Disturbance Attenuation in Tree-Structured Interconnected Systems

Yaqi Zhu

Submitted in total fulfilment of the requirements of the degree of

Master of Philosophy

Department of Mechanical Engineering
THE UNIVERSITY OF MELBOURNE

May 2018

Abstract

Tree topologies are a subset of networked systems that can be used to represent many large-scale applications ranging from transportation networks to irrigation systems. One important aspect of network design for these systems is to certify a specific controller design with desired disturbance propagation characteristics.

Most existing literature cast desirable disturbance propagation characteristics as a stability property, which guarantees the boundedness of states rather than disturbance attenuation. Given the fact, however, that in most real-world systems the states are expected to stay within a specific set, it is highly desirable that the disturbance is attenuated when propagating through the network. The benefit is that the largest effect of the disturbance is always caused in the subsystem where it originally occurs, and state constraints can be satisfied for all subsystems as long as the origin subsystem is well controlled.

This research proposes computationally scalable sufficient conditions for tree topologies to provide guarantees of disturbance attenuation. Only neighbouring information is required by these conditions, which enables distributed controller design. Under these conditions, a centralised controller synthesis method is then developed for tree networks with controlled interconnections between subsystems. Distributed design of controllers is then presented by breaking up the centralised controller synthesis problem into sub-problems for subsystems.

Subsequently, a relaxation of the requirements of network topology to "tree-like" systems is proposed to widen the scope of the results. Specifically, an approach to approximate a system with an arbitrary topology by a tree is developed, facilitating the application of disturbance attenuation results in more general cases.

Finally, demonstrations via a water supply system and a traffic simulation are used to validate the theoretical findings for continuous-time and discrete-time systems, respectively. Results show

that the developed controller can effectively attenuate disturbance. Subsequently, the time efficiency of the proposed sufficient conditions are demonstrated. Moreover, the efficacy of the tree approximation method is shown by comparing various tree extractions.

Declaration

This is to certify that

1. the thesis comprises only my original work towards the MPhil,
2. due acknowledgement has been made in the text to all other material used,
3. the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Yaqi Zhu, May 2018

This page intentionally left blank.

Acknowledgements

First I am extremely grateful for Professor Chris Manzie and Dr. Iman Shames' supervision throughout my research. Together they have taught me, consciously or unconsciously, how research is done. In particular, I would like to express my sincere gratitude to Chris for his patience, inspiration, and his professional guidance when I am getting lost in my project. Iman has given me continuous and great support in my MPhil study. Many good ideas were sparked during our discussion, which made my study experience both stimulating and productive. I also appreciate the invaluable comments and constructive criticism from my committee members Professor Stephan Winter and A/Professor Ying Tan.

I gratefully acknowledge the University of Melbourne (Unimelb) for supporting my research financially.

A sincere thank goes to Professor René Boel and Zhenzhen Hao, with whom we have productive collaborations. The enthusiasm René has for his research is very motivational for me, and I believe this influence will continue throughout my life.

A number of people at Unimelb have helped shape my research, and given their best suggestions in my presentation, paper writing, e.t.c. In particular, I would like to thank Ronny Kutadinata, Changfu Zou, Chenyang Liu, Kaixiang Wang, Meng Yuan, Gokul Sankar, Miguel Ramos, Noam Olshina, and David Whittle.

Special mention goes to my best friends in Unimelb, Yu Bai, Tian Liang, Jun Wang, Florence Leong, and Abigail Augustine, for their strong support and help.

Most importantly, my deepest gratitude goes to my family for their unconditional and unparalleled love and encouragement. I am grateful to my parents for giving me the opportunities and experiences that have made me who I am.

To everybody else who accompanied me in this MPhil journey: THANK YOU!

To my beloved parents, Shufen and Lijun.

This page intentionally left blank.

Contents

List of Figures

This page intentionally left blank.

List of Tables

NOMENCLATURE

Acronyms

MFD	Microscopic Fundamental Diagram
AHS	Automated Highway System
SISO	Single-Input-Single-Output
LTI	Linear Time Invariant
LMI	Linear Matrix Inequality
MST	Maximum Spanning Tree
O-D	Origin-Destination

Operators & Functions

$\|X\|$	the absolute value of a complex number X
$X(z)$	the z-transform of a discrete-time signal $x(k)$
$X(s)$	the Laplace transform of a continuous-time signal $x(t)$
$\|\|\cdot\|\|_2$	L_2 norm
$\boldsymbol{A}_{(i,:)}$	the i-th row of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{(:,i)}$	the i-th column of matrix $\boldsymbol{A}$
$\|\|\cdot\|\|_\infty$	$\mathcal{H}_\infty$ norm
$\|\|\cdot\|\|_F$	Frobenius norm
$\mathcal{O}$	big O notation
$\boldsymbol{card}(\cdot)$	the cardinality of a set
$\subseteq$	subset or equal to

$\cap$	the intersection (of two sets)
$\cup$	the union (of two sets)
$\bigcap$	the intersection of an arbitrary nonempty collection of sets, i.e. $(x \in \bigcap_{A \in M} A) \Longleftrightarrow (\forall A \in M, x \in A)$
$\lvert\mathcal{V}\rvert$	the number of nodes in a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$
$\lvert\mathcal{E}\rvert$	the number of edges in a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$
$\lvert V(i)\rvert$	the degree of a node i in a graph
$\mathcal{P}(i,j)$	the path from i to j in a tree
$\mathcal{L}_i(j)$	the level of node j in a rooted tree $\mathcal{T}_i$
$\mathcal{N}(i)$	the neighbour set of a node i
$h(\mathcal{T}_i)$	the height of a rooted tree $\mathcal{T}_i$
$\mathcal{C}_i(g)$	the children set of g in a rooted tree $\mathcal{T}_i$
$S(\mathcal{G})$	sparsity matrix function of $\mathcal{G}$, i.e. $S_{i,j}(\mathcal{G}) = 1$ for $i = j$ and $(i,j) \in \mathcal{E}$, and is zero otherwise.
$\circ$	element-wise multiplication of matrices with same dimensions

