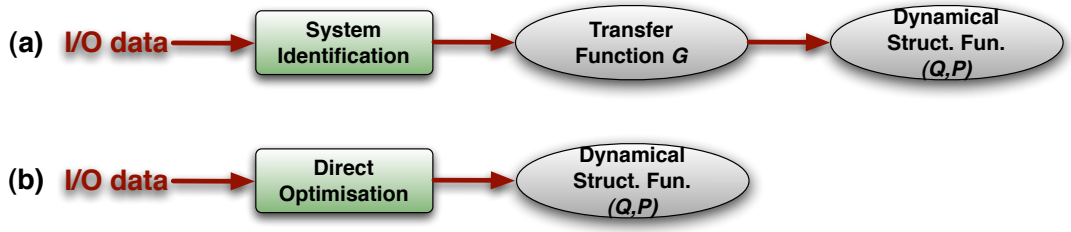


FIGURE 3.5: Two approaches to obtain dynamical structure functions.

Concerning the type of input-output data collected, we first assume the case of time-series input-output data. We then consider the special case where only steady-state data are available.

3.4.1 Dynamical network reconstruction from identified transfer functions

In this section, we describe the first method relying on system identification. This method allows us to obtain dynamical structure functions from a transfer function identified using measured time-series data (see Figure 3.5 (a)).

Consider a transfer matrix $G(s)$ estimated from noisy data. According to Corollary 3.2.1, if G is full rank there is a unique Q and diagonal P satisfying $(I-Q)G = P$. Since G is an approximation of the actual system, Q and P will typically be mere approximations of the actual Q and P. Moreover, due to noise and unmodelled dynamics, it is likely that Q does not even have the correct Boolean structure. Typically, the internal structure function Q obtained from such a procedure will be fully connected, i.e. all non-diagonal elements of Q will be non-zero.

The main idea in the solution of the network reconstruction problem from noisy data is the following. For p measured states, Q has p^2-p unknowns. We want to quantify the *distance* from G (or directly from the measured data) to all possible Boolean structures (and there are 2^{p^2-p} of them). Some of such distances will be large revealing that the corresponding Boolean structures are likely to be incorrect while others will be small, making them candidates for the correct structure.

To make the notion of distance rigorous, consider a Boolean mapping from a transfer matrix to a Boolean matrix in Definition 3.4.1 and a particular Boolean structure $\mathscr{B}_k$ in Definition 3.2.4.

Definition 3.4.1. *A Boolean mapping from a transfer matrix to a Boolean matrix $Q(s) \to \mathscr{B}(Q)$, where $\mathscr{B}(Q)$ is a Boolean matrix with the same dimension as transfer matrix Q and $\{\mathscr{B}(Q)\}_{i,j} = 0 \ \forall i,j$ if and only if $Q_{ij}(s) = 0$ for all s, otherwise, $\{\mathscr{B}(Q)\}_{i,j} = 1$.*

For a given p, there are 2^{p^2-p} possible Boolean networks B_k (remember that $Q(s)$ has zeros on the diagonal and therefore $\mathscr{B}(Q)$ will always have zeros on the diagonal) which can be ordered using the index $k = 1,...,2^{p^2-p}$.

Definition 3.4.2. *A Boolean structure $\mathscr{B}_k$ corresponding to a Boolean network B_k is defined as follows: $\{Q(s) : \mathscr{B}(Q) = B_k\}$.*

The distance from G to the Boolean structure $\mathscr{B}_k$ is defined as the smallest perturbation Δ to G (measured in some norm) so that the perturbed system G_Δ belongs to the set of transfer functions $\tilde{G}$ such that $Q \in \mathscr{B}_k$, where Q is obtained from $(I-Q)\tilde{G} = P$. Finding the distance from G to a Boolean structure $\mathscr{B}_k$, gives us a quantitative information about how much we would need to perturb G (or the data) to obtain a new system transfer function for which the associated Q corresponds to the considered Boolean structure, i.e. for which $Q \in \mathscr{B}_k$.

There are many possible approaches defining such "smallest perturbations", including several uncertainty models and norms to choose from. This choice is vital to obtain a convex minimisation problem. For example, additive, multiplicative or uncertainty in the coprime factors all lead to non-convex minimisation problems. In order to obtain a convex minimisation problem, we consider the output (could also be input) feedback uncertainty model. In this framework, the "true" system is given by $(I+\Delta)^{-1}G$, where Δ represents unmodelled dynamics, including nonlinearities, and noise.

Given this choice of dynamic uncertainty, the problem is defined as follows. Given a particular Boolean structure $\mathscr{B}_k$, the objective is to minimise $\|\Delta\|$, in some norm, such that Q obtained from $(I+\Delta)^{-1}G = (I-Q)^{-1}P$ has the desired Boolean structure, i.e. $Q \in \mathscr{B}_k$. All P_{ii} are also free (remember that, by assumption, P is diagonal).

We can rewrite the above equation as $\Delta = GP^{-1}(I-Q)-I$. So, we intend to minimise $\|GP^{-1}(I-Q)-I\|$ over $Q \in \mathscr{B}_k$ and P diagonal. Since P is diagonal, its inverse P^{-1} is also diagonal.

Define a new matrix $X = P^{-1}(I-Q)$ whose diagonal is the diagonal of P^{-1} and for which the off diagonal elements are given by $P_{ii}^{-1}Q_{ij}$. Since $Q \in \mathscr{B}_k$ this imposes structural constraints on X, i.e. some off-diagonal $X_{ij} = 0$. These zero X_{ij} correspond to those Q_{ij} which are equal to zero (since $X_{ij} = P_{ii}^{-1}Q_{ij}$ for $i \neq j$).

Definition 3.4.3. *For all k, define $\mathscr{X}_k \triangleq \{X(s) : \mathscr{B}(X) = B_k + I_p\}$, where I_p is identity matrix of dimension p.*

Remark 3.4.1. *Definition 3.4.3 implies the following facts:*

(i) when $i \neq j$, $X_{ij}(s) = 0$ for all the Boolean structures $\mathscr{B}_k$ in Definition 3.2.4 which are such that $B_k[i,j] = 0$; all other $X_{ij}(s)$ are free variables;

(ii) when $i = j$, $X_{ii}(s)$ is a free variable.

Using Defintion 3.4.3, the distance from G to a particular Boolean structure $\mathscr{B}_k$ can be written as

$$\alpha_k = \inf_{X \in \mathscr{X}_k} \|GX - I\|^2$$

which is a convex minimisation problem by choosing some norm.

Remark 3.4.2. *In this optimisation problem, $X(s) \in \mathscr{X}_k$ approximates the inverse of G as "close" as possible. If $\mathscr{X}_k$ corresponds to the fully connected Boolean network, then the solution to this optimisation is exactly $X = G^{-1}$.*

Next we show that this problem can be cast as a least squares optimisation problem. If we use the norm defined by $\|\Delta\|^2 =$ sum of all $\|\Delta_{ij}\|_2^2$, where $\|\cdot\|_2$ stands as the $\mathscr{L}_2$-norm over $s = j\omega$, then using the projection theorem [94] the problem reduces to

$$\begin{aligned} \alpha_k = \inf_{X \in \mathscr{X}_k} \|GX - I\|^2 &= \inf_{X \in \mathscr{X}_k} \sum_i \|GX_i - e_i\|_2^2 \\ &= \sum_i \inf_{Y_i} \|A_i Y_i - e_i\|_2^2 \\ &= \sum_i \|A_i (A_i^* A_i)^{-1} A_i^* e_i - e_i\|_2^2, \end{aligned}$$

where X_i is the i^{th} column of $X \in \mathscr{X}_k$, Y_i is a column vector composed by the free (i.e. nonzero) elements of X_i, A_i is obtained by deleting the j^{th} column of G when the corresponding element $X_i(j)$ is 0 for all j, and $(\cdot)^*$ denotes transpose conjugate. The infimum

is achieved by choosing $X_i = (A_i^* A_i)^{-1} A_i^* e_i$, and $A_i^* A_i$ is always invertible since G is full rank in Corollary 3.2.1. After obtaining all the α_k for all k, the optimal distance

$$\alpha = \min_k \alpha_k.$$

If experiments are repeated N times (as they should be) and we obtain a transfer function G_i for each experiment, then the above analysis still follows simply by forming a higher dimensional matrix $G = [G_1^T \ \dots \ G_N^T]^T$.

Remark 3.4.3. *The optimal distance α_k can be seen as a measure of robustness for the network structure in question given the input-output data.*

3.4.2 Dynamical network reconstruction directly from time-series data

The previous sections used a two-step approach in which system identification was first used to estimate a transfer function from measured input-output data and then, in a second step, the identified transfer function was used to obtain a dynamical structure function representation of the system which is optimal in terms of a particular metric. This section proposes a method which allows identification of the optimal dynamical structure function representation directly from the measured input output data (see Figure 3.5 (b)). The advantage of this direct network structure reconstruction from data is that no information is lost during the initial transfer function identification stage.

Due to the equivalence between dynamical uncertainty perturbations, we are free to choose, without loss of generality, the type of uncertainty perturbation that best suits our needs. For the direct method, instead of a feedback uncertainty as was considered in the previous section, the uncertainty perturbation we are considering here is the additive dynamic uncertainty on the output, i.e. $Y = G_\Delta(U + \Delta)$. In this case we think of the "distance" in terms of how much we need to change the input (data) to fit a particular Boolean structure. Since $G_\Delta = (I - Q)^{-1} P = X^{-1}$, the equality $Y = G_\Delta(U + \Delta)$ can be written as

$$\Delta = XY - U,$$

where $X \in \mathscr{X}_k$, for some particular Boolean network k. Recall that structural constraints in Q can be imposed directly on X from the equality $X = P^{-1}(I - Q)$. The difficulty of identifying a non-causal transfer matrix X with structural constraints arises, therefore standard system identification tools can not be applied and new tools need be introduced.

3.4.3 Penalising connections

The above method suffers from a crucial weakness: there are several Boolean structures with distances smaller than or equal to the distance to the "true" network. Indeed, the extra degrees of freedom of the fully-connected network allow the corresponding distance α_k to be the smallest of all. This is similar to the noisy data over-fitting problem encountered in system identification where the higher the order of the transfer function, the better the fit. Obviously, if we only focus on noisy data best fit, eventually we end up fitting noise and so a large system order is not typically a good choice. Accordingly a compromise has to be struck.

If the true network has l non-existent connections (l off-diagonal elements in Q are zero) and the data are non-noisy, then there are $2^l - 1$ different networks that have a smaller or equal distance (due to the additional degrees of freedom provided by the extra connections). When noise is present, the "true" network will typically have an optimal distance similar to these other l networks. The question of how to find the "true" network thus arises. With repeated experiments, small enough noise (i.e. large enough signal-to-noise ratio) and negligible nonlinearities, the optimal distances of those l networks become comparable, and they are typically much smaller than those of the other networks. To try to reveal the "true" network, one can strike a compromise between network complexity (in terms of number of connections) and data fitness by penalising extra connections. The are several methods by which to strike this compromise. Here we introduce methods known as Akaike's information criterion (AIC) [37], or some of its variants such as AICc (which is AIC with a second order correction for small sample sizes), and the Bayesian information criterion (BIC).

The AIC-type approach is a test between models - a tool for model selection. Given a data set, several competing models may be ranked according to their AIC value, that having the lowest AIC being the best. From the AIC value one may typically infer that the best models are in a tie and the rest are far worse, but it would be arbitrary to assign a value above which a given model is rejected [12]. The AIC value in our case for a particular Boolean network B_k is defined as:

$$AIC_k = 2L_k - 2\ln\alpha_k, \tag{3.15}$$

where L_k is number of (non-zero) connections in the Boolean network B_k and α_k is the optimal distance based on this parameter constraint.

Although finding the optimal distance in the second term of eq. (3.15) can be done efficiently, the number of Boolean networks 2^{p^2-p} grows very fast with the number of

measured states p. To find the network with the smallest distance it is thus not desirable to compute the optimal distance for each possible Boolean network. Fortunately, there are ways to reduce the number of networks that need to be considered. As we saw in the previous section $\inf_{X\in\mathscr{X}_k}\|GX-I\|^2=\sum_i \inf_{Y_i}\|A_iY_i-e_i\|_2^2$ meaning that we can solve each optimisation problem separately. Since each Y_i corresponds to $p-1$ unknowns in the i^{th} row of Q, this reduces the problem to solving $p2^{p-1}$ optimal distances. Finding a polynomial-time algorithm to compute the optimal distance through this method is a subject of current investigation.

Reconstruction with the zero norm

Another way of taking the number of connections into account is to formulate the optimisation problem as follows:

$$\inf_{X\in\mathscr{X}_k}\left(\alpha_k+\beta\|X\|_0\right), \tag{3.16}$$

where β is a parameter balancing data-fitting and model complexity (i.e. the number of non-zero connections). In (3.16), $\|X\|_0$ denotes the number of nonzero elements in the matrix X, and it is known as the zero norm. Note that this minimisation problem (e.g., direct optimisation) can be equivalently written as:

$$\inf_{X\in\mathscr{X}_k}\left(\|XY-U\|^2+\beta\|X\|_0\right)=\sum_i \inf_{X_i}\left(\|X_i^TY-U_i^T\|_2^2+\beta\|X_i^T\|_0\right), \tag{3.17}$$

where X_i^T is the i^{th} row of $X\in\mathscr{X}_k$ and U_i^T is the i^{th} row of U. Directly solving this problem is in general NP-hard. A frequently discussed approximation is to relax this problem in the same way as [44]. Moreover, since there are p independent optimisations in eq. (3.17), we can choose different β_i for each i. Alas, there is no clear rule for selecting β_i to optimally balance the two terms in eq. (3.17) [44]. The choice of β_i is currently under investigation.

3.4.4 Boolean network reconstruction from steady-state data

So far we have assumed that time-series data are available. Frequently, however, experimentation costs and limited resources only permit steady-state measurements. In addition, with steady-state measurements it is typically possible to perform a larger number of experiments for the same time, effort and cost. As shown below, most of the connectivity of the network together with the associated steady-state gains (and the associated positive or negative sign) can still be reconstructed from steady-state data.

However, no dynamical information will be obtainable. In other words in most cases we can still recover the Boolean network from steady-state data.

Assume that after some time of maintaining the control input concentrations at a constant value, the measured outputs y have converged to a steady-state value. This is equivalent (if the system is stable or quasi-stable [79]) to assuming that we can obtain $G_0 = G(0)$, i.e. $G(s)$ evaluated at $s = 0$. If $Q_0 = Q(0)$ and $P_0 = P(0)$, then $(I - Q(s))G(s) = P(s)$ evaluated at $s = 0$ becomes $(I - Q_0)G_0 = P_0$. From this equation all of the results given in Section 3.4.1 and 3.4.2 follow provided that no element of $G(s)$ has a system zero [107] at 0. In that case a nonzero element in the obtained Boolean network indicates the existence of a causal relationship between the corresponding pair of nodes while a zero element indicates the absence of such relationship.