Frequently used variables & Sets

$\mathbb{R}$	the set of real numbers
$\mathcal{G}$	a graph
$\mathcal{V}$	the set of nodes in a graph
$\mathcal{E}$	the set of edges in a graph
Γ	undirected tree

$\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$	weighted graph with adjacency matrix $\mathcal{A}$
m	the number of subsystems in an interconnected system
$\boldsymbol{d}^j$	disturbance vector with a single non-zero element at the j-th location
x_i^j	the state of subsystem i under $\boldsymbol{d}^j$
$\boldsymbol{Q}$	closed-loop state matrix of the discrete-time system
$\boldsymbol{F}$	closed-loop state matrix of the continuous-time system
$\boldsymbol{A}$	state matrix of the open-loop discrete-time system
$\boldsymbol{B}$	input matrix of the open-loop discrete-time system
$\boldsymbol{K}$	state feedback matrix of the discrete-time system
$\boldsymbol{u}$	input vector of the discrete-time system
$\boldsymbol{H}$	open-loop state matrix of the continuous-time system
$\boldsymbol{J}$	input matrix of the continuous-time system
$\boldsymbol{R}$	closed-loop state matrix of the continuous-time system

Important parameters used in Chapter 5

Parameters in water supply systems:

$\tilde{\boldsymbol{H}}$	system matrix before normalisation
$\tilde{\boldsymbol{J}}$	input matrix before normalisation
$\boldsymbol{H}$	system matrix after normalisation.
$\boldsymbol{J}$	input matrix after normalisation.

Parameters in perimeter controlled traffic systems:

$\bar{\boldsymbol{A}}$	system matrix of continuous-time model
$\bar{\boldsymbol{B}}$	input matrix of continuous-time model
$\tilde{\boldsymbol{A}}$	system matrix of discrete-time model
$\tilde{\boldsymbol{B}}$	input matrix of discrete-time model
$\boldsymbol{A}$	system matrix of normalised discrete-time model
$\boldsymbol{B}$	input matrix of normalised discrete-time model

This page intentionally left blank.

This page intentionally left blank.

Chapter 1
Introduction

1.1 Background and motivation

Interconnected systems can be found in a wide range of applications, including automated highway systems [1, 2] irrigation systems [3, 4], vehicle and spacecraft formation flights [5, 6], and signal processors [7]. For large-scale interconnected systems, the synthesis and implementation of a centralized controller is often not feasible, due to limited communication resources and large dimension of the problem [8]. Therefore, distributed control laws are widely employed for systems of this type.

As overall knowledge of a distributed control system cannot be achieved by subsystems, one important aspect of network design is accessing how the impact of a disturbance in one subsystem propagates throughout the network. Poorly designed distributed controllers may exhibit an undesired disturbance propagation characteristic, which lead to system failures or network instability.

These phenomenon can be illustrated by an example of automated highway system (AHS), where a group of automated vehicles travel in a platoon. One target of the controller of each vehicle is to avoid collisions by keeping a desired inter-vehicle spacing between its neighbours and itself. To this end, the dynamics of spacing error e are captured. As shown in Figure 1.1, the spacing error is defined as the difference between the actual and desired vehicle spacing, i.e. $e_i := x_{i-1} - x_i - \delta$, where x_{i-1} and x_i are the positions of the $i-1$-th and i-th vehicle, respectively, and δ is the desired inter-vehicle spacing. Figures 1.2a and 1.2b depict the responses of a platoon under two different control laws when the lead vehicle has a sudden deceleration. It can be seen that in Figure 1.2a the spacing errors are attenuated when propagating down through the platoon, which implies that no collision happens as long as the first two vehicles do not collide. On the other

hand, Figure 1.2b shows that under a poorly designed control strategy the peak of spacing errors are magnified through propagation in the vehicle string. This implies that no matter how small the initial disturbance e_1 is, the vehicles farthest from the leader must experience a collision, as long as the vehicle string is long enough.

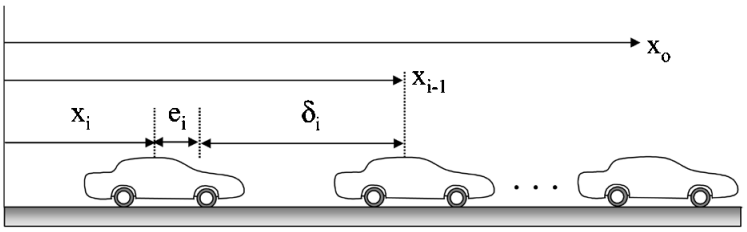
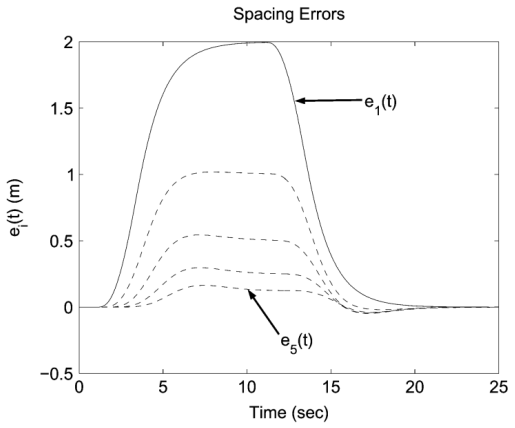
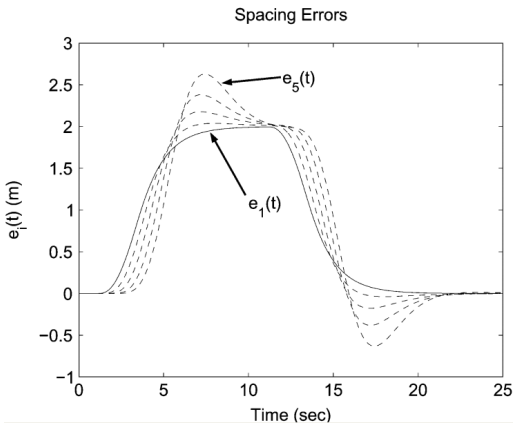