In the literature, there are indeed a number of methods [79, 26] using steady-state data to reconstruct Boolean network. Sontag [79] takes the inverse of the constructed square matrix G (similar to G_0 above which is formed by outputs from different perturbations) and that claim that the nonzero entries in $X = G^{-1}$ correspond to the edges in the network and the zero entries of $X[i, j]$ correspond to non-relation from node j to i in the network.

However, for most cases, $X = G^{-1}$ has no element as 0 due to noise effect. Then truncation step is needed to find the true X. Can one claim that the small element in X, i.e., element close to 0 corresponds to a non-relation from j to i as claimed in the paper? What is the proper threshold for truncation?

This claim is not sound for certain cases and what we shall illustrate that the small change in $X[i, j]$ to 0 might lead to large change in G.

Lemma 3.4.1. *For invertible matrices A, B, we have the following equality*

$$A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}.$$

We now consider the difference of the data matrix when we delete a element of X which is close to 0 but not equal to 0. If we directly write down the difference

$$\begin{aligned}(X - X[i,j]e_i e_j^T)^{-1} - X^{-1} &= (X - X[i,j]e_i e_j^T)^{-1} X[i,j] e_i e_j^T X^{-1} \\ &= G e_i (X[i,j]^{-1} - e_j^T G e_i)^{-1} e_j^T G \\ &= G[:,i](X[i,j]^{-1} - G[j,i])^{-1} G[j,:] \\ &= (X[i,j]^{-1} - G[j,i])^{-1} G[:,i] G[j,:].\end{aligned}$$

The second equality is from Woodbury matrix identity. If we take the matrix norm (for

example, 2-norm) of the above error

$$\begin{aligned}\delta_{ij} \triangleq \|(X - X[i,j]e_i e_j^T)^{-1} - X^{-1}\| &= \|(X[i,j]^{-1} - G[j,i])^{-1} G[:,i]G[j,:]\| \\ &= \left\|(X[i,j]^{-1} - G[j,i])^{-1} G[j,:]G[:,i]\right\|. \end{aligned} \tag{3.18}$$

If $G[j,i]$ is small, we can then approximate eq. (3.18) to $\|X[i,j]G[j,:]G[:,i]\|$ and δ_{ij} might be small. However, if $G[j,i]$ is large and is similar to $X[i,j]$ then δ_{ij} might be large.

Remark 3.4.4. *The above analysis indicates that the distance δ_{ij} is related to G and not necessarily small. This also means that the solution with matrix inverse is not robust with respect to noise comparing with the AIC optimisation.*

3.5 Biologically-inspired examples

This section illustrates with two examples the theoretical results presented in the previous section. The corresponding sets of ordinary differential equation describing the dynamics of the networks considered are used to generate noisy data, which are then fed to our reconstruction algorithm in order to assess its ability to recover the correct network structure.

3.5.1 Example 1

In this first example, we consider the following nonlinear system:

$$\dot{y}_1 = -y_1 + \frac{V_{max}}{K_m + z_3^3} + u_1 \tag{3.19}$$

$$\dot{y}_2 = -2y_2 + 1.5z_1 + u_2 \tag{3.20}$$

$$\dot{y}_3 = -1.5y_3 + 0.5z_2 + u_3 \tag{3.21}$$

$$\dot{z}_1 = 0.8y_1 - 0.5z_1 \tag{3.22}$$

$$\dot{z}_2 = 1.2y_2 - 0.8z_2 \tag{3.23}$$

$$\dot{z}_3 = 1.1y_3 - 1.3z_3 \tag{3.24}$$

where $V_{max} = 0.5$ and $K_m = 0.1$. Equation (3.19) includes a nonlinear function of z_3 known as a Hill equation. It represents a negative regulation of the rate of reaction of y_1 by z_3. For simplicity, all other terms are linear. In this example, $p = 3$, i.e. there are three measured states (y_1, y_2 and y_3) while the other 3 states are hidden (z_1, z_2 and z_3). The corresponding network is given in Figure 3.6(a).

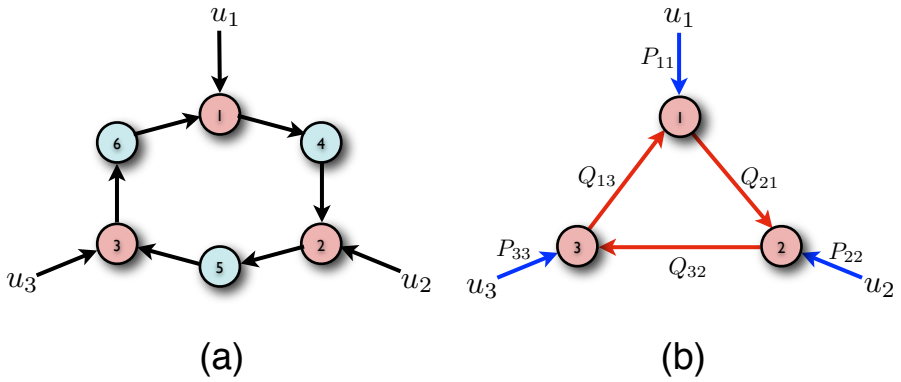


FIGURE 3.6: (a) Complete network with all the states. The red circles represent the measured states while the blue circles correspond to hidden states. (b) Network of the measured states only.

On this model we generate noisy data by numerically integrating the differential equations in (3.19)-(3.24) and adding independent Gaussian noise with zero mean and standard deviation 0.1. Three experiments are performed where one input is a step while the others are set to zero and data are collected for each of the measured species. These experiments are repeated 3 times to average out the noise. In this example only steady-state data (approximated as the final value reached over the considered time interval) are used. Since the true network has 3 elements in Q equal to zero, we expect that there are $2^3 = 8$ networks with a better or equal optimal cost. The results are presented in Table 3.1. Computing the corresponding distances and AICc values for all possible networks between the three measured species we observe that the distance decreases by an order of magnitude when we arrive at the true network. In addition, AIC, BIC and in particular AICc are able to pick the correct network.

3.5.2 Example 2

In this section we consider the application of our method to the reconstruction of the underlying dynamic network responsible for chemotaxis in *Rhodobacter sphaeroides*. The complete network as per today knowledge is represented in Figure 3.7(a) (see [73] for a detailed explanation of this model and its biological interpretation). It involves 10 species dynamically interacting through a complex set of interconnections. As an illustrative example of the application of our method, we consider the case where steady-state data are collected from 3 species only: Y_3^p, Y_6^p and the "motor" (circled in red in Figure 3.7(a)). As a proof of concept of the type of results that our method allows to obtain, we generate data for these 3 species based on simulations of the nonlinear ordinary

Boolean structure	α_k	AIC	AICc	BIC
[0 0 0 0 0 0]	6.94	11	11	2.51
⋮				
[1 1 1 0 1 0]	0.836	12.7	22.7	0.562
[1 1 1 0 1 1]	0.836	14.7	34.7	1.66
[1 0 0 1 1 0]	0.088	3.93	8.73	-7.29
[1 0 0 1 1 1]	0.0879	5.92	15.9	-6.19
[1 1 0 1 1 0]	0.0871	5.9	15.9	-6.22
[1 1 0 1 1 1]	0.0871	7.9	27.9	-5.12
[1 0 1 1 1 0]	0.0866	5.88	15.9	-6.24
[1 0 1 1 1 1]	0.0866	7.88	27.9	-5.14
[1 1 1 1 1 0]	0.0858	7.85	27.8	-5.17
[1 1 1 1 1 1]	0.0858	9.85	51.8	-4.07

TABLE 3.1: The binary values in the table are arranged according to the following order $[Q_{12}\ Q_{13}\ Q_{21}\ Q_{23}\ Q_{31}\ Q_{32}]$. These binary values indicate the presence or absence of a causal relationship (i.e. an edge) between the corresponding elements of the considered Boolean network. The red row indicates the "true" Boolean network we obtain as a result of our reconstruction method.

differential equation model proposed by [73]. Gaussian noise is added to the collected data to simulate measurement noise in the data set.

We follow our prescribed experimental protocol where data are collected for each measured species when a step is imposed on the corresponding input while the other inputs are zero. Again, for simplification, only steady-state data (approximated as the final value reached over the considered time interval) are used. Based on the complete network given in Figure 3.7(a), the correct network that we should aim to recover is presented in Figure 3.7(b).

Computing the corresponding distances and AICc values for all the $2^6 = 64$ possible Boolean networks (Table 3.2), we observe that the network with the smallest AICc (Figure 3.9(e)) is not the correct network in Figure 3.7(b). This is not because the method failed but because of the very low signal to noise ratio observed in the measurements when a step is imposed on u_2 (see Figure 3.8). Y_6^p has a very small influence on Y_3^p since the pathway from Y_6^p to Y_3^p includes a reversible reaction with very small rate constant.

The next set of smallest values of AICc in Table 3.2 consists of 4 networks, including the true one. The corresponding candidate networks for the reconstruction are represented in Figure 3.9. These five candidate networks can then be further distinguished by performing additional and more precise experiments, with reduced noise and increased amplitude (if possible) in some of the connections to help differentiate them.

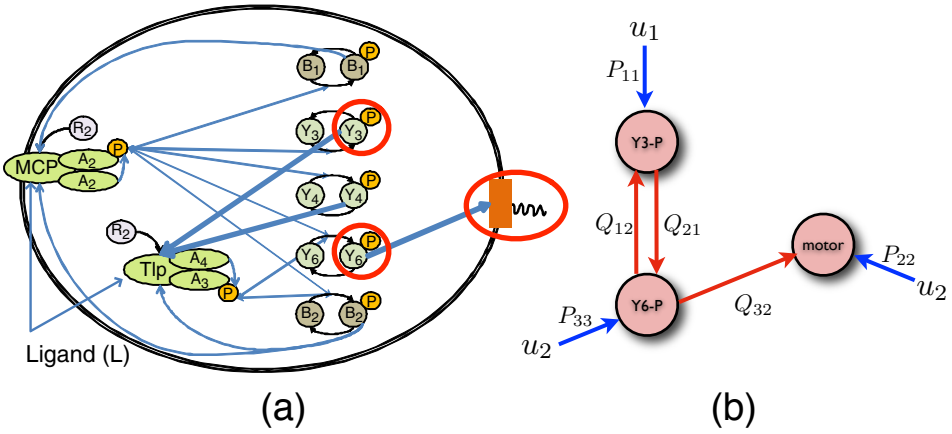


FIGURE 3.7: (a) Network representing the dynamical interaction between the 10 species believed to be responsible for the chemotactic response of *Rhodobacter sphaeroides*. We assume that only species Y_3^p, Y_6^p and "motor" are measured (circled in red). (b) Network connecting the measured states only.

Boolean structure	α_k	AICc
[0 0 0 0 0 0]	1.93	7.18
⋮		
[0 1 1 1 1 1]	0.157	29.7
[1 0 0 1 0 0]	0.0309	0.786
[1 1 0 1 0 0]	0.0306	5.56
[1 0 0 1 0 1]	0.0303	5.52
[1 1 0 1 0 1]	0.03	12.7
[1 0 0 1 1 0]	0.0293	5.43
[1 0 0 1 1 1]	0.029	12.6
[1 1 0 1 1 1]	0.0287	24.6
[1 0 1 1 0 0]	0.0274	5.22
[1 1 1 1 0 0]	0.0271	12.4
⋮		
[1 1 1 1 1 1]	0.0252	48.2

TABLE 3.2: The binary values in the table are arranged according to the following order $[Q_{12}\ Q_{13}\ Q_{21}\ Q_{23}\ Q_{31}\ Q_{32}]$. These binary values indicate the presence or absence of a causal relationship (i.e. an edge) between the corresponding elements of the considered Boolean network. The red row indicates the Boolean network obtained as a result of our reconstruction method, while the blue row indicates the true one.

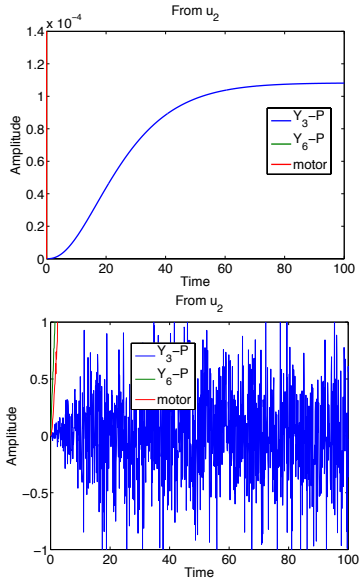


FIGURE 3.8: Above and below are the deterministic and stochastic concentrations of Y_3^p, respectively, in response to a step input in u_2. Note that the amplitude without noise (top) is much weaker than with noise (bottom), and so the signal information is lost.

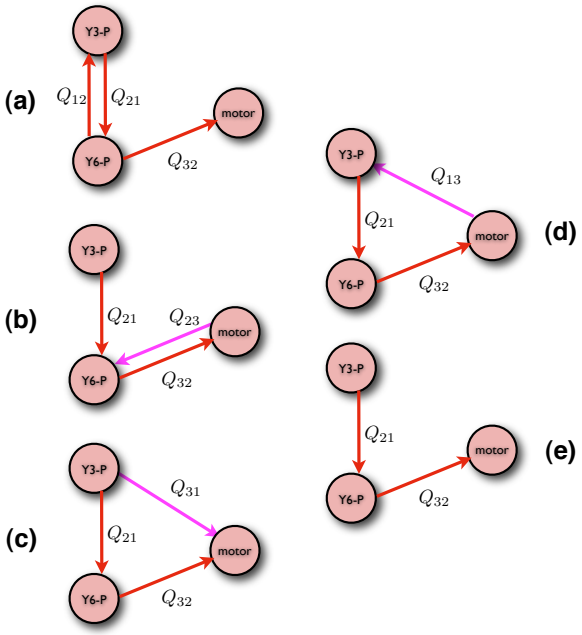


FIGURE 3.9: Candidate networks. Pink arrows represent spurious connections.

3.5.3 Application to real data

We are currently using our algorithm in the exploration of the unknown network structure in circadian clocks [36] and also in proteins in WNT pathway. WNT pathway is one of the key signal transduction pathways with its role in embryogenesis and cancer we are using data obtained from the van Oudenaarden systems biology Lab at MIT to obtain the network of important genes. The available data is knock-out data (as we shall explain more later) by which we could not directly apply our algorithm based on the assumption of perturbation. We here assume that the underlying dynamics is linear, if the system is nonlinear then knockout will potentially change the equilibrium of the system and make the reconstruction impossible. The data we obtain is by knocking out each of the genes in the reconstructed network. Here we only consider the case using steady-state data to reconstruct Boolean networks for simplicity, the result can be easily extended to time-series data.

Assuming that the experiment is performed in the following way for a p-node network: We firstly measure the wild-type data without knocking out/ removing any node in such network, denoting y_0. Secondly, for the i^{th} experiment, we knock out node i and measure the other nodes. And $y_0 \in \mathbb{R}^p$ is the raw wild-type data with the i^{th} element equals to the value of the i^{th} node. Similarly, y_i^d are the raw data for the i^{th} mutant. We also assume that $y_i^d \neq 0$ for any i.

It can be verified that

$$y_0 = Q y_0 + P u. \tag{3.25}$$

When we knockout gene 1, it is equivalent to remove the state of node 1 from the above eq. (3.25), i.e. $y_1^d \triangleq \begin{bmatrix} 0 \\ y_1 \end{bmatrix}$

$$\begin{bmatrix} 0 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} 0 \\ y_1 \end{bmatrix} + \begin{bmatrix} 0 \\ P_2 \end{bmatrix} u \tag{3.26}$$

We can similarly write down the equation for other mutant, if we write down the matrix form. Let $v_1 \triangleq Q_{12} y_1 + P_1 u$, we can then write

$$\begin{bmatrix} v_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} 0 \\ y_1 \end{bmatrix} + \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} u \tag{3.27}$$

Notice that for wild-type data, we have $y_0 = Qy_0 + Pu$. Take the difference, we obtain

$$\begin{bmatrix} y_0[1] - v_1 \\ y_0[2:p] - y_1 \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} y_0[1] \\ y_0[2:p] - y_1 \end{bmatrix}$$

Similarly, we can define v_i for the i^{th} mutant data and write the above in the following matrix form

$$G^d - \text{diag}\{v_1,\ v_2,\ \ldots,\ v_p\} = QG^d \tag{3.28}$$

where

$$G^d \triangleq \begin{bmatrix} y_0[1] & y_0[1] - y_2[1] & \ldots & y_0[1] - y_p[1] \\ y_0[2] - y_1[2] & y_0[2] & \ldots & y_0[2] - y_p[2] \\ \vdots & \vdots & \ddots & \vdots \\ y_0[p] - y_1[p] & y_0[p] - y_2[p] & \ldots & y_0[p] \end{bmatrix}.$$

From eq. (3.28), $(I - Q)G^d = \text{diag}\{v_1,\ v_2,\ \ldots,\ v_p\}$ and it further writes ($H^d \triangleq (G^d)^{-1}$)

$$I - Q = \text{diag}\{v_1,\ v_2,\ \ldots,\ v_p\}H^d. \tag{3.29}$$

Corollary 3.5.1. *If G^d is full rank, and there is no information about the internal structure of the system, Q, then the dynamical structure can be reconstructed as follows*

$$Q[i,j] = -\frac{H^d[i,j]}{H^d[i,i]} \ \textit{and} \ \ v_i = \frac{1}{H^d[i,i]}. \tag{3.30}$$

Proof. Without loss of generality, we look at the first rows of both sides of eq. (3.29).

On the left part, we have $1,\ -Q[1,2],\ \ldots, -Q[1,p]$, compare with the terms on the right part, $v_1H^d[1,1],\ v_1H^d[1,2],\ \ldots,\ v_1H^d[1,p]$. This implies

$$\begin{aligned} 1 = v_1H^d[1,1] &\Rightarrow v_1 = \frac{1}{H^d[1,1]} \\ -Q[1,2] = v_1H^d[1,2] &\Rightarrow Q[1,2] = -\frac{H^d[1,2]}{H^d[1,1]} \\ &\vdots \\ -Q[1,p] = v_1H^d[1,p] &\Rightarrow Q[1,p] = -\frac{H^d[1,p]}{H^d[1,1]}. \end{aligned}$$

We can also look at the other rows of eq. (3.29) and this proves eq. (3.30). □

Here is the proposed algorithm:

Algorithm 6 Network reconstruction using mutant data

$G^d = []$
for $i = 1 : p$ **do**
  $G^d = [G^d; y_0 - y_i^d]$
end for
$H = (G^d)^{-1}$
$Q[i,j] = -\frac{H^d[i,j]}{H^d[i,i]}$

We have now obtained the dynamic structure function Q which gives the interconnection information from mutant data.

Remark 3.5.1. *Again, we can extend our above results to in Section 3.2.1 for a scalable algorithm at the expense of reconstruction accuracy.*

3.5.4 Discussion and summary of this section

There are several tools in the literature which enable us to to infer causal network structures. These tools are mainly rooted in three fields: Bayesian inference, information theory and ODE methods. The vast majority of network reconstruction methods produce estimates of network structure regardless of the informativity of the underlying data. In particular, most methods produce estimates of network structure even in cases with data from only a few experiments. Such data may not contain enough information to enable the accurate reconstruction of the actual network; thus the obtained network estimates can differ arbitrarily from the true network structure. To compensate for lack of information in data, most methods have heuristics that try to "guess" the remaining information, either by specifying prior distributions or by appealing to beliefs about the nature of real biological networks, such as looking for the sparsest network. Nevertheless, these heuristics bias the results and lead to incorrect estimates of the network structure.

By contrast, our approach has been to identify the conditions that obtain when data are sufficiently informative to enable accurate network reconstruction. The results indicate that even in an ideal situation, when the underlying network is linear and time-invariant (LTI) and the measurements are noise-free, network reconstruction is impossible without additional information [34]. Surprisingly, this information gap is not due to a lack of data, nor a deficiency in the number of experiments, but rather it occurs because system states are only partially observed; the information gap is present in all data sets except those that satisfy certain experimental conditions.

3.6 Algorithm to find a (Q,P) minimal realisation

After obtaining the dynamical structure function from input-output data (Step (C)), we start to consider Step (B) in Fig. 3.1, to find out the minimal realisation consistent with the obtained dynamical structure function. There are, of course, a large number of such state-space realisations; we are only interested in those with the lowest complexity, i.e. those with a minimal number of states. The underlying principle in finding a (Q,P) minimal realisation is to search for a realisation with the minimal number of hidden states. Such a realisation is characterised by the minimal number of pole-zero cancellations in the transfer functions Q and P.

Proposition 3.6.1. *Given a dynamical system (3.1) and the associated dynamical structure functions (Q,P) with D^o constructed as explained above (see* (3.1)-(3.6)*), the following conditions must hold*

$$diag\{A_{11}^o\} = \lim_{s\to\infty} D^o(s); \tag{3.31}$$

$$A_{11}^o - diag\{A_{11}^o\} = \lim_{s\to\infty} sQ(s); \tag{3.32}$$

$$B_1^o = \lim_{s\to\infty} sP(s). \tag{3.33}$$

Proof. Eq. (3.31) is directly obtained from the definition of $D^o(s)$:

$$\begin{aligned}\lim_{s\to\infty} D^o(s) &= \lim_{s\to\infty} \text{diag}\{W^o(s)\} \\ &= \text{diag}\{\lim_{s\to\infty} W^o(s)\} = \text{diag}\{A_{11}^o\}\end{aligned}$$

Since the proofs for eq. (3.32) and (3.33) are very similar, we focus on eq. (3.32) only. Using the fact that for any square matrix M, if $M^n \to 0$ when $n \to +\infty$, then $(I-M)^{-1} = \sum_{i=0}^{\infty} M^i$, we obtain, from the definition of Q given in (3.5), $Q(s) = \sum_{i=1}^{\infty} s^{-i} D^{o\ i-1}(W^o - D^o)$ and $W^o = A_{11}^o + \sum_{i=1}^{\infty} s^{-i} A_{12}^o A_{22}^{o\ i-1} A_{21}^o$, when $s \to +\infty$. Hence, $Q(s) = (A_{11}^o - D^o(s))s^{-1} + r(s)$, in which $r(s)$ is a matrix polynomial of s, whose largest degree is -2. Finally, multiplying by s on both sides and taking the limit as s goes to ∞ results in eq. (3.32). A similar argument can be used to prove eq. (3.33). □

Remark 3.6.1. *Proposition 3.6.1 concludes that all the realisations consistent with (Q,P) share the same matrices A_{11}^o (minus its diagonal) and B_1^o in eq.* (3.1)*. Hence consistency with (Q,P) imposes constraints on the realisations that need to be considered.*

There are many realisations consistent with (Q,P). In the following section, we focus

on finding a (Q,P) minimal realisation, i.e. a realisation which is consistent with (Q,P) and which has minimal order (and hence the lowest possible complexity).

From a dynamical structure function (Q,P) we cannot reconstruct (W^o,V^o) since there is no information regarding the diagonal transfer function matrix D^o. We now discuss properties of realisations obtained from transfer matrices (W,V) consistent with (Q,P). We start with an arbitrarily chosen D, and then use a state-space realisation approach to find a D which minimises the order of a minimal realisation of $[W \;\; V]$.

Given (Q,P) and a diagonal proper transfer function matrix D, consider a minimal realisation of $[W \;\; V] = [(sI-D)Q+D \;\; (sI-D)P]$. On the other hand, $[W, \;\; V]$ has the following minimal realisation

$$[W \;\; V] = [A_{11} \;\; B_1] + A_{12}(sI - A_{22})^{-1}[A_{21} \;\; B_2] \tag{3.34}$$

Lemma 3.6.1. *Given a dynamical structure function (Q,P) and a diagonal proper transfer matrix D, the realisation (A,B) obtained from eq.* (3.34) *is consistent with (Q,P) and the pair $(A, \begin{bmatrix} I_p & 0 \end{bmatrix})$ is observable.*

Proof. The consistency of the realisation with (Q,P) follows from the definition of (Q,P). From the Popov-Belevitch-Hautus (PBH) rank test [107], a matrix pair $(A \in \mathbb{R}^{l\times l}, C)$ is observable iff

$$\operatorname{rank} \begin{bmatrix} sI - A \\ C \end{bmatrix} = l, \tag{3.35}$$

for all $s \in \mathbb{C}$. A minimal realisation of $\begin{bmatrix} W & V \end{bmatrix}$ implies that the pair (A_{22}, A_{12}) is observable, i.e.

$$\operatorname{rank} \begin{bmatrix} sI_{l-p} - A_{22} \\ A_{12} \end{bmatrix} = l - p,$$

Hence

$$\operatorname{rank} \begin{bmatrix} sI - A_{11} & -A_{12} \\ -A_{21} & sI_{l-p} - A_{22} \\ I_p & 0_{p\times(l-p)} \end{bmatrix} = l,$$

which concludes the proof. □

Remark 3.6.2. *Given matrices A and B obtained in eq.* (3.34)*, the dimension of A is equal to the dimension of a minimal realisation of G iff the pair (A,B) is controllable.*

Given a dynamical structure function (Q,P), a random choice of a proper diagonal transfer function matrix D will lead to additional zeros in G which are associated with uncontrollable eigenvalues of the considered realisation [19].

At this stage the following question arises: how can we find a proper diagonal transfer function matrix D such that a minimal realisation of $[W \quad V]$ is a (Q,P) minimal realisation? Note that, since there are many choices for D that minimise the order of minimal realisations of $[W \quad V]$, a chosen D may be different from D^o.

To answer this question, first note that for all D, $[W \quad V]$ can be written as

$$[W \quad V] = (sI - D)s^{-1}[sQ \quad sP] + [D \quad 0]. \tag{3.36}$$

Assume that all elements in $[Q \quad P]$ have only simple poles. This assumption can be relaxed but we adopt it here for simplicity. Also we assume that $[Q \quad P]$ does not possess any poles at 0 (otherwise we can change eq. (3.36) to $(sI-D)(s-a)^{-1}[(s-a)Q \quad (s-a)P] + [D \quad 0]$, where $a \in \mathbb{R}$ is not a pole of $[Q \quad P]$). In this case we shall show that a minimal order realisation of $[W \quad V]$ can always be found using a constant diagonal matrix D.

Proposition 3.6.2. *Assume* $[I-Q \quad P]$ *only has simple poles and does not have any zeros*[2]. *A minimal order realisation of* $[W \quad V]$ *in* (3.34) *can be achieved using a constant diagonal matrix* D.

Proof. Assume D has at least one term on the diagonal with the degree of the numerator greater or equal to 1, e.g., suppose the i^{th} term in $(sI-D)s^{-1} = \frac{(s+b)\epsilon_i(s)}{s\phi_i(s)}$ with any $b \in \mathbb{R}$ and $deg(\epsilon_i(s)) = deg(\phi_i(s)) \geq 1$, where $deg(\cdot)$ returns the degree of a polynomial. Hence, the product $(sI-D)s^{-1}[sQ \quad sP]$ will introduce $deg(\phi_i(s))$ new poles and, due the assumption of simple poles, can at most eliminate $deg(\epsilon_i(s)) = deg(\phi_i(s))$ poles since $[I-Q \quad P]$ does not have any zeros. As a consequence, we can change the i^{th} term from $\frac{(s+b)\epsilon_i(s)}{s\phi_i(s)}$ to $\frac{s+a}{s}$ without increasing the order. Doing this along all the elements of D proves the result. □

If D is a constant matrix, the term $[D \quad 0]$ in eq. (3.36) is also a constant matrix. Therefore the order of a minimal realisation is only determined by $(sI-D)s^{-1}[sQ \quad sP] \triangleq N[sQ \quad sP]$. Thus, finding the "optimal" D which leads to the minimal order in eq. (3.36) is equivalent to finding a diagonal proper transfer matrix N (with corresponding minimal realisation (A_2,B_2,C_2,D_2)) such that it cancels as many poles of $[sQ \quad sP]$ as possible. Based on this idea, the following algorithm is proposed:

Step 1: *Find a Gilbert's realisation of the dynamical structure function.*
First we find a minimal realisation (A_1,B_1,C_1,D_1) of $[sQ \quad sP]$. When $[sQ \quad sP]$ has l simple poles, using Gilbert's realisation [30] gives

$$[sQ \quad sP] = \sum_{i=1}^{l} \frac{K_i}{s-\lambda_i} + \lim_{s\to\infty}[sQ \quad sP], \tag{3.37}$$

[2]These assumption can be relaxed as discussed in the later sections

where $K_i = \lim_{s\to\lambda_i}(s-\lambda_i)[sQ \;\; sP]$ and has rank 1 since we are assuming that $[Q \;\; P]$ has simple poles.

Consider a matrix decomposition of K_i in the following form:

$$K_i = E_i F_i, \; \forall i, \tag{3.38}$$

where $E_i \in \mathbb{R}^l$ and $F_i = (E_i^T E_i)^{-1} E_i^T K_i$. Then $A_1 = \text{diag}\{\lambda_i\} \in \mathbb{R}^{l\times l}$, $B_1 = \begin{bmatrix} F_1^T & F_2^T & \dots & F_l^T \end{bmatrix}^T$, $C_1 = \begin{bmatrix} E_1 & E_2 & \dots & E_l \end{bmatrix}$ and $D_1 = \lim_{s\to\infty}[sQ \;\; sP]$.