Figure 1.1: AHS platoon and spacing error illustration.[9]



Case 1: Disturbance attenuation.



Case 2: Disturbance magnification.

Figure 1.2: Disturbance propagation in an AHS platoon.[9]

Disturbance propagation in large-scale interconnected systems has been intensively studied in various applications [10–14]. Despite of this, there are still challenges in disturbance propagation analysis and controller synthesis.

In most existing literature, desirable disturbance propagation characteristics are casted as a stability property of the overall system [10–12, 15]. A range of definitions for stability in interconnected systems have been proposed, e.g., see [7, 16–18] and the references there-in. In [18], it is defined as the uniform boundedness of all the states of the interconnected system for all time if the initial states of the subsystems are uniformly bounded, which has been widely adopted in many

practical applications.

This concept of stability implies the attenuation of disturbances when propagating through identical subsystems. However, while the main idea can also be applied to systems with non-identical subsystems, such as heterogeneous vehicle strings [10], the disturbance attenuation property is no longer guaranteed. As the disturbance bound normally has a physical meaning, this may be problematic. For example, as introduced in the AHS platoon control, the bound of the spacing error caused by a disturbance in a vehicle platoon must be less than the nominal inter-vehicle spacing to avoid collisions [19]. In a stable interconnected system under the definition given in [18], this bound can be reached in any of the subsystems in the non-identical case, which makes it difficult to analyse. On the other hand, the largest effect of the disturbance is always caused in the subsystem where it originally occurs in the identical case, as the disturbance is attenuated through propagation. Inspired by this, we impose a requirement stronger than uniform boundedness of the states of all the subsystems to interconnected systems – identical or not – to guarantee disturbance attenuation through propagation.

Disturbance propagation analysis depends on the interconnection topology. One dimensional networks, imposing a string structure, only allow disturbance propagation to either end of the string. This enables relatively straightforward analysis techniques to be used, since there is only one path for the disturbance to follow, however, this overlooks the true structure of many real world systems. A more general mesh connection structure may have multiple propagation paths between subsystems, making it difficult or even impossible to enforce disturbance attenuation when designing the system.

In this regard, tree topologies can be a promising solution, as it can be seen as an intermediate point between strings and more general topology such as meshes. A tree is a graph in which any two nodes are connected by exactly one path. The existence and uniqueness of this path facilitates the disturbance propagation analysis. Trees belong to a representative class of interconnection topologies, and this class of systems captures the salient properties of many real world applications, such as irrigation systems [20, 21], mechanisms [22, 23], robot formation [24, 25], vehicle formation [26, 27], and transportation networks [28].

With this in mind, the major aim of this research is to develop a distributed controller synthesis method with guaranteed disturbance attenuation for tree-structured interconnected systems. In

addition, the extension of the results to arbitrarily connected networks will also be investigated.

1.1.1 Thesis outline

The remainder of this thesis is structured as follows. In Chapter 2, existing literature relating to disturbance propagation methods is discussed critically. It is found that easily verified disturbance attenuation conditions for heterogeneous interconnected systems are still lacking in the literature. Furthermore, disturbance analysis may potentially benefit from a tree topology, which is reviewed subsequently. In addition, as an application example used in this work, traffic perimeter control is reviewed. Finally, research opportunities and objectives are stated in the end of the chapter.

In Chapter 3, rigorous definitions of disturbance attenuation are provided, and disturbance attenuation conditions are derived both for discrete and continuous-time tree-structured interconnected systems. Necessary and sufficient conditions depending on global information are developed first, from which sufficient conditions that can be verified locally are derived by some mathematical transformations.

Following on from the results in Chapter 3, a disturbance attenuating controller design method is proposed in Chapter 4. The controller synthesis is initially formulated as a convex optimisation problem for the overall system, with constraints of stability, disturbance attenuation, and tree topology of the resulting closed-loop system. Later, it is shown that this problem can be broken up into local optimisation problems and solved by each subsystem via dual decomposition method. Therefore, distributed controller design method is presented. Another contribution in this chapter is that an approach to approximate an arbitrarily connected network by a tree is developed. The approximation is transformed into a maximum spanning tree problem.

In Chapter 5, the theoretical results in preceding chapters is validated through a water supply system simulation and a traffic simulation, respectively. In particular, a controller is developed following the method proposed in Chapter 4, and the proposed controller is demonstrated to be able to attenuate disturbance. Furthermore, the time efficiency of sufficient disturbance attenuation conditions proposed in Chapter 3 is demonstrated by a comparison to those necessary and sufficient conditions. Lastly, the performance of the tree approximation method developed in Chapter 4 is demonstrated via a network with 50 subsystems.

Finally, the conclusions and further research opportunities are summarised in Chapter 6.

This page intentionally left blank.