Step 2: *Find the maximal number of cancelled poles.*

We define Φ as a largest subset of $\{\mathscr{B}(E_1),\dots,\mathscr{B}(E_l)\}$ such that all the elements in Φ are mutually orthogonal. We also define ϕ as the cardinality of Φ. Computationally, ϕ can be obtained using the algorithm presented in the Appendix. We claim that ϕ is equal to the maximum number of poles we can eliminate (the proof is in Appendix). Therefore the minimal order of $[W \;\; V]$ is

$$l - \phi.$$

In consequence, the order of the minimal reconstruction is the dimension of A_{11} (constant p) plus the minimal dimension of A_{22} (obtained above): $p + l - \phi$.

Step 3: *Construct D to obtain the minimal reconstruction.*

Once we have Φ, using eq. (3.69) and $D = sI - sN$, we know that $N(\lambda_i)[j,j] = 0$ implies $D[j,j] = \lambda_i$. Consequently, each element in the set Φ will determine at least one element in D. This last fact can be used to construct D element by element. Once D is found, we can obtain A, B using eq. (3.34).

Example 3.6.1. *Consider the dynamical structure function (Q,P):*

$$[Q \mid P] = \left[\begin{array}{ccc|c} 0 & \frac{1}{s+2} & \frac{1}{s+3} & \frac{1}{s+4} \\ \frac{1}{s+1} & 0 & \frac{1}{s+3} & \frac{1}{s+4} \\ \frac{1}{s+1} & \frac{1}{s+2} & 0 & \frac{1}{s+4} \end{array}\right].$$

The different steps of the algorithm proposed in the previous section successively yield the following:

Step 1: *A minimal Gilbert realisation of $s[Q,P]$ is*

$$A_1 = diag\{-1,-2,-3,-4\},\; B_1 = diag\{2,2,2,4\},$$
$$C_1 = \begin{bmatrix} 0 & -1 & -1.5 & -1 \\ -0.5 & 0 & -1.5 & -1 \\ -0.5 & -1 & 0 & -1 \end{bmatrix},\; D_1 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

Step 2: *By definition,* $E_i = C_1 v_i$ *where* $v_i \in \mathbb{R}^4$ *has* 1 *in its* i^{th} *position and zero otherwise. Thus,*

$$\{\mathscr{B}(E_1),\ldots,\mathscr{B}(E_4)\} = \left\{ \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\}.$$

Furthermore, ϕ *is* 1 *and the order of a minimal realisation of the given dynamical structure function is* $p+l-\phi = 3+4-1 = 6$*. Hence, the system must contain at least* 3 *hidden states.*

Step 3: D *can be chosen as* $diag\{a,-1,-1\}$*,* $diag\{-2,a,-2\}$*,* $diag\{-3,-3,a\}$*, or* $diag\{-4,-4,-4\}$ *for any* $a \in \mathbb{R}$*.*

From a biological perspective this indicates that there are at least 3 *unmeasured species interacting with the measured species. Of course, the "true" biological system might be even more complicated, i.e. it might have more than* 6 *states. Yet when more states are measured the dynamical structure functions can be easily updated and a new search for a minimal realisation of the updated system can be performed to reveal the corresponding minimum number of hidden states.*

3.6.1 Discussion on assumptions

In this section, we shall discuss how strong the assumption that $[I-Q, P]$ has zeros is.

Definition 3.6.1. *[107] Let* $G(s)$ *be a* $p \times m$ *proper transfer matrix with full row normal rank(defined in [107]). Then* $z_o \in \mathbb{C}$ *is a transmission zero of* $G(s)$ *if and only if there exists a* $0 \neq \eta_0 \in \mathbb{C}^p$ *such that* $\eta_0^T G(z_0) = 0$*.*

We also have the following Proposition.

Proposition 3.6.3. *Let* Z_{I-Q} *be the set of zeros of* $I-Q$ *and* Z_P *be that of* P*, then the set of zeros of* $[I-Q,\ P]$*, i.e.,* $Z_{[I-Q,P]}$ *is contained in the set of intersection of* Z_{I-Q} *and* Z_P*.*

Proof. This can be easily shown using definition 3.6.1. □

Remark 3.6.3. *Not all the zeros in* $Z_{I-Q} \cap Z_P$ *are zeros of* $Z_{[I-Q,P]}$*.*

3.6.2 Specific result

With respect to whether $[I-Q,\ P]$ might have zeros, the above argument works for random P and Q. Based on a large number of numerical experiments and because of the way Q and P are generated it seems that generically over random A,B,C,D there will be zeros.

This is because

$$(I-Q,\ P) = (sI-D^o)^{-1}(sI-W^o,\ V^o), \tag{3.39}$$

since $sI-D^o$ is a diagonal transfer matrix, then every zeros of square diagonal matrix $(sI-D^o)^{-1}$ (poles of $sI-D^o$).

Proposition 3.6.4. *If a zero z_0 of the i^{th} element of $(sI-D^o)^{-1}$, i.e. $(sI-D^o)^{-1}[i,i]$ is not a pole of every $(sI-W^o,\ V^o)[i,j]$ $j \neq i$, then z_0 is a zero of $(I-Q,\ P)$.*

Proof. This can be easily shown by definition. □

Remark 3.6.4. *Proposition 3.6.4 is equivalent to say, $[I-Q,\ P]$ has zeros if and only if there exist $i,\ j \in \{1,\ 2,\ \ldots,\ p\}$ such that a pole p_0 of $(sI-W^o)[i,i]$ is not a pole of $(sI-W^o,V^o)[i,j]$.*

By looking at the definitions of

$$D^o = \text{diag}\{W^o\} = \text{diag}(W^o_{11}, W^o_{22}, ..., W^o_{pp})$$

$$[W^o\ \ V^o] = [A^o_{11}\ \ B^o_1] + A^o_{12}(sI-A^o_{22})^{-1}[A^o_{21}\ \ B^o_2], \tag{3.40}$$

and since the poles of $sI-W^o$ are the same as W^o, we then have

$$\begin{aligned}
W^o[i,i] &= e_i^T\left(A^o_{11} + A^o_{12}(sI-A^o_{22})^{-1}A^o_{21}\right)e_i \\
&= A^o_{11}[i,i] + A^o_{12}[i,:](sI-A^o_{22})^{-1}A^o_{21}[:,i] \\
W^o[i,j] &= A^o_{11}[i,i] + A^o_{12}[i,:](sI-A^o_{22})^{-1}A^o_{21}[:,j] \\
V[i,k] &= A^o_{11}[i,i] + A^o_{12}[i,:](sI-A^o_{22})^{-1}B^o_2[:,k]
\end{aligned}$$

For simplicity, let A^o_{22} diagonalisable, i.e., $A^o_{22} = UJU^{-1}$ where J is a diagonal matrix. We can substitute this to the above equations and then we have

Proposition 3.6.5. *$[I-Q,\ P]$ does not have zeros if and only if for all i,j,k*

$$\mathscr{B}\{UA^o_{21}[:,j]\} - \mathscr{B}\{UA^o_{21}[:,i]\} \geq 0 \tag{3.41}$$

$$\mathscr{B}\{UB^o_2[:,k]\} - \mathscr{B}\{UA^o_{21}[:,i]\} \geq 0 \tag{3.42}$$

Here a vector $v \geq 0$ means, all the elements in v are greater or equal to 0.

Proof. From above, we have that

$$\begin{aligned} W^o[i,i] &= A^o_{11}[i,i] + A^o_{12}[i,:]U^{-1}(sI-J)^{-1}UA^o_{21}[:,i] \\ W^o[i,j] &= A^o_{11}[i,i] + A^o_{12}[i,:]U^{-1}(sI-J)^{-1}UA^o_{21}[:,j] \\ V[i,k] &= A^o_{11}[i,i] + A^o_{12}[i,:]U^{-1}(sI-J)^{-1}UB^o_2[:,k]. \end{aligned}$$

From Proposition 3.6.4, $[I-Q,\ P]$ has zeros if and only if there exist $i,\ j \in \{1,\ 2,\ \ldots,\ p\}$ such that a pole p_0 of $(sI-W^o)[i,i]$ is not a pole of $(sI-W^o, V^o)[i,j]$.

Given i, since J is diagonal, $[I-Q,\ P]$ has zeros can only be happened if a nonzero element in the r^{th} position of $UA^o_{21}[:,i]$, there exists a l such that either $(UA^o_{21}[:,l])[r]=0$ or $(UB^o_2[:,l])[r]=0$. □

In the next Section 3.6.3, we shall propose an algorithm that finds a minimal structural realisation of $(Q,\ P)$.

3.6.3 Scenario that is not included in the Proposition

We found that for random choice of $(A,\ B,\ C,\ D)$, the probability of $[I-Q,\ P]$ has zero was high. This indicates that due to the special construction of $[I-Q,\ P]$, the assumption in Proposition 3.6.2 is actually strong. We shall now consider the scenario when $[I-Q,\ P]$ has zeros.

To simplify the analysis, we have the following observation

$$[I-W/s,\ V/s] = (I-D/s)[I-Q,\ P], \tag{3.43}$$

when $Q,\ P$ does not posses any poles at 0 (otherwise can change eq. (3.43) to $[s/(s-a)I - W/(s-a),\ V/(s-a)] = (sI-D)(s-a)^{-1}[I-Q\ P]$, where $a \in \mathbb{R}$ is not a pole of $[I-Q\ P]$), we can see that

$$\deg[W,\ V] = \deg[I-W/s,\ V/s] - p,$$

therefore

$$D^* = \mathbf{argmin}_D\ \deg[I-W/s, V/s] = \mathbf{argmin}_D\ \deg\{(I-D/s)[I-Q,\ P]\}. \tag{3.44}$$

We define $N(s) = (sI-D)s^{-1}$.

Remark 3.6.5. *Eq.* (3.44) *reduces the complexity in determining* D^* *as seen from Fig. 3.10 and Fig. 3.11.*

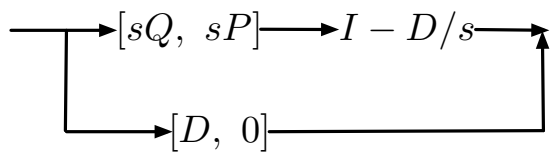


FIGURE 3.10: Cascaded and paralleled expression for $[W,\ V]$ in eq. (3.34).

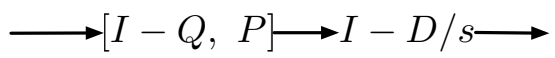


FIGURE 3.11: Cascaded expression for $[I - W/s,\ V/s]$ in eq. (3.44).

The difference from Proposition 3.6.2 ($[I-Q,\ P]$ does not have zeros) is that it now allows to have poles for $I - D/s$, that is to say, the solution D^* in eq. (3.44) will no longer be a constant matrix.

Assume that a minimal realisation of $[I-Q,\ P]$: $C_1(A_1 - sI)^{-1}B_1 + [I,\ 0]$. If a zero of $[I-Q,\ P]$, say λ_i, is cancelled by cascading another system $N(s) = (sI - D^*)s^{-1} \triangleq C_2(A_2 - sI)^{-1}B_2 + I$, then the realisation of the cascaded system $(sI - D^*)s^{-1}[I - Q\ P]$ loses controllability. In this case, it follows that there exists a nonzero vector $z_i^T = [z_{1,i}^T, z_{2,i}^T]$ such that

$$\begin{bmatrix} z_{1,i}^T & z_{2,i}^T \end{bmatrix} \begin{bmatrix} A_1 - \lambda_i I & 0 & B_1 \\ B_2 C_1 & A_2 - \lambda_i I & B_2[I,\ 0] \end{bmatrix} = 0.$$

This shows that $z_{2,i}$ is an eigenvector of A_2 corresponding to λ_i. Since A_2 is diagonal, we can directly compute $z_{2,i}^T = \begin{bmatrix} 0 & \dots & 0 & 1_{i^{th}} & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{1\times l}$. Therefore we have

$$\begin{bmatrix} z_{1,i}^T & z_{2,i}^T B_2 \end{bmatrix} \begin{bmatrix} A_1 - \lambda_i I & B_1 \\ C_1 & [I,\ 0] \end{bmatrix} = 0.$$

$$\begin{bmatrix} A_1 - sI & B_1 \\ C_1 & [I,\ 0] \end{bmatrix} \begin{bmatrix} I & -(A_1 - sI)^{-1}B_1 \\ 0 & I \end{bmatrix} = \begin{bmatrix} A_1 - sI & 0 \\ C_1 & [I - Q(s),\ P(s)] \end{bmatrix},$$

We obtain, since $\lambda_i \neq 0$ is not a pole of $N(s)$,

$$z_{2,i}^T B_2 [I - Q(\lambda_i),\ P(\lambda_i)] = 0 \tag{3.45}$$

By comparing the definition of zeros of $[I - Q(\lambda_i),\ P(\lambda_i)]$, we can see that $z_{2,i}^T B_2$ is a vector that predetermined by the zero direction of $[I - Q(\lambda_i),\ P(\lambda_i)]$. Recall that a minimal

realisation of diagonal transfer matrix can also be obtained by Gilbert realisation with

$$A_2 = \text{diag}[a_1,\ a_2,\ \ldots,\ a_k],\ B_2 = \text{diag}\begin{bmatrix} b_1 & b_2 & \ldots & b_k \end{bmatrix},\ C_2 = B_2^T,$$

where $(a_m \in \mathbb{R}^{k_m \times k_m}, b_m \in \mathbb{R}^{k_m \times 1}, b_m^T)$ is the minimal realisation of the the m^{th} transfer function on the diagonal.

From eq. (3.45), we have the following equality (let $\mathscr{B}(\cdot)$ be the Boolean operator which maps a matrix/vector to a Boolean one)

$$\mathscr{B}(z_{2,i}^T B_2) = \mathscr{B}(v_i^T), \tag{3.46}$$

where v_i^T is the corresponding zero direction of $[I - Q(\lambda_i),\ P(\lambda_i)]$. This imposes constraint on cancelling the zeros of $[I - Q,\ P]$. To cancel zero λ_i with direction v_i^T with nonzero elements $i_1, i_2, \ldots, i_j$, then we should have the $i_1,\ i_2, \ldots, i_j^{th}$ elements of $N(s)$ have λ_i as a pole.

Based on the analysis above, we now propose the principle of designing zeros for $I - D^*/s$. If a pole of $[I - Q,\ P]$, say λ_i, is cancelled by $N(s) = (sI - D^*)s^{-1} \triangleq C_2(A_2 - sI)^{-1}B_2 + I$, then the realisation of the cascade $(sI - D)s^{-1}[I - Q\ P]$ loses observability. In this case, it follows that there exists a nonzero vector $w_i = [w_{1,i}^T, w_{2,i}^T]^T$ such that

$$\begin{bmatrix} A_1 - \lambda_i I & 0 \\ B_2 C_1 & A_2 - \lambda_i I \\ C_1 & C_2 \end{bmatrix} \begin{bmatrix} w_{1,i} \\ w_{2,i} \end{bmatrix} = 0.$$

The first equation shows that $w_{1,i}$ is an eigenvector of A_1 corresponding to λ_i. Since A_1 is diagonal, we can directly compute $w_{1,i}^T = \begin{bmatrix} 0 & \ldots & 0 & 1_{i^{th}} & 0 & \ldots & 0 \end{bmatrix} \in \mathbb{R}^{1 \times l}$. Therefore we have

$$\begin{bmatrix} A_2 - \lambda_i I & B_2 \\ C_2 & I \end{bmatrix} \begin{bmatrix} w_{2,i} \\ C_1 w_{1,i} \end{bmatrix} = 0.$$

Noticing that $C_1 w_{1,i} = E_i$ and that

$$\begin{bmatrix} I & 0 \\ -C_2(A_2 - sI)^{-1} & I \end{bmatrix} \begin{bmatrix} A_2 - sI & B_2 \\ C_2 & I \end{bmatrix} = \begin{bmatrix} A_2 - sI & B_2 \\ 0 & N(s) \end{bmatrix},$$

we obtain, since $\lambda_i \neq 0$ is not a pole of $N(s)$,

$$N(\lambda_i) E_i = 0. \tag{3.47}$$

In summary, designing D^* to cancel any pole λ_i of $[I-Q \;\; P]$ is equivalent to imposing that eq. (3.47) holds. The next question is: given $[I-Q \;\; P]$, what is the maximal number of poles that can be cancelled by left multiplication of $N(s)$, i.e. what is the largest number of poles for which eq. (3.47) is satisfied?

Remark 3.6.6. *The Boolean structure of E_i, $\mathscr{B}(E_i)$ imposes constraints on the diagonal terms in $N(s)$ for cancelling the poles of $[I-Q,\ P]$.*

We are aiming to have all zeros of $[I-Q,\ P]$ as poles of $(I-D/s)$, in this case, we can then have more poles of $[I-Q,\ P]$ cancelled without introducing new poles. We know that there are elements on the diagonal can have a pole/poles. For those elements, we have more degrees of freedom to choose a larger number of zeros of $N(s)$ to cancel the poles of $[I-Q,\ P]$.

To maximise the the number of poles in $[I-Q,\ P]$ that can be cancelled, we use the following table. Table 3.3 is a table generated by the poles of $[I-Q,\ P]$ and the

Poles	Place 1	Place 2	...	Place $p-1$	Place p
p_1^0	1	0	...	1	0
p_2^0	1	0	...	0	1
p_3^0	1	0	...	0	1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
p_{l-1}^0	0	0	...	0	1
p_l^0	0	0	...	1	1

TABLE 3.3: Table for computing pole cancellation, the 1/0 is illustrative but does not have any meaning. p_i^o are labelled in the following way as they appeared in vec($[I-Q,\ P]$) (from top to down).

requirement to cancel each of them from eq. (3.47). More specifically, the rows are the poles of $[I-Q,\ P]$ in any order and we then try to maximise the largest number of rows that can be chosen such that for any column, the summation of the elements on selected rows is less or equal to some constraint; or in other word, how to cancel the largest number of poles without introducing more poles in the cascaded system in Fig. 3.11.

Mathematically, let $s[j]$ be the number of zeros that allowed for the jth diagonal element in $N(s)$ and let $T_{i,j} \in \{0,1\}$ be the binary element in the ith row and jth column

of Table 3.3, then

$$\begin{aligned} &\max\ k \\ &\text{s.t.,}\ \sum_{h=1}^{k} T_{i_h,j} \le s[j],\ \forall j. \\ &\quad \{i_1,\ldots,i_k\} \subseteq \{1,\ldots,l\}. \end{aligned} \tag{3.48}$$

When the number of poles is small, the problem is easy to solve, we can use the exhaustive attack method to go through all the possible cases and find the largest k. But in general, it is an integer optimisation problem and can be viewed as a n-dimensional Knapsack problem and therefore NP-hard.

The solution to the above optimisation problem goes beyond the scope of this thesis. Once we have determine the $\{i_1,\ \ldots,\ i_k\}$ and the we can compute the corresponding zeros and poles of $N(s)$. Similarly as in the main context, once $N(s)$ is fixed, we can then determine D^*.

From above analysis, we have the following algorithm when $[I-Q,\ P]$ has a zero/zeros.

Algorithm 7 Minimal $[Q,\ P]$ realisation when $[I-Q,\ P]$ has a zero/zeros

Step 3.6.1. *Compute the zeros z_i^0 of $[I-Q,\ P]$ and the corresponding directions v_i^T. Take the Boolean structure $\mathscr{B}(v_i^T)$, and define a vector that $s = \sum \mathscr{B}(v_i^T) + 1^T$;*

Step 3.6.2. *Find a Gilbert realisation of $[I-Q,\ P]$ and find the conditions in eq.* (3.47) *for cancelling the poles p_i^0;*

Step 3.6.3. *Build a table for the cancelling conditions from Step* 2 *and compute the maximum number of poles that can be cancelled from eq.* (3.48) *and $[W,\ V]$;*

Step 3.6.4. *Compute D^* based on the determined positions and values of zeros/poles.*

We shall illustrate the Algorithm 7 on this example.

Example 3.6.2.

$$Q = \begin{pmatrix} 0 & 0 & 0 \\ \frac{s+1}{(s+1)^3+1} & 0 & 0 \\ 0 & \frac{1}{(s+1)(s+2)} & 0 \end{pmatrix}, P = \begin{pmatrix} \frac{1}{s+3} & 0 \\ 0 & \frac{(s+1)^2}{(s+1)^3+1} \\ 0 & 0 \end{pmatrix}$$

Step 3.6.1. *Compute the zeros and corresponding zero directions of $[I-Q,\ P]$. It has one*

zero at -1 *and a corresponding zero direction at* $[0,\ 1,\ 0]$. *From eq.* (3.46) *we can see that* $s = [1,\ 2,\ 1]$.

Step 3.6.2. *Gilbert realisation of* $[I-Q,\ P]$

$$A_1 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -.5+0.866i & 0 & 0 \\ 0 & 0 & 0 & 0 & -.5-0.866i & 0 \\ 0 & 0 & 0 & 0 & 0 & -3 \end{bmatrix}, D_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{3.49}$$

$$B_1 = \begin{bmatrix} 0 & 1.41 & 0 & 0 & 0 \\ 0 & 2.24 & 0 & 0 & 0 \\ 0.408 & 0 & 0 & 0 & 0.408 \\ 0.488-0.423i & 0 & 0 & 0 & -0.61-0.211i \\ 0.488+0.423i & 0 & 0 & 0 & -0.61+0.211i \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{3.50}$$

$$C_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0.816 & -0.488+0.169i & -0.488-0.169i & 0 \\ -0.707 & 0.447 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.51}$$

Based on the above analysis, we can draw the following Table 3.5. From Table 3.5, we see that to cancel pole p_1^0, *we need have a constraint on the first diagonal element of* $N(s)$, *similarly for other poles.*

Poles	Place 1	Place 2	Place 3
$p_1^0 = -3$	1	0	0
$p_2^0 = -2$	0	1	0
$p_3^0 = -.5+0.866i$	0	1	0
$p_4^0 = -.5-0.866i$	0	1	0
$p_5^0 = -1$	0	0	1
$p_6^0 = -2$	0	0	1

TABLE 3.4: Table for computing maximum cancelled poles by choosing N.

Step 3.6.3. *To solve the following optimisation problem,* $T_{i,j} \in \{0,1\}$ *be the binary element*

in the ith row and jth column of Table 3.5

$$\begin{aligned} &\max\ k \\ &s.t.,\ \sum_{h=1}^{k} T_{i_h,j} \le 1,\ \forall j = 1,3. \\ &\sum_{h=1}^{k} T_{i_h,j} \le 2,\ \forall j = 2. \\ &\{i_1,\ldots,i_k\} \subseteq \{1,\ldots,l\}. \end{aligned}$$

We know that the optimal solution is $k = 4$, then the dimension of A is $p+l-k = 3+6-4 = 5$. We can choose the solution to be either $\{i_1,\ldots,i_k\} = \{1,3,4,5\}$ or $\{i_1,\ldots,i_k\} = \{1,3,4,6\}$[3] *and*

$$N_1(s) = diag\left[k_1\frac{s+3}{s},\ k_2\frac{s^2+s+1}{s^2+s},\ k_3\frac{s+1}{s}\right]$$

or

$$N_2(s) = diag\left[k_1\frac{s+3}{s},\ k_2\frac{s^2+s+1}{s^2+s},\ k_3\frac{s+1}{s}\right],$$

where k_is are nonzero parameters. Let $k_i = 1$, then

$$[I - W_1/s,\ V_1/s] = N_1[I - Q,\ P] = \begin{bmatrix} \frac{s+3}{s} & 0 & 0 & \frac{1}{s} & 0 \\ \frac{-1}{s(s+2)} & \frac{(s+1)^2}{s(s+2)} & 0 & 0 & \frac{s+1}{s^2+2s} \\ 0 & \frac{-1}{s^2+2s} & \frac{s+1}{s} & 0 & 0 \end{bmatrix},$$

further we have

$$[W_1,\ V_1] = \begin{bmatrix} -3 & 0 & 0 & 1 & 0 \\ \frac{1}{s+2} & \frac{-1}{s+2} & 0 & 0 & \frac{s+1}{s+2} \\ 0 & \frac{-1}{s+2} & -1 & 0 & 0 \end{bmatrix}.$$

Similarly, we can obtain $[W_2,\ V_2]$.

Step 3.6.4. *We finally obtain*

$$D_1^* = sI - sN_1 = diag\left[s - k_1(s+3),\ s - k_2\frac{s^2+s+1}{s+1},\ s - k_3(s+1)\right] = {}^4diag\left[-3,\ \frac{1}{s+1},\ -1\right].$$

[3] since $[W,\ V]$ requires to have real coefficients

[4] since D is required to be proper

or

$$D_2^* = sI - sN_2 = diag\left[s - k_1(s+3),\ s - k_2\frac{s^2+s+1}{s+1},\ s - k_3(s+2)\right] = {}^5diag\left[-3,\ \frac{1}{s+1},\ -2\right].$$

Remark 3.6.7. *The obtained W_i, V_i differs from W^o, V^o, but they are with the same degree. I will continue pursue whether I could parameterise all the possible $[W,\ V]$ with order $p+l-k=5$.*

Example 3.6.3. *Assume*

$$Q = \begin{bmatrix} 0 & -\frac{7(s+2)}{s^2+s+1} \\ -\frac{s+5}{s^2+2} & 0 \end{bmatrix},\ P = \begin{bmatrix} \frac{s-4}{s^2+s+1} & 0 \\ 0 & \frac{s-4}{s^2+2} \end{bmatrix}.$$

Step 3.6.1. *Compute the zeros and corresponding zero directions of $[I-Q,\ P]$. It has one zero at 4 and a corresponding zero direction at $[1,\ 2]$. From eq. (3.46) we can see that $s=[2,\ 2]$.*

Step 3.6.2. *Gilbert realisation of $[I-Q,\ P]$*

$$A_1 = \begin{bmatrix} -0.5+0.866i & 0 & 0 & 0 \\ 0 & -0.5-0.866i & 0 & 0 \\ 0 & 0 & 1.4142i & 0 \\ 0 & 0 & 0 & -1.4142i \end{bmatrix}, D_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (3.52)$$

$$B_1 = \begin{bmatrix} 0 & 2.4749 & -0.7071+0.6124i & 0 \\ 0 & 2.4749 & -0.7071-0.6124i & 0 \\ 0.7655+0.2165i & 0 & 0 & -0.6124+0.2165i \\ 0.7655-0.2165i & 0 & 0 & -0.6124-0.2165i \end{bmatrix}, \quad (3.53)$$

$$C_1 = \begin{bmatrix} 1.4142-2.4495i & 1.4142+2.4495i & 0 & 0 \\ 0 & 0 & 0-2.3094i & 0+2.3094i \end{bmatrix}. \quad (3.54)$$

Based on the above analysis, we can draw the following Table 3.5. From Table 3.5, we see that to cancel pole p_1^0, we need have a constraint on the first diagonal element of $N(s)$, similarly for other poles.

[5] since D is required to be proper

Poles	Place 1	Place 2
$p_1^0 = -0.5 + 0.866i$	1	0
$p_2^0 = -0.5 - 0.866i$	1	0
$p_3^0 = 1.4142i$	0	1
$p_4^0 = -1.4142i$	0	1

TABLE 3.5: Table for computing maximum cancelled poles by choosing N.

Step 3.6.3. *To solve the following optimisation problem*

$$\begin{aligned} &\max\ k \\ &s.t.,\ \sum_{h=1}^{k} T_{i_h,j} \leq 2,\ \forall j = 1,2. \\ &\{i_1,\ldots,i_k\} \subseteq \{1,\ldots,l\}. \end{aligned}$$

We know that the optimal solution is $k = 4$ *and* $\{i_1,\ldots,i_k\} = \{1,2,3,4\}$ *and*

$$N(s) = diag\left[\frac{s^2+s+1}{s-4},\ \frac{s^2+2}{s-4}\right].$$

Step 3.6.4. *We finally obtain*

$$D^* = sI - sN = diag\left[sI - \frac{s^2+s+1}{s-4},\ sI - \frac{s^2+2}{s-4}\right] = diag\left[\frac{-5s+1}{s-4},\ \frac{-4s+2}{s-4}\right].$$

3.7 Structure-preserving model reduction

In previous Sections and in [103] an efficient method is developed to reconstruct networks in the presence of noise and nonlinearities. This method relies on the assumption that the conditions for network reconstruction presented in [34] have been met. In our approach, we use the same information as traditional system identification methods, i.e. input-output data. However, time-series data can be used to reconstruct the dynamical network structure of the system. Once the dynamical structure function is obtained, an algorithm for constructing a minimal order state-space representation consistent with such function is developed. In an application, this provides a way to estimate the complexity of the system by determining the minimal number of hidden states in the system. However, when the dynamical structure function is obtained by experiment data, the order of obtained dynamical structure function might possibly have a higher McMillan degree than the true one due to the noisy and nonlinear effect. We shall need to use model

reduction to cancel some noisy effect or to have a simpler system for further analysis.

Problem 3.7.1. *Given a stable system G with dynamical structure function $(Q,\ P)$ and structural degree[6] n, and a non-negative integer $\tilde{n} < n$, find an approximate system $\tilde{G}$ with dynamical structure function $(\tilde{Q},\tilde{P})$ and structural degree $\tilde{n}$ such that*

1. $\mathscr{B}(\tilde{Q},\tilde{P}) = \mathscr{B}(Q,P)$*, and*
2. $\tilde{G}(s) = \arg\min \|G - \tilde{G}\|_\infty$.