Chapter 2
Literature Review

This chapter will present the literature review in three sections. Firstly, disturbance propagation in interconnected systems will be discussed. Various characteristics of disturbance propagation are summarised, and typical analysis methods proposed in the literature will be critically reviewed. Later, as an available tool for analysing interconnected systems, algebraic graph theory is introduced. Because this work mainly focuses on networks with tree topologies, the relevant aspects of tree graphs is provided in particular. Finally, a typical traffic management strategy – perimeter control – is presented as an applied example of interconnected systems. This transportation control problem will be used as an example to validate the theoretical results in this research.

2.1 Disturbance propagation in interconnected systems

2.1.1 Classification of existing literature

A large body of work has been devoted to disturbance propagation analysis over the past few decades. To locate our work, the classification of the existing literature is first introduced in this section.

Depending on controller and network features, most existing literature can be captured by a combination of the following characteristics.

Unidirectional vs bidirectional

Unidirectional and bidirectional control has been extensively discussed in multi-agent systems, such as robot formation and platoon control [15, 19, 29, 30]. Under unidirectional control strategies, each subsystem only takes control actions in response to its preceding subsystem and other subsystems ahead, while a bidirectional controller takes its following subsystems into consideration as well. Most existing work in AHS control has focused on unidirectional control [9, 31–34],

as it is well-known that adding the information of the following vehicles may sometimes hinder the disturbance attenuation property [19]. Compared to bidirectional control, the analysis of disturbance propagation is generally less complicated in unidirectional cases due to less propagation directions. Despite of this, bidirectional interactions of the subsystems can be found in many real-world networks. For example, a long highway system can be divided into a string of shorter sections for decentralised control, where the transfer flow between neighbouring sections is bidirectional. Bidirectional control strategies are studied for a string structure in [19] and [18]. In networks with two-dimensional interconnection topologies, disturbance propagation discussions can be found in [17, 35, 36].

Homogeneous vs heterogeneous

Most early studies for disturbance propagation focus on networks with identical subsystems (e.g. a platoon consists of identical vehicles and controllers [9, 19, 32]). Subsystems being identical may simplify the disturbance propagation analysis due to the homogeneity of the network. However, it is realised that in many applications the subsystems may have non-identical dynamics. For example, the vehicles may have different characteristics (e.g. trucks, buses, cars, etc.) in a platoon, as is the case in actual traffic. Thus, disturbance propagation in heterogeneous networks are intensively studied for strings [11, 37–39] as well as two-dimensional interconnected systems [36].

String topology vs arbitrary topology

The majority of early research on disturbance propagation focuses on networks with a string structure [16, 19, 40–42]. A disturbance can only propagate to one direction (unidirectional case) or two directions (bidirectional case) in a string, which enables relatively straight forward analysis tools to be used. Later, two-dimensional interconnection topologies, which exist widely in real-world applications, attract much research interest [5, 17, 24, 35]. In [5], the disturbance propagation analysis methods for strings are extended to networks with a mesh topology. However, only unidirectional (called as "look-ahead" in mesh interconnection) control strategies are discussed. Specifically, a look-ahead system is an interconnected system whose (i, j)-th subsystem is connected only to the subsystems (k, l) such that $k < j$ and $l < j$, as schematically shown in Figure 2.1. On the other hand, "look-around" control strategies may increase the difficulty in disturbance propagation analysis, as the disturbance propagation direction is ambiguous. In [35],

disturbance propagation is analysed for arbitrary underlying graphs, and conditions guaranteeing state boundedness of the overall system are proposed.

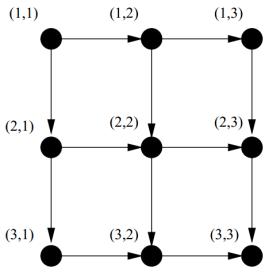


Figure 2.1: Look-ahead control policy in a network with a mesh topology.[5]

To summarise, disturbance analysis for systems with characteristics of bidirectional, heterogeneous, and arbitrary topologies are more difficult than those with unidirectional, homogeneous, and string topologies. In this research, a bidirectional, heterogeneous, and tree structured system is considered.

2.1.2 Typical analysis methods

Various tools have been introduced for disturbance propagation analysis, among which three typical methods will be reviewed.

Frequency-domain analysis

Frequency-domain approaches provide a powerful and handy tool for analysing disturbance propagation in networks with linear subsystems [9,19,31,34,43]. In the following, a representative work [19] will be introduced to show the application of frequency-domain analysis in disturbance propagation.

In [19], every pair of adjacent vehicles are treated as a single-input-single-output (SISO) system, with the input and output being the spacing errors on the upstream and downstream of disturbance propagation path, respectively. The spacing error transfer function is denoted by $\boldsymbol{G}(s)$. The desired disturbance propagation characteristic in a platoon is termed as *string stability* in [19], which is defined as follows:

Definition 2.1. *If the energy (represented by the L_2 norm) of the output error is smaller than the energy of the input error (equivalent to $||\boldsymbol{G}(s)||_\infty < 1$), then it is L_2 string stable.*

This notion can be readily extended to any one-dimensional linear-time-invariant (LTI) interconnected system with identical subsystems. Note that the proposed L_2 string stability guarantees that the energy of the disturbance is attenuated through propagation along the string.

One drawback associated with Definition 2.1 is that the infinity norm of the transfer function $\boldsymbol{G}(s)$ is often difficult to calculate, especially in heterogeneous strings. A computationally efficient approach to determine the value of $||\boldsymbol{G}(s)||_\infty$ is still lacking in the literature.

Lyapunov methods

Lyapunov stability has been extended to define desired disturbance propagation characteristics in distributed control interconnected systems [15, 18, 40].