Note that even without the structural requirements, the optimisation problem given by the second point is well known to be non-convex and hard. In this study we aim to propose a solution to this problem. Noting from eq. (3.5) and eq. (3.6), we have

$$W = (sI - D)Q + D \text{ and } V = (sI - D)P,$$

since D is a diagonal matrix, the multiplication of $(sI-D)$ would not change the Boolean structure of Q ,P. In consequence,

$$\mathscr{B}(W) = \mathscr{B}(Q) \oplus \mathscr{B}(D) \text{ and } \mathscr{B}(V) = \mathscr{B}(P).$$

We can then transform the Problem 3.7.1 into the following problem:

Problem 3.7.2. *Given a stable system G with dynamical structure function $(Q,\ P)$ and structural degree n and known corresponding $(W,\ V)$, and a non-negative integer $\tilde{n} < n$, find an approximate system $\tilde{G}$ with $(\tilde{W},\tilde{V})$ and structural degree $\tilde{n}$ such that*

1. $\mathscr{B}(\tilde{W},\tilde{V}) = \mathscr{B}(W,V)$*, and*
2. $\tilde{G}(s) = \arg\min \|G - \tilde{G}\|_\infty$.

For any stable system $G_1,\ G_2$ with the same V

$$\|G_1 - G_2\|_\infty = \|(sI - W_1)^{-1}(W_1 - W_2)(sI - W_2)^{-1}V\|_\infty = \|(sI - W_1)^{-1}(W_1 - W_2)G_2\|_\infty \quad (3.55)$$

An intuitive but naive idea is to do the term-wise model reduction on W, this will largely affect the dynamics of the system, i.e. G [75]. We shall then develop new theory for zero-pattern preserving model reduction or transform it to a problem that is solvable.

[6] the order of minimal state-space realisations that consistent with $(Q,\ P)$

3.7.1 Model reduction through Gilbert realisation

Problem 3.7.3. *Given a stable transfer matrix $G(s) \in \mathscr{RH}_\infty$ with certain Boolean structure $\mathscr{B}(G)$ and given a pre-defined small constant $\epsilon \in \mathbb{R}^+$, we are interested in find any low-order stable approximation that is "close" in terms of infinite norm, i.e.*

$$\|G - \hat{G}\|_\infty \leq \epsilon, \tag{3.56}$$

and the zero pattern in G perserves, i.e.

$$\mathscr{B}(G) = \mathscr{B}(\hat{G}). \tag{3.57}$$

Solution: Let us start from the Gilbert realisation of G [107] gives

$$G(s) = \sum_{i=1}^{l} \frac{K_i}{s - \lambda_i} + G(\infty) \tag{3.58}$$

where $K_i = \lim_{s\to\lambda_i}(s-\lambda_i)G(s)$ and has rank 1 since we are assuming that $G(s)$ has simple real poles. Consider a matrix decomposition of K_i in the following form:

$$K_i = E_i F_i, \ \forall i, \tag{3.59}$$

where $E_i \in \mathbb{R}^p$ and $F_i = (E_i^T E_i)^{-1} E_i^T K_i$. Then $A = \text{diag}\{\lambda_i\} \in \mathbb{R}^{l\times l}$, $B = \begin{bmatrix} F_1^T & F_2^T & \dots & F_l^T \end{bmatrix}^T$, $C = \begin{bmatrix} E_1 & E_2 & \dots & E_l \end{bmatrix}$ and $D = G(\infty)$.

Proposition 3.7.1. *For any $i = 1,\dots,l$, if there exists a subset Γ_i of $\{1,\dots,l\}/\{i\}$, such that $\mathscr{B}(K_i) \oplus \left(\oplus_{j\in\Gamma_i}\mathscr{B}(K_j)\right) = \mathscr{B}(K_i)$, then we call the eigenvalue λ_i in eq.* (3.58) *reducible mode and set*

$$\Lambda = \{i | \lambda_i \text{ is reducible mode}. \ \forall i\}.$$

Let $\hat{G} = G - \frac{K_i}{s+\lambda_i}$, then $\mathscr{B}(\hat{G}) = \mathscr{B}(G)$.

Proof. From the definition of Gilbert realisation, a truncation of terms in eq. (3.58) leads to a one-order lower realisation of G. We then characterise the gap between $\hat{G}$ and G. By direct calculation

$$\|G - \hat{G}\|_\infty = \left\| \frac{K_i}{s+\lambda_i} \right\|_\infty = \frac{\text{trace}(K_i)}{|\lambda_i|},$$

the last equality holds because knowing that (K_i is a rank-1 matrix)

$$\frac{\sigma_{max}(K_i)}{|\lambda_i|} = \frac{\lambda_{max}(K_i^T K_i)}{|\lambda_i|} = \frac{\text{trace}(K_i)}{|\lambda_i|}.$$

When $\frac{\text{trace}(K_i)}{|\lambda_i|} \leq \epsilon$, then the Problem 3.7.3 is solvable; we can construct $\hat{G}$ satisfying all three requirements in Problem 3.7.3. □

Remark 3.7.1. *This method here only provides a sufficient solution to Problem 3.7.3.*

3.7.2 Balanced truncation on hidden variables

This section will propose a solution to Problem 3.7.2 by mathematically transforming it to a solvable problem.

Problem 3.7.4. *Given a stable system G with dynamical structure function (W,V) and structural degree n, and a non-negative integer $\tilde{n} < n$, find an approximate system $\tilde{G}$ with dynamical structure function $(\tilde{W},\tilde{V})$ and structural degree $\tilde{n}$, for any prescribed positive real number ϵ, we are searching for $\tilde{G}$ such that*

1. $\mathscr{B}(\tilde{W},\tilde{V}) = \mathscr{B}(W,V)$, *and*
2. $\|I - \tilde{G}^{-1}G\|_\infty \leq \epsilon$.

Assumption 3.7.1. *Assume that the principle sub-matrix of A, A_{22} is stable.*

A minimal realisation of $[W \;\; V]$ can be obtained as follows:

$$[W \;\; V] = [A_{11} \;\; B_1] + A_{12}(sI - A_{22})^{-1}[A_{21} \;\; B_2] \tag{3.60}$$

Therefore we consider a "virtual" system from the above realisation

$$\begin{aligned} \dot{z} &= A_{22}z + A_{21}y \\ h &= A_{12}z + A_{11}y, \end{aligned} \tag{3.61}$$

where z is the hidden states and y is the measured states in eq. (3.1).

Lemma 3.7.1. *Consider a linear transformation mapping $(A,B,[I\ 0])$ to $(T^{-1}AT, T^{-1}B, [I\ 0]T)$ by selecting*

$$T = \begin{bmatrix} I & 0 \\ 0 & T_2^{-1} \end{bmatrix},$$

for any nonsingular matrix T_2. The transformed system is

$$\left(\begin{bmatrix} A_{11} & T_2A_{12} \\ A_{21}T_2^{-1} & T_2A_{22}T_2^{-1} \end{bmatrix}, \begin{bmatrix} B_1 \\ T_2B_2 \end{bmatrix}, [I\ 0]\right).$$

Such linear transformations do not change $(W,\ V)$ and G.

Proof. This can be shown from eq. (3.60). □

Remark 3.7.2. *Lemma 3.7.1 states the fundamental limitation of network reconstruction problem: one can not obtain the "true" connections between hidden and measured states and connections between hidden states themselves, which is natural.*

Since A_{22} is assumed to be stable from Assumption 3.7.1 and (A_{22},A_{12}) is controllable and (A_{21},A_{22}) is observable, then we can compute the unique solution [107] for Controllability Gramians Σ_c and Observability Gramiams Σ_o satisfying

$$A_{22}\Sigma_c + \Sigma_c A_{22}^* + A_{12}A_{12}^* = 0 \tag{3.62}$$

$$A_{22}^*\Sigma_o + \Sigma_o A_{22} + A_{21}^*A_{21} = 0. \tag{3.63}$$

A physical interpretation of such Gramians can be found in [85, 75]. Essential properties of Gramians are as follows, by changing the coordinate of the hidden states z, $z' = Uz$, the corresponding new Gramians are

$$\Sigma_c' = U^{-1}\Sigma_c U^{-T} \text{ and } \Sigma_o' = U^T\Sigma_o U.$$

Notice that $\Sigma_o, \Sigma_c > 0$, so we can select U such that $\Sigma_c' = \Sigma_0' = (\Sigma_c\Sigma_0)^{1/2}$, we call the system after such transform *balanced*.

Lemma 3.7.2. *Suppose W, A_{22} and A are defined as in eq.* (3.60)*, if A is Hurwitz, then $(sI-W)^{-1}$ is a stable transfer function.*

Proof. Since $sI-W$ is the Schur complement of $sI-A_{22}$ in $sI-A$, then

$$\det(sI-W)^{-1} = \frac{\det(sI-A_{22})}{\det(sI-A)}. \tag{3.64}$$

From the above equation, since A is Hurwitz, then $(sI-W)^{-1}$ is a stable transfer function. □

Remark 3.7.3. *Frequency-weighted model reduction can only be applied when the frequency weight is stable.*

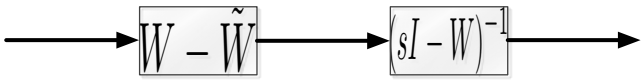


FIGURE 3.12: Cascade interconnected systems.

Since

$$\|I - G\tilde{G}^{-1}\|_\infty = \|I - (sI - W)^{-1}(sI - \tilde{W})\|_\infty = \|(sI - W)^{-1}(W - \tilde{W})\|_\infty,$$

we can view $(sI - \tilde{W})^{-1}(W - \tilde{W})$ as a cascade system of a known system $(sI - W)^{-1} \in \mathscr{RH}_\infty$ and a to-be-determined system $(W - \tilde{W})$ and form it as a frequency weighted model reduction problem.

Assume the corresponding state-space realisation of $(sI - W)^{-1} = C'(sI - A')^{-1}B'$, and recall that $W = A_{11} + A_{21}(sI - A_{22})^{-1}A_{12}$, then the cascaded system can be written in the form

$$H(z) = \left[\begin{array}{cc|c} A_{22} & 0 & A_{21} \\ B'A_{12} & A' & B'A_{11} \\ \hline 0 & C' & 0 \end{array}\right] \triangleq \left[\begin{array}{c|c} \bar{A} & \bar{B} \\ \hline \bar{C} & 0 \end{array}\right] \tag{3.65}$$

Let $\bar{\Sigma_c}, \bar{\Sigma_o}$ be solutions to the following Lyapunov equations

$$\bar{A}\bar{\Sigma_c} + \bar{\Sigma_c}\bar{A}^* + \bar{B}\bar{B}^* = 0 \tag{3.66}$$

$$\bar{\Sigma_o}\bar{A} + \bar{A}^*\bar{\Sigma_o} + \bar{C}^*\bar{C} = 0. \tag{3.67}$$

Then the input weighted Gramian

$$\Sigma_c = \begin{bmatrix} I & 0 \end{bmatrix} \bar{\Sigma_c} \begin{bmatrix} I \\ 0 \end{bmatrix} \text{ and } \Sigma_o = \begin{bmatrix} I & 0 \end{bmatrix} \bar{\Sigma_o} \begin{bmatrix} I \\ 0 \end{bmatrix}.$$

Assuming that $\bar{\Sigma_c}, \bar{\Sigma_o}$ have the following general forms

$$\bar{\Sigma_c} = \begin{bmatrix} \Sigma_c & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \text{ and } \bar{\Sigma_c} = \begin{bmatrix} \Sigma_o & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}.$$

Definition 3.7.1. *A realisation is* hidden balanced *if its controllability and observability Gramians with corresponding block structure partition,*

$$X_c = \begin{bmatrix} X_{c11} & X_{c12} \\ X_{c21} & X_{c22} \end{bmatrix}, \quad Y_o = \begin{bmatrix} Y_{o11} & Y_{o12} \\ Y_{o21} & Y_{o22} \end{bmatrix},$$

satisfy $X_{c11} = Y_{o11} = \Sigma \triangleq \Sigma_c \Sigma_o$*, were* $\Sigma > 0$ *is diagonal.*

We now transform the unsolvable problem to a standard frequency-weighted model reduction problem, which can be approached using methods in [107]. We omit the rest routine procedures here but only list Algorithm 2 for unweighted truncations on hidden state. We shall finalise this idea and expand this section with more details and examples.

3.8 Conclusion

In this Chapter, we aim to introduce/define dynamical structure functions to solve the network reconstruction problem.

We firstly proposed a new network reconstruction method in the presence of noise and nonlinearities based on dynamical structure functions. The key idea is to find minimal distances between existing data and the data required to obtain particular Boolean networks. The method was illustrated with two biologically-oriented examples. They showed that even in the presence of nonlinearities and considerable noise network reconstruction was possible. Eventually, when the signal to noise ratio was too small, reconstruction was no longer possible, and that is true irrespective of the method used.

We have also presented a method for obtaining a minimal order realisation consistent with a given dynamical structure function. We showed that the minimal order realisation of a given dynamical structure function can be achieved by choosing a constant diagonal matrix D. This provides a way to estimate the complexity of the system by determining the minimum number of hidden states that needs to be considered in the reconstructed network. For example, in the context of reconstruction of biological networks from data, it helps understand the number of unmeasured molecules in a particular pathway.

Obviously, the proposed network reconstruction method has limitations with respect to nonlinearities. With stronger nonlinear terms the method eventually fails. Currently we are investigating a method that deals with nonlinearities. For example, network reconstruction for oscillatory systems is still an open problem. However, when applied to the reconstruction of various equilibrium point models given in the literature, we observed that reconstruction was always possible when the signal-to-noise ratio of the measured data was not too small (far less than 1).

Ongoing research on network reconstruction:

Time-reversal anti-causal system identification: when the experiments are conducted as the way indicated in the report, we found that the network structure information is encoded in the inverse of transfer function, i.e. $X = G^{-1}$. Recall that a very small

Algorithm 8 Direct balanced truncation on realisation of $[W,\ V]$

Step 3.7.1. *Find a balanced realisation [107] of A through linear transformation U, from Lemma 3.7.1, we can see that it has the same* $(Q,\ P)$ *and G as the original system before transformation.*

$$\tilde{G} = \left(\begin{bmatrix} A_{11} & UA_{12} \\ A_{21}U^{-1} & UA_{22}U^{-1} \end{bmatrix} \triangleq \tilde{A}, [I\ 0]^T, [I\ 0] \right).$$

(sub-algorithm to find U is similar to balance realisation in [107] by solving Lyapunov equations)

Step 3.7.2. *We can re-partition* $\tilde{A}$ *to*

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} \\ \tilde{A}_{21} & \tilde{A}_{22} & \tilde{A}_{23} \\ \tilde{A}_{31} & \tilde{A}_{32} & \tilde{A}_{33} \end{bmatrix},$$

where

$$\tilde{A}_{11} = A_{11}, [\tilde{A}_{12}\ \tilde{A}_{13}] = UA_{12},$$
$$\begin{bmatrix} \tilde{A}_{22} & \tilde{A}_{23} \\ \tilde{A}_{32} & \tilde{A}_{33} \end{bmatrix} = UA_{22}U^{-1}, \begin{bmatrix} \tilde{A}_{21} \\ \tilde{A}_{31} \end{bmatrix} = A_{21}U^{-1}.$$

Step 3.7.3. *Here we need to solve Lyapunov equations in eq.* (3.66) *and* (3.67) *[107] and find the corresponding* $\Sigma = diag(\sigma_{s_1} I_{s_1},\ \sigma_{s_k} I_{s_k})$, *where* $\sigma_{s_1} > \ldots > \sigma_{s_k}$. *If* $\sigma_{s_k} \leq \epsilon$ *or* $\sigma_{s_k} << \sigma_{s_{k-1}}$, *then we can do balanced truncation on* $\tilde{G} = \left(\tilde{A}, [0\ I]^T, [0\ I]\right)$ *to have*

$$\hat{G} = \left(\hat{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} \in \mathbb{R}^{n \times n},\ B = \begin{bmatrix} I_p \\ 0 \end{bmatrix}, C = \begin{bmatrix} I_p & 0 \end{bmatrix} \right).$$

Step 3.7.4. *By choosing*

$$\begin{aligned} \hat{A} &= \left(\begin{bmatrix} I & 0 \\ 0 & U^{-1} \end{bmatrix} A \begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix} \right) [1:n-1, 1:n-1] \\ &= \begin{bmatrix} I & 0 \\ 0 & U^{-1}[1:t, 1:(t+1)] \end{bmatrix} A \begin{bmatrix} I & 0 \\ 0 & U[1:(t+1), 1:t] \end{bmatrix}, \end{aligned}$$

where $t = n - p - 1$, $A[1:t, 1:t]$ *for any matrix A denotes the submatrix of A defined by the rows* 1 *to t and columns* 1 *to t.*

perturbation might lead to totally different network structure, i.e. given any positive real number $\epsilon \in \mathscr{R}^+$, there always exists $\|\Delta\| \leq \epsilon$, such that

$$\mathscr{B}(X) \neq \mathscr{B}(G + \Delta)^{-1}.$$

Therefore we wish to use direct optimisation to obtain X from input-output data in Section 3.4.2 to avoid the uncertainty propagation in the inverse operator. However, the inverse of a transfer matrix might be a non-causal system and thus hard to analyse. Georgiou and Smith in a recent paper [28] pointed out that the role of time-asymmetry of stability plays in feedback control. Currently, a reversed time-axis system identification algorithm is under investigation to tackle this difficulty.

Integrate Bayesian analysis: the machine learning community has strong arability of handling noisy and missing data and there are a number of sophisticated tools in this field. Our further analysis can be done to embed such algorithms, e.g. [69] into our framework[7].

[7]Some preliminary research has been conducted in [66].

3.9 Appendix

3.9.1 Proof of the claim in Step 2 of the proposed algorithm:

Proof. Using results from [19], if a pole of $[sQ \;\; sP]$, say λ_i, is cancelled by $N = (sI - D)s^{-1}$, then the realisation of the cascade $(sI - D)s^{-1}[sQ \;\; sP]$ loses observability. In this case, it follows that there exists a nonzero vector $w = [w_1^T, w_2^T]^T$ such that

$$\begin{bmatrix} A_1 - \lambda_i I & 0 \\ B_2 C_1 & A_2 - \lambda_i I \\ D_2 C_1 & C_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 0.$$

The first equation shows that w_1 is an eigenvector of A_1 corresponding to λ_i. Since A_1 is diagonal, $w_1 = \begin{bmatrix} 0 & \dots & 0 & 1_{i^{th}} & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{1\times l}$. Therefore we have

$$\begin{bmatrix} A_2 - \lambda_i I & B_2 \\ C_2 & D_2 \end{bmatrix} \begin{bmatrix} w_2 \\ C_1 v_i \end{bmatrix} = 0.$$

Noticing that $C_1 w_i = E_i$ and that

$$\begin{bmatrix} I & 0 \\ -C_2(A_2 - sI)^{-1} & I \end{bmatrix} \begin{bmatrix} A_2 - sI & B_2 \\ C_2 & D_2 \end{bmatrix} = \begin{bmatrix} A_2 - sI & B_2 \\ 0 & N \end{bmatrix}, \tag{3.68}$$

we therefore obtain

$$N(\lambda_i)E_i = 0. \tag{3.69}$$

In summary, designing D to cancel any pole λ_i of $[sQ \;\; sP]$ is equivalent to imposing that eq. (3.69) holds. The next question is: given $[sQ \;\; sP]$ what is the maximal number of poles that can be cancelled by N, i.e. what is the largest number of poles for which eq. (3.69) is satisfied? To answer this, recall that to cancel a pole λ_i, eq. (3.69) must be satisfied. Furthermore, $E_i[j]$ being nonzero, for some j, implies that there exists at least one nonzero element in the j^{th} row of E_i. In this case, satisfying eq. (3.69) imposes that the j^{th} diagonal element of $N(\lambda_i)$ is 0, i.e. the j^{th} diagonal element of D is λ_i. In other words, a nonzero element in E_i corresponds to a fixed value in the corresponding diagonal position in D. Since D is a constant diagonal matrix then any pair of orthogonal vectors in $\{\mathscr{B}(E_1), \dots, \mathscr{B}(E_l)\}$ do not intervene in the choice of an element on the diagonal of D. □

3.9.2 Algorithm to find ϕ and Φ:

As is presented in [33], an undirected graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, \ldots, v_l\}$ is the set of nodes and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges.

For our purposes, we construct an undirected graph $\mathcal{G}_a$ using the following rules:

- A node is associated with each vector in the set $\{\mathcal{B}(E_1), \ldots, \mathcal{B}(E_l)\}$. There are thus l nodes in the considered graph.
- An undirected edge (i, j) is drawn between node i and node j if the equality $\mathcal{B}(E_i)^T \mathcal{B}(E_j) = 0$ is satisfied.

It is easy to see that the maximum cardinality of the set Φ corresponds to the maximum number of nodes in a complete subgraph K_n of the graph $\mathcal{G}_a$.

Although the problem of finding a largest complete subgraph in an undirected graph is a NP-hard problem, methods to this have been proposed in computer science and some corresponding MATLAB code can be downloaded[8]. Therefore we can use these methods to obtain a largest complete subgraph and consequently compute the corresponding set Φ and its corresponding cardinality ϕ.

[8] http://www.mathworks.com/matlabcentral/fileexchange/19889.

Chapter 4

The Path Ahead

The conclusions and future work related to this thesis are included in Chapter 2 and Chapter 3 respectively, and I would like to take this opportunity to reflect on research opportunities in the field more widely.

I studied control theory at the time when people thought that control theory is mature. Prof. Y. C. Ho, a Control Field Award winner in his sequential blogs[1] claimed that "control is dead" based on the fact that NSF would not support any research on classic control theories.

Hence, the control and systems societies are trying to find new directions: Professor Richard Murray from Caltech summarised some findings and recommendations of a panel on "Future Directions in Control, Dynamics, and Systems" in 2003. A set of grand challenges that illustrate some of the recommendations and opportunities are also provided in [55]. A conference was held in LIDS, MIT in 2009 discussing the "path ahead"[2]. Thus, control as a field is clearly not dead, it just needs to refocus its attention in "non-classical" applications of extreme importance such as systems and synthetic biology, power systems, internet, networks, etc.. For the future directions, here are some of my thoughts:

4.1 Solving existing open problems

There are a large number of interesting open problems in control theory [9, 8]. Prof. David Hilbert proposed 23 unsolved problems known as "the Hilbert's problems". These problem were very influential for 20^{th} century mathematics. For the similar reason, I

[1] http://blog.sciencenet.cn/home.php?mod=space&uid=1565&do=blog&id=329153, http://blog.sciencenet.cn/home.php?mod=space&uid=1565&do=blog&id=344686

[2] http://paths.lids.mit.edu/

believe the solutions to some of the open problems in [9, 8] will be influential and will transform the field.

4.2 Motivation from promising applications

Like robust control theory [107] motivated by the aerospace engineering in the 1990s, nowadays there are a number of interesting applications of control theory, e.g., biology and power systems. This application-driven research may lead to ask a large number of fundamental problems that cannot be solved with current available methods. This motivates the development of new theory. In this thesis we have identified two such problems and we have provided answers for them.

4.3 Encounter with other techniques

In Professor Lennart Ljung's plenary talk at IEEE Conference on Decision and Control 2011, he mentioned two new encounters (sparsity and compressive sensing, and machine learning) where system identification meets and tries to absorb the essence of new techniques to push the identification methodology forward. This coincides with what I mentioned above. To keep the control theory young and dynamic, we need to encounter advanced techniques in other fields, e.g., information theory, machine learning. I shall briefly list some of my ideas as simple first attempts.

4.3.1 Encounter with information theory

Here, more specifically, we shall not only focus on sparsity and compressive sensing but also point the readers to a paper by Lestas, Vinnicombe and Paulsson [49], which studied the fundamental limitations given the information channel.

Before going through the technical details of compressive sensing, what it can bring to our study of network reconstruction problem is a relaxation in the experimental requirements in Corollary 3.2.1 in Chapter 3 by knowing the network is a-priori sparse. It also can provide possibility for network reconstruction with nonlinear dynamics [66].

Mathematically, a signal x is *k-sparse* when it has at most k nonzeros, i.e. $\|x\|_0 \leq k$. We let

$$\sum\nolimits_k = \{x : \|x\|_0 \leq k\}$$

denote the set of all k-sparse signals. Typically, we shall be dealing with signals that are not themselves sparse, but which admit a sparse representation in some basis Ψ. In

this case we shall still refer to x as being k-sparse, with the understanding that we can express x as $x = \Psi c$ where $\|c\|_0 \leq k$.

Compressive sensing is a technique that encodes a signal x of dimension N by computing a measurement vector y of dimension of $m \ll N$ via linear projections, i.e.

$$y = \Phi x,$$

where $\Phi \in \mathbb{R}^{m \times N}$ is referred to as the *measurement matrix*. In general, it is not possible to uniquely recover the unknown signal x using measurements y with reduced-dimensionality. Nevertheless, if the input signal is sufficiently sparse, exact reconstruction is possible. In this context, suppose that the unknown signal $x \in \mathbb{R}^N$ is at most K-sparse, i.e. that there are at most K nonzero entries in x.

A naive reconstruction method is to search among all possible signals and find the sparsest on which is consistent the linear measurements. This requires only $m = 2K$ random linear measurements, but finding the sparsest signal representation is an NP-hard problem. On the other hand, [13] demonstrated that reconstruction of x from y is a *polynomial time* problem if more measurements are taken. This is achieved by casting the reconstruction problem as an l_1-minimisation problem, i.e.

$$\min \|x\|_1 \text{ subject to } y = \Phi x, \tag{4.1}$$

where $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ denotes the l_1-norm of the vector x. It is a convex optimisation problem and can be solved efficiently by linear programming (LP) techniques.

When Shannon meets Bode

This progress in information theory will definitely accelerate the development of control theory. In particular, beyond the implication of compressive sensing to network reconstruction, one may immediately ask the question: “how this affects the observability theory?”

In control theory observability is a measure for how well internal states of a system can be inferred by knowledge of its external outputs [107]. Recall that the proof to the observability is equivalent to solving linear equations, if we assume that the initial state x_0 is known a-priori sparse or admit a sparse representation in some basis Ψ, we can then relax the requirement for observability using the l_1 decoding algorithm above.

More specifically, assume that for standard DTLTI system,

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k,$$

where $x \in \mathscr{R}^N$. Without loss of generality, we assume $C \in \mathscr{R}^{1\times N}$. It then follows

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} x_0. \tag{4.2}$$

There are two scenarios, by which the compressive sensing technique might be useful. First, when the system is not observable but with a-priori information that the initial state is sparse. Then the solution of x_0 in eq. (4.2) can be found using the l_1 optimisation in eq. (4.1). Second, if N is very large, the number of observations, i.e. $\{y_i\}$ is less than N but satisfies the condition that the number of observations is large than $2 \times$ Sparsity of x_0, then the solution of x_0 in eq. (4.2) again can be found using the l_1 optimisation in eq. (4.1).

These ideas can be used to re-think about the relaxation in the observability assumption in control theory when there exists a-priori knowledge about sparsity of the considered signal or system, e.g., in model-predictive control [50], fault-detection theory [105], etc..

4.3.2 Encounter with machine learning

Machine learning is a discipline concerned with the design and development of algorithms that allow computers to evolve behaviours based on empirical data [70, 6].

In control theory, system identification methods are based on the fact that when the number of samples in data goes to infinity, such method will be unbiased. However, there is no such guarantee for finite samples, i.e., given the amount of data and the prior knowledge of the system, the results obtained by system identification may not be the optimal solution. And normally, the identification and control processes are separated. It is easy to understand that the use of identification is for control, we shall emphasis here that control is also for identification. It would be more natural to think in the machine learning way that the data is used to generate a posterior knowledge of the model. Once such model is obtained, we may then ask: How to develop algorithms that

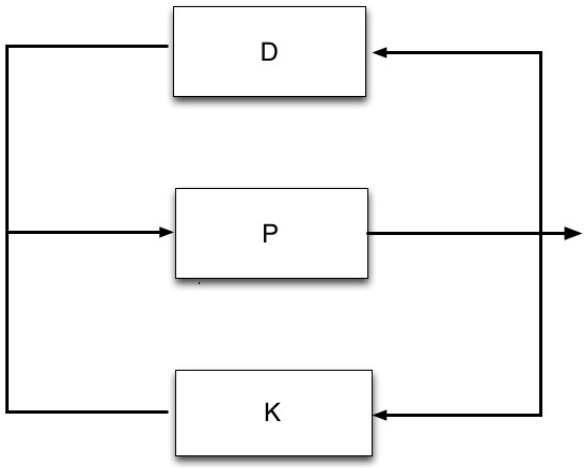


FIGURE 4.1: Machine control diagram.

allow computers to control?

Machine control?

This section illustrates the proposed machine control problem using the control diagram in Fig 4.1. In Fig. 4.1, P is a nominal plant where its dynamics is known, D is the uncertainty in the plant and K is the controller to be designed.

As a simple example of illustration, let us consider the simplest case, a SISO transfer function G as the model of P in Fig. 4.1. The unmodelled dynamics (uncertainty) D (might be structured or unstructured) in Fig. 4.1 has some probabilistic distribution, e.g., uniform or Gaussian distribution if we have some knowledge about D. In the machine learning literature, there are lots of results that can generate a posterior distribution of D based on the learning of model from empirical data.

If D is bounded, then we can directly apply Small Gain Theorem [22] to design a stabilised controller K. However, this does not take the prior distribution into account. The system is stable but not "optimal".

If the distribution of D is unbounded and not symmetric with respect to 0, the Small Gain Theorem [22] does not hold. And of course, there is no *absolute* stability for such system for unbounded distribution D. A question to ask is that given the plant P, a positive real value $p \in (0,1)$ and a distribution on D, how to design controller K such that the whole system is stable with probability p. Secondly, even if we adopt the probabilistic version of Small Gain Theorem as just mentioned, it is too conservative. How to design controller using the information on the posterior distribution in the model?