One representative definition is presented in [18]. Consider the following unidirectional interconnected system:

$$\dot{x}_i = f(x_i, x_{i-1}, ..., x_{i-r+1}), \tag{2.1}$$

where $i \in \mathcal{N}$, $x_{i-j} \equiv 0 \; \forall i \leq j$, $x \in \mathcal{R}^n$, $f : \mathcal{R}^n \times ... \times \mathcal{R}^n$(r times) $\rightarrow \mathcal{R}^n$ and $f(0, 0, ..., 0) = 0$.

Definition 2.2. *The origin $x_i = 0$, $i \in \mathcal{N}$ of* (2.1) *is string stable if for all $\epsilon > 0$, there exists a δ such that*

$$||x_i(0)||_\infty < \delta \Rightarrow \sup_i(||x_i(\cdot)||_\infty) < \epsilon. \tag{2.2}$$

Given a string of identical subsystems, string stability in the sense of definition 2.2 can also guarantee disturbance attenuation, as in the case of definition 2.1. Nevertheless, when the string is heterogeneous, only uniform boundedness can be guaranteed. As discussed in the Introduction, this requirement may not be strong enough to guarantee performance.

Another disadvantage associated with this method is that the derived conditions often rely on parameters that are difficult to determine. For example, in [18], a simple neighbouring-coupled interconnected system with dynamics

$$\dot{x}_i = f(x_{i-1}, x_i, x_{i+1}), \tag{2.3}$$

is string stable if

1. f is globally Lipschitz in its arguments, i.e.

$$|f(x_1, x_2, x_3) - f(y_1, y_2, y_3)| \leq l_1|x_1 - y_1| + l_2|x_2 - y_2| + l_3|x_3 - y_3|, \tag{2.4}$$

2. the origin of $f(0, x_i, 0)$ is globally exponentially stable, and

3. $l_1 + l_3$ is sufficiently small.

The value of 'sufficiently small' parameters l_1 and l_3 depends on the parameters in the converse Lyapunov theorem, which may be difficult to be verified in practice.

Spatial shift operators

Spatial shift operators have shown their advantages in synthesis and analysis in spatially invariant systems. In an early work [44], a bilateral Z-transform which is analogous to spatial shift operators is employed to realise the optimal regulation of a system with countably infinite number of subsystems. Later, linear matrix inequality (LMI) stability results are proposed for a homogeneous system with distributed control in [17]. Then, this work is extended to heterogeneous networks [36] and arbitrarily structured interconnected systems [35]. In [45], it is pointed out that optimal control problem for spatially decaying systems lends itself to distributed solutions without too much loss in performance.

One example of disturbance propagation analysis using spatial shift operators is given in the following [17]. Consider a signal $\boldsymbol{d} := d(t, s_1, ..., s_i, ..., s_L)$ with 1 temporal variable and L independent spatial variables. The spatial shift operator $\boldsymbol{S}_i$ acting on signal $\boldsymbol{d}$ is defined as

$$(\boldsymbol{S}_i \boldsymbol{d}(t))(\boldsymbol{s}) := \boldsymbol{d}(t, s_1, ..., s_{i+1}, ..., s_L), \tag{2.5}$$

for $i = 1, ..., L$. Consider an interconnected system as shown in Figure 2.2(a) with subsystems depicted in Figure 2.2(b), where state x is of size m_0, v_+ and w_+ are of the same size m_+, and v_- and w_- are of the same size m_-. Let $v = [v_+, v_-]^\top$ and $w = [w_+, w_-]^\top$. Assume the system is described by the following state space equation:

$$\begin{bmatrix} \dot{x}(t, \boldsymbol{s}) \\ w(t, \boldsymbol{s}) \\ z(t, \boldsymbol{s}) \end{bmatrix} = \begin{bmatrix} A_{TT} & A_{TS} & B_T \\ A_{ST} & A_{SS} & B_S \\ C_T & C_S & D \end{bmatrix} \begin{bmatrix} x(t, \boldsymbol{s}) \\ v(t, \boldsymbol{s}) \\ d(t, \boldsymbol{s}) \end{bmatrix}, \tag{2.6}$$

where A_{TT}, A_{TS}, A_{ST}, A_{SS}, B_T, B_S, C_T, C_S, and D are constant matrices with compatible dimensions.

Let $\boldsymbol{m} = (m_0, m_+, m_-)$ and define a structured spatial shift operator:

$$\Delta_{S,m} := \begin{bmatrix} \boldsymbol{S} I_{m+} & 0 \\ 0 & \boldsymbol{S}^{-1} I_{m-} \end{bmatrix}. \tag{2.7}$$

Therefore, (2.6) can be translated to

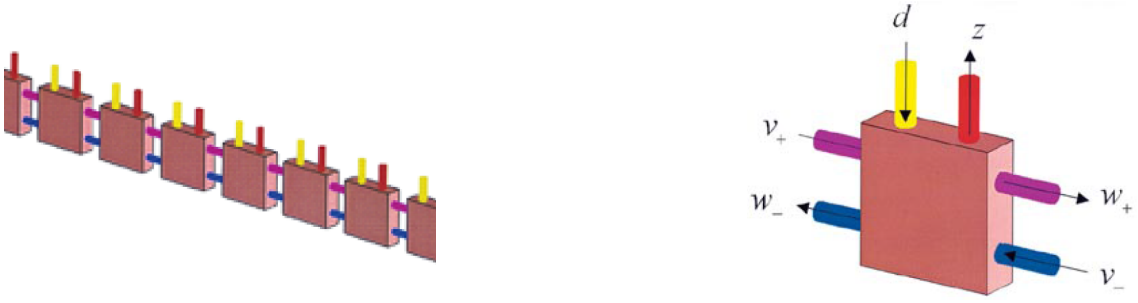
$$\begin{bmatrix} \dot{x}(t,\boldsymbol{s}) \\ (\Delta_{S,m} v(t))(\boldsymbol{s}) \\ z(t,\boldsymbol{s}) \end{bmatrix} = \begin{bmatrix} A_{TT} & A_{TS} & B_T \\ A_{ST} & A_{SS} & B_S \\ C_T & C_S & D \end{bmatrix} \begin{bmatrix} x(t,\boldsymbol{s}) \\ v(t,\boldsymbol{s}) \\ d(t,\boldsymbol{s}) \end{bmatrix}. \tag{2.8}$$

Let $\Delta_m = \mathrm{diag}(\frac{d}{dt} I_{m_0}, \Delta_{S,m})$, and $r(t,s) = [x(t,s), v(t,s)]^\top$. Equation (2.8) can be rewritten as

$$\begin{bmatrix} (\Delta_m r)(t,s) \\ z(t,s) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} r(t,s) \\ d(t,s) \end{bmatrix}, \tag{2.9}$$

where matrices $A = \begin{bmatrix} A_{TT} & A_{TS} \\ A_{ST} & A_{SS} \end{bmatrix}$, $B = \begin{bmatrix} B_T \\ B_S \end{bmatrix}$, and $C = \begin{bmatrix} C_T & C_S \end{bmatrix}$.

By analysing (2.9), LMI stability results can be obtained.



(a) System connection. (b) Subsystem structure.

Figure 2.2: A spatially invariant system with a string connection topology [17].

2.1.3 Summary

The available results in disturbance propagation, including the definitions, conditions, and analysis tools, establish the basis for disturbance attenuation analysis in our work. Based on the characteristics of systems, different classifications of existing disturbance propagation analysis work are introduced in this section. Specifically, the classifications are: unidirectional vs bidirectional, homogeneous vs heterogeneous, and string topology vs arbitrary topology. Furthermore, three typical disturbance propagation analysis tools, i.e. frequency-domain approaches, Lyapunov methods, and spatial shift operators, are introduced.

Although various results have been proposed in literature, there is still much room for improvements. Most existing work on disturbance propagation focuses on stability for interconnected systems, whereas a practically desired property - disturbance attenuation - is commonly neglected. As discussed in the Introduction, states constraints may still be violated even if the system is stable. On the other hand, disturbance attenuation can guarantee that the largest effect of disturbance is always caused in the subsystem where the disturbance originally occurs.

In addition, many results on disturbance propagation are difficult to verify, due to the use of Lyapunov theorems and norms of transfer functions, etc.. For systems of many types, computationally efficient conditions are still lacking in the literature.

This research focuses on bidirectional, heterogeneous, and tree-structured interconnected systems. In particular, a tree is an intermediate point between strings and more general topology such as meshes.

Considering that a graph is naturally a representation of the information topology for an interconnected system, graph theory and the related application in interconnected systems will be reviewed in the next section.

2.2 Graph theory for interconnected systems

Graph theory provides many useful tools in modelling the interconnected systems and relating its topology to control algorithms [46–48]. In this section, an introduction of the relevant aspects of graph theory is given[1] and then its application in interconnected system modelling and analysis

[1] Many materials on graph theory can be found in publications [49–52]. The presentation here is based on [49,51,52].

will be reviewed.

2.2.1 Basics of a graph

The set of all k-element subsets of a set $\mathcal{A}$ is denoted by $[\mathcal{A}]^k$. An unweighted graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ consists of a finite non-empty set of *vertices*, denoted by $\mathcal{V}$, and a set of pairs of vertices, called *edges*, denoted by $\mathcal{E} \in [\mathcal{V}]^2$. For any edge $a = (\alpha,\beta) \in \mathcal{E}$ and $\alpha,\beta \in \mathcal{V}$, it is assumed in this work that $\alpha \neq \beta$, which means that there exist no self-loops in the graph. It is also assumed that each element of $\mathcal{E}$ is unique. In this report, sometimes the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is denoted by $\mathcal{G}$ for brevity. The number of vertices and the number of edges in $\mathcal{G}$ are denoted by $|\mathcal{V}|$ and $|\mathcal{E}|$, respectively. As an example, a graph $\mathcal{G}$ with 9 vertices and 11 edges, i.e. $|\mathcal{V}| = 9$, $|\mathcal{E}| = 11$ is presented in Figure 2.5. As can be seen in the figure, no self-loops or repeated edges are contained in $\mathcal{G}$. Given two graphs $\mathcal{G}(\mathcal{V},\mathcal{E})$ and $\mathcal{G}'(\mathcal{V}',\mathcal{E}')$, if $\mathcal{V} \subseteq \mathcal{V}'$ and $\mathcal{E} \subseteq \mathcal{E}'$, then $\mathcal{G}$ is a subgraph of $\mathcal{G}'$.

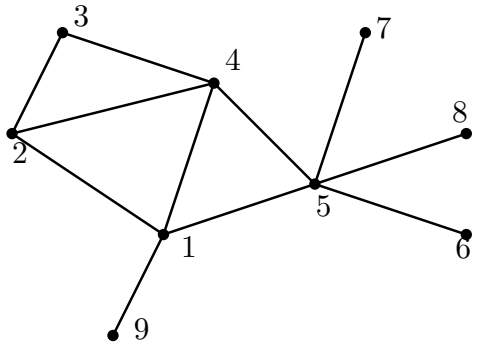


Figure 2.3: An example of a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$.

A graph with the property that for any $(\alpha,\beta) \in \mathcal{E}$, the edge (β,α) is also in $\mathcal{E}$ is said to be undirected. The degree $V(i)$ of a vertex $i \in \mathcal{V}$ in an undirected graph is the number of edges $|\mathcal{E}(i)|$ at i. If $(i,j) \in \mathcal{E}$, then i is a neighbour of j. The neighbour set of a vertex j is denoted by $\mathcal{N}(j)$.

Definition 2.3. *A* path *on $\mathcal{G}(\mathcal{V},\mathcal{E})$ is a non-empty graph $\mathcal{P} = (\mathcal{V}_P,\mathcal{E}_P) \subseteq (\mathcal{V},\mathcal{E})$ of the form*

$$\mathcal{V}_P = \{x_0, x_1, ..., x_k\} \quad \mathcal{E}_P = \{x_0x_1, x_1x_2, ..., x_{k-1}x_k\}, \tag{2.10}$$

where the x_i are all distinct.

If for any vertex in a undirected graph $\mathcal{G}$, there exists a path from itself to all the other vertices in $\mathcal{G}$, then it is said to be *connected*. On the contrary, if there exist disjoint subsets of vertices that cannot be joined by any path, the graph is said to be *disconnected*. This work focuses on connected graphs only. If $\mathcal{P} = x_0 ... x_{k-1}$ is a path and $k \geq 3$, then the graph $\mathcal{C} := \mathcal{P} \cup \{x_{k-1}x_0\}$ is called a *cycle*. A path without cycles is said to be *acyclic*.

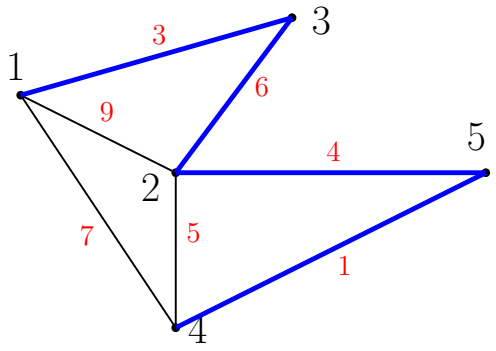


Figure 2.4: An example of a weighted graph and its minimum spanning tree (bold blue).

A weighted graph is denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{A}$ is the weighted adjacency matrix with non-negative entries. Specifically, the (i, j)-th entry of matrix $\mathcal{A}$ represents the weight of edge (i, j) in graph $\mathcal{G}$. For undirected graphs, matrix $\mathcal{A}$ is symmetric. An example of a weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$ is depicted in Figure 2.4, where the weighted adjacency matrix is given by:

$$\mathcal{A} = \begin{bmatrix} 0 & 9 & 3 & 7 & 0 \\ 9 & 0 & 6 & 5 & 4 \\ 3 & 6 & 0 & 0 & 0 \\ 7 & 5 & 0 & 0 & 1 \\ 0 & 4 & 0 & 1 & 0 \end{bmatrix}. \tag{2.11}$$

As a comparison of undirected graphs, if there exists at least one edge (β, α) in a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ that $(\alpha, \beta) \notin \mathcal{E}$, then the graph is a *digraph* (directed graph). If there is a directed edge from vertex i to vertex j, then i is defined as the parent vertex and j is defined as the child vertex.

In the following section, tree graphs will be introduced in particular, as the acyclic properties of trees can potentially benefit the analysis of disturbance propagation.

2.2.2 Basics of a tree graph

The following theorem gives the properties of a tree [49].

Theorem 2.1. *The following assertions are equivalent for a graph* $\Gamma(\mathcal{V}, \mathcal{E})$*:*

- Γ *is a tree.*
- *Any two vertices of* Γ *are linked by a unique path in* Γ*.*
- Γ *is minimally connected, i.e.* Γ *is connected but* $\Gamma(\mathcal{V}, \mathcal{E} \setminus \{e\})$ *is disconnected for every edge* $e \in \Gamma$*.*
- Γ *is maximally acyclic, i.e.* Γ *contains no cycle but* $\Gamma(\mathcal{V}, \mathcal{E} \cup \{xy\})$ *does, for any two non-adjacent vertices* $x, y \in \mathcal{V}$*.*

In a tree graph $\Gamma(\mathcal{V}, \mathcal{E})$, a vertex is called a leaf if it has the degree of 1. For simplicity, the path between any two vertices i, j in a tree Γ is denoted by $\mathcal{P}(i, j)$. The example of a tree Γ with 10 vertices is shown in Figure 2.5. As an example, the (unique) path between vertex 3 and vertex 6 is highlighted in blue colour.

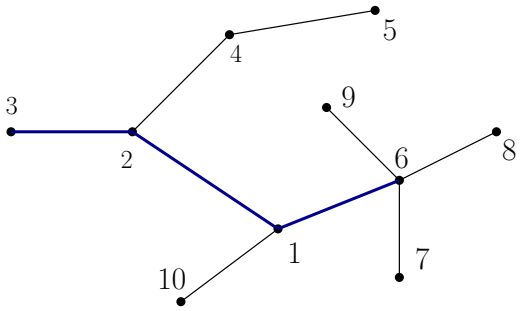


Figure 2.5: An example of a tree Γ and the path $\mathcal{P}(3, 6)$ (bold blue).

When conducting the disturbance propagation analysis, it is useful to introduce a notion of *directed tree* (or *rooted tree*) by treating the disturbance vertex of origin as the root.

Definition 2.4 (Rooted tree)**.** *A rooted tree is a digraph, where every vertex has exactly one parent except for one vertex, called the root, which has no parent, but has a directed path to every other vertex.*

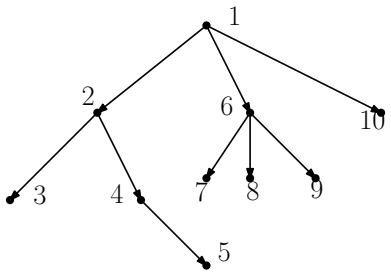


Figure 2.6: Rooted tree $\mathcal{T}_1$.

A rooted tree is denoted by $\mathcal{T}_i$ where the subscript i refers to the root. In a rooted tree $\mathcal{T}_i$, the level $L_i(j)$ of a vertex j is its distance from the root i, i.e. $L_i(j) = |\mathcal{P}_i(i,j)|$. The height $h(\mathcal{T}_i)$ of a rooted tree $\mathcal{T}_i$ is the maximum level in the tree. For example, the rooted tree $\mathcal{T}_1$ shown in Figure 2.6 is transformed from Γ by taking vertex 1 as the root. The height of the rooted tree is 3. Suppose the topology of an interconnected system can be represented by tree Γ, then the arrows in the rooted tree $\mathcal{T}_1$ show the disturbance propagation direction when a single disturbance occurs in subsystem 1. The children set of a vertex g in the rooted tree $\mathcal{T}_i$ is denoted as $\mathcal{C}_i(g)$.

Definition 2.5 (Minimum spanning tree)**.** *If a weighted undirected tree* $\Gamma(\mathcal{V},\mathcal{E},\mathcal{A})$ *satisfies:*

1. *it is a subgraph of a graph* $\mathcal{G}(\mathcal{V}',\mathcal{E}',\mathcal{A}')$*, i.e.* $\mathcal{E}\subseteq\mathcal{E}'$, $\mathcal{V}\subseteq\mathcal{V}'$,
2. *it includes all vertices of* $\mathcal{G}$*, i.e.* $\mathcal{V}=\mathcal{V}'$*, and*
3. *the weighted adjacency matrix satisfies that* $\mathcal{A}(i,j)=\mathcal{A}'(i,j)$ *for every edge* $(i,j)\in\mathcal{E}$*,*

then Γ *is a* spanning tree *of* $\mathcal{G}$*. In particular, if a spanning tree* $\Gamma(\mathcal{V},\mathcal{E},\mathcal{A})$ *of graph* $\mathcal{G}(\mathcal{V}',\mathcal{E}',\mathcal{A}')$ *has the minimum total edge weight, then it is called* minimum spanning tree*.*

As an example, the minimum spanning tree in graph $\mathcal{G}$ is highlighted in blue colour, as shown in Figure 2.4.

The research of algorithms for finding a minimum spanning tree can be traced back to 1920s, when Otakar Borůvka first developed a greedy algorithm to solve the problem [53]. However, when discussing this type of problems, researchers commonly refer to two other later but more famous algorithms, i.e. Kruskal's algorithm [54] and Prim's algorithm [55]. The characteristic of Prim's algorithm is that it starts from an arbitrary vertex and then grows the minimum spanning

tree one edge at a time. All these three run in polynomial time, so there has been interest in developing more computationally-efficient algorithms [56][57].

The *maximum spanning tree* (MST) in a undirected weighted graph $\mathcal{G}$ is the spanning tree with the maximum possible total edge weight. By negating the edge weights, i.e. $\mathcal{A} := -\mathcal{A}$, the algorithms for minimum spanning trees can be readily applied to this type of problems.

The sparsity matrix function of a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is denoted by $S(\mathcal{G}) \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, where $S_{i,j}(\mathcal{G}) = 1$ for $i = j$ and $(i,j) \in \mathcal{E}$, and is zero otherwise. Let $\circ$ denote the entry-wise multiplication of matrices with the same dimensions. If a matrix $\boldsymbol{A}$ satisfies $\boldsymbol{A} \circ S(\mathcal{G}) = \boldsymbol{A}$, then $\boldsymbol{A}$ is a $\mathcal{G}$-structured matrix. Particularly, if the graph is a tree $\Gamma(\mathcal{V},\mathcal{E})$, then $\boldsymbol{A}$ is a Γ-tree-diagonal matrix.

Based on the above preliminaries, a graph can be used to model an interconnected system, with vertices representing subsystems (e.g. the vehicles or robots in formation control), and with the directed edges representing the transfer flow (e.g. traffic flow in transportation systems, or information exchange in communication systems) between the subsystems. Furthermore, the properties of the interconnections of the system can be readily read from the graph. For example, an interconnected system may be modelled as a weighted graph, as shown in Figure 2.4. It can be seen from the figure that this is a connected graph with cycles in it. In the following section, the application of graph theory in interconnected systems will be reviewed.

2.2.3 Application in interconnected systems

The graph theory has been applied to modelling the interconnected systems since 1970s, when decentralized control attracted a great deal of interest [58]. In particular, it has been used to capture the interconnection topology of flight formations [59–62], robot formations [24, 25, 63], power systems [64], irrigation systems [20, 21, 65, 66], etc.. In [59], the edges are allocated with costs, and the Dijkstra algorithm is utilized to find the optimal topology with respect to the given costs. The stability of nonlinear systems with linear feedback interconnections is explored via graph Laplacian in [67]. In [68], routing algorithms are discussed for interconnection networks.

Tree topologies has been used in a great many real-world interconnected systems, due to the fact that trees require the least connections while keeping the subsystems connected. Examples of tree-structured networks can be found in irrigation systems [20, 21, 65, 66], mechanisms [22, 23],

robot formation [24, 25], vehicle formation [26, 27], etc.. An automated irrigation system is installed in irrigated land at Jumilla (Murcia, Spain), where the hydraulic network has a tree topology [20]. In [24], a group of autonomous mobile robots are controlled to achieve predetermined tree-like formations