In general, how to provide a practical machine control approach to tackle such questions for general systems remains open. It is currently under investigation using random matrix theory, e.g. [21].

4.4 Ending

“Oh! There are still other **128** lines in ‘Auguries of Innocence‘!”

References

[1] U. Alon, *An Introduction to Systems Biology - Design Principles of Biological Circuits,* Chapman & Hall/CRC 2007.

[2] A. Barabasi and R. Albert, "Emergence of scaling in random networks," Science, 1999.

[3] M. Barahona and L. Pecora, "Synchronization in small-world systems," Physical Review Letters, 2002.

[4] B. Palsson, *Systems Biology: Properties of Reconstructed Networks,* Cambridge University Press, 2006.

[5] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods,* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[6] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[7] V. Blondel, J. Hendrickx and J. Tsitsiklis, "Continuous-time average-preserving opinion dynamics with opinion-dependent communications," SIAM Journal on Control and Optimization, 2010.

[8] V. Blondel and A. Megretski (Eds), *Unsolved Problems in Mathematical Systems and Control Theory,* Princeton University Press, 2004.

[9] V. Blondel, E. Sontag, M. Vidyasagar and J. Willems (Eds),*Open Problems in Mathematical Systems and Control Theory,* Springer Verlag, Heidelberg, 1999.

[10] D. Boley, F. Luk and D. Vandevoorde, "Vandermonde factorization of a Hankel matrix," Scientific Computing, 1997.

[11] S. Brin, L. Page, "The anatomy of a large-scale hyper textual Web search engine," Computer Networks and ISDN Systems, 1998.

[12] K. Burnham and D. Anderson, *Model Selection and Inference - A Practical Information-Theoretic Approach*.Springer-Verlag, 1998.

[13] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," IEEE Transactions on Information Theory, 2006.

[14] I. Cantone, L. Marucci, F. Iorio, M. Rucci, V. Belcastro, M. Bansal, S. Santini, M. Bernardo, D. Bernardo and M. Cosma, "A yeast synthetic network for *in vivo* assessment of reverse-engineering and modeling approaches," Cell, 2009.

[15] V. Cerone, D. Piga, and D. Regruto, "Bounding the parameters of linear systems with stability constraints," In Proceedings of American Control Conference, 2010.

[16] T. Chen, H. Ohlsson and L. Ljung "On the estimation of transfer functions, regularizations and Gaussian processes - Revisited," to appear, Automatica.

[17] J. Cortes, "Finite-time convergent gradient flows with applications to network consensus," Automatica, 2006.

[18] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," Journal of Parallel and Distributed Computing, 1989.

[19] M. Dahleh, *Lectures on Dynamic Systems and Control,* MIT open course.

[20] M. Dahleh and J. Pearson, "L1-optimal feedback controllers for MIMO discrete-time systems," IEEE Transactions on Automatic Control, 1987.

[21] P. Deift, *Orthogonal Polynomials and Random Matrices: a Riemann-Hilbert approach,* Volume 3 of Courant Lecture Notes in Mathematics, New York University, Courant Institute of Mathematical Sciences, New York 1999.

[22] J. Doyle, B. Francis and A. Tannenbaum, *Feedback control theory,* Macmillan, 1992.

[23] M. Egerstedt, "Controllability of networked system," in Proceedings of International Symposium on Mathematical Theory of Networks and Systems 2010.

[24] P. Erdos and A. Renyi, "On random graphs," Public Maths,1959.

[25] R. Freeman, P. Yang and K. Lynch. "Stability and convergence properties of dynamic average consensus estimators," in Proceedings of the IEEE Conference on Decision and Control, 2006.

[26] T. Gardner, D. di Bernardo, D. Lorenz and J. Collins, "Inferring genetic networks and identifying compound mode of action via expression profiling," Science, 2003.

[27] M. George, "B-A scale-free network generation and visualization," Matlab Central, 2007

[28] T. Georgiou and M. Smith, "Feedback control and the arrow of time," International Journal of Control, 2010.

[29] E. Gilbert, "Random graphs," Annual Mathematical Statistics, 1959.

[30] E. Gilbert, "Controllability and observability in multivariable control systems," Journal of SIAM Control Series A, 1963.

[31] K. Glover and L. Silverman, "Characterisation of structural controllability," IEEE Transactions on Automatic Control, 1976.

[32] E. Gluskin, "Let us teach this generalization of the final-value theorem," European Journal of Physics, 2003.

[33] C. Godsil and G. Royal, *Algebraic Graph Theory*, New York: Springer-Verlag, 2001.

[34] J. Goncalves and S. Warnick, "Necessary and sufficient conditions for dynamical structure reconstruction of LTI networks," IEEE Transactions on Automatic Control, 2008.

[35] R. Gray, "Toeplitz and circulant matrices: A review," Now Publisher, 2006.

[36] E. Herrero, E. Kolmos, N. Bujdoso, Y. Yuan, M. Wang, M. Berns, G. Coupland, R. Saini, M. Jaskolski, A. Webb, J. Goncalves and S. Davis, "Effector binding to a co-repressor complex sustains the plant circadian oscillator," the Plant Cell, 2012.

[37] A. Hirotugu, "A new look at the statistical model identification," IEEE Transactions on Automatic Control, 1974.

[38] M. Hitz, "On computing the nearest singular Hankel matrices," in Proceedings of the International Symposium on Symbolic and Algebraic Computation, 2005.

[39] Y. Ho, "Review of the Witsenhausen problem," in Proceedings of the IEEE Conference on Decision and Control, 2008.

[40] B. Ho and R. Kalman, "Effective construction of linear state-variable models from input/output functions," in Proceedings of the 3rd Annual Allerton Conference on Circuit and System Theory, 1965.

[41] L. Hogben, "Minimum rank problems," Linear Algebra and its Applications, 2010.

[42] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press 1999.

[43] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," IEEE Transactions on Automatic Control, 2003.

[44] A. Julius, M. Zavlanos, S. Boyd, and G. Pappas, "Genetic network identification using convex programming," IET Systems Biology, 2009.

[45] R. Kalman, "On minimal partial realizations of a linear input/output map," in Aspects of Network and System Theory (R.E. Kalman and N. DeClaris, eds.), New York: Holt, Rinehart and Winston, 1971.

[46] S. Kung, "A new identification and model reduction algorithm via singular value decomposition," in Proceedings of the 12th Asilomar Conference on Circuits, Systems and Computers, 1978.

[47] J. Leskovec and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2005.

[48] I. Lestas, "On the analysis of scalable heterogeneous networks," Ph.D. Thesis, University of Cambridge, 2006.

[49] I. Lestas, G. Vinnicombe and J. Paulsson, "Fundamental limits on the suppression of molecular fluctuations," Nature, 2010.

[50] J. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, 2002.

[51] L. Ljung, *System Identification–Theory for the User*, Prentice Hall, 1999.

[52] J. Makhoul, "Linear prediction: a tutorial," Proceedings of the IEEE, 1975.

[53] S. Martini, M. Egerstedt and A. Bicchi, "Controllability analysis of networked systems using relaxed equitable partitions," International Journal of Systems, Control and Communications, 2010.

[54] S. Mason and H. Zimmermann, *Electronic Circuits, Signals and Systems.* John Wiley and Sons, INC., 1968.

[55] R. M. Murray, "Future directions in control, dynamics, and systems: overview, grand challenges, and new courses," European Journal of Control, 2003.

[56] K. Moore and M. Dahleh, "L1-optimal deconvolution with stable predictor polynomials," in Proceedings of IEEE Conference on Decision and Control, 1990.

[57] M. Newman, "The Laplacian spectrum of graphs," Master thesis, University of Manitoba, Canada, 2000.

[58] N. O'Clery, Y. Yuan, G. Stan and M. Barahona, "Find an mEEP in polynomial time," in preparation.

[59] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in Proceedings of IEEE Conference on Decision and Control, 2007.

[60] R. Olfati-Saber. "Ultrafast consensus in small-world networks," in Proceedings of American Control Conference, 2005.

[61] R. Olfati-Saber, "Algebraic connectivity ratio of Ramanujan graphs," in Proceedings of American Control Conference, 2007.

[62] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," IEEE Transactions on Automatic Control, 2004.

[63] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging." SIAM Journal on Control and Optimization, 2009.

[64] A. Olshevsky and J. N. Tsitsiklis, "A lower bound on distributed averaging," in Proceedings of IEEE Conference on Decision and Control, 2010.

[65] L. Ou, Y. Yuan and R. M. Murray, "Parametric delay-margin maximisation of consensus network using local control scheme," in preparation.

[66] W. Pan, Y. Yuan, and G. Stan, "Reconstruction of arbitrary biochemical reaction networks: a compressive sensing approach," submitted, IEEE Conference on Decision and Control, 2012.

[67] J. Partington, *An Introduction to Hankel Operators,* Cambridge University Press, 1988.

[68] L. Pecora and M. Barahona, "Synchronization of oscillators in complex networks," Chaos and Complexity Letters, 2005.

[69] C. Rangel, J. Angus, Z. Ghahramani and D. Wild, "Modeling genetic regulatory networks using gene expression profiling and state-space models," Probabilistic Modeling in Bioinformatics and Medical Informatics, 2005.

[70] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, the MIT Press, 2006.

[71] W. Ren, "Consensus seeking in multi-vehicle systems with a time-varying reference state," in Proceedings of American Control Conference, 2007.

[72] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*, Springer, 2007.

[73] M. Roberts, E. August, A. Hamadeh, P. Maini, P. McSharry, J. Armitage and A. Papachristodoulou, "A model invalidation-based approach for elucidating biological signalling pathways, applied to the chemotaxis pathway in R. sphaeroides", BMC Systems Biology, 2009.

[74] S. Rump, "Structured perturbations part I: Normwise distances," SIAM Journal of Matrix Analysis, Application, 2003.

[75] H. Sandberg and R. Murray, "Model reduction of interconnected linear systems," Optimal Control, Applications and Methods, Special Issue on Directions, Applications, and Methods in Robust Control, 2009.

[76] L. Shi, Y. Yuan and J. Chen, "Finite horizon LQR control with limited controller-system communication," in revision, IEEE Transaction on Automatic Control, 2010.

[77] L. Silverman, "Realization of linear dynamical systems," IEEE Transactions on Automatic Control, 1971.

[78] C. Song, Z. Qu, N. Blumm, and A. Barabsi, "Limits of predictability in human mobility," Science, 2010.

[79] E. Sontag, "Network reconstruction based on steady-state data", Essays in Biochemistry, 2008.

[80] D. Spanos, R. Olfati-Saber and R. M. Murray, "Dynamic consensus for mobile networks," in Proceedings of IFAC World Congress, 2005.

[81] G. Stan and R. Sepulchre, "Analysis of interconnected oscillators by dissipativity theory", IEEE Transactions on Automatic Control, 2007.

[82] S. Sundaram and C. Hadjicostis, "Finite-time distributed consensus in graphs with time-invariant topologies," in Proceedings of American Control Conference, 2007.

[83] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part I: Fixed topology," in Proceedings of IEEE Conference of Decision and Control, 2003.

[84] A. Taylor and D. Higham, "CONTEST: A controllable test matrix toolbox for MATLAB,"ACM Transactions on Mathematical Software, 2009.

[85] A. Vadendorpe and P. VanDooren, "Model reduction of interconnected systems," Mathematics in Industry, 2008.

[86] E. van Dam, "Regular graphs with four eigenvalues," Linear Algebra Applications, 1995.

[87] E. van Dam, "Nonregular graphs with three eigenvalues," Journal of Combinatorial Theory, Series B, 1998.

[88] L. Wang and F. Xiao, "Finite-time consensus problems for networks of dynamic agents," IEEE Transactions on Automatic Control, 2010.

[89] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," Nature, 1998.

[90] H. Witsenhausen, "A counterexample in stochastic optimum control," SIAM Journal of Control, 1968.

[91] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," Systems and Control Letter, 2004.

[92] E. Yeung, J. Gonçalves, H. Sandberg, and S. Warnick. "Network structure preserving model reduction with weak a priori structural information," in Proceedings of the Conference on Decision and Control, 2009.

[93] E. Yeung, J. Gonçalves, H. Sandberg, and S. Warnick. "The meaning of structure in interconnected dynamic systems," IEEE Control Systems Magazine, Special Issue on Designing Controls for Modern Infrastructure Networks, 2011.

[94] N. Young, *An introduction to Hilbert space*. Cambridge university press, 1988.

[95] Y. Yuan, "Minimal-time final value theorem of unknown DT-LTI systems," M. Phil. thesis, Cambridge University, 2009.

[96] Y. Yuan and J.Gonçalves, "Minimal-time network reconstruction of DTLTI system," In Proceedings of 49th IEEE Conference on Decision and Control, 2010.

[97] Y. Yuan, J. Liu, R. M. Murray and J. Gonçalves, "Decentralised minimal-time dynamic consensus," in Proceedings of American Control Conference, 2012.

[98] Y. Yuan, G. Stan, M. Barahona, L. Shi and J. Gonçalves, "Minimal-time uncertain output final value of unknown DT-LTI systems with application to the decentralised network consensus problem," in Proceedings of International Symposium on Mathematical Theory of Networks and Systems, 2010.

[99] Y. Yuan, G. Stan, L. Shi and J. Gonçalves "Decentralized final value theorem for discrete-time LTI systems with application to minimal-time distributed consensus," in Proceedings of IEEE Conference on Decision and Control, 2009.

[100] Y. Yuan, G. Stan, L. Shi, M. Barahona and J. Gonçalves, "Decentralised minimal-time consensus," accepted, Automatica, 2012.

[101] Y. Yuan, G. Stan, S. Warnick and J. Gonçavles, "Minimal dynamical structure realisations with application to network reconstruction from data," in Proceedings of Conference on Decision and Control, 2009.

[102] Y. Yuan, G. Stan, S. Warnick and J. Gonçavles, "Minimal realisation of dynamical structure function," accepted, IEEE Transaction on Automatic Control.

[103] Y. Yuan, G. Stan, S. Warnick and J. Gonçavles, "Robust network reconstruction from data," to appear, Special issue on Systems Biology, Automatica, 2011.

[104] Y. Yuan, Z. Zhang, R. Murray and J. Gonçavles, "Decentralised consensus-value computation algorithm using imperfect observations", in preparation.

[105] Z. Zhang, "Fault detection and isolation for linear dynamic systems," Ph.D. Thesis, Imperial College London, 2011.

[106] H. Zhang, M. Chen, G. Stan, T. Zhou and J. Maciejowski, "Collective behavior coordination with predictive mechanisms," IEEE Circuits and Systems Magazine, 2008.

[107] K. Zhou, J. Doyle and K. Glover, *Robust and Optimal Control*, Prentice Hall, 1996.

[108] M. Zhu and S. Martinez, "On discrete-time dynamic average consensus," Automatica, 2010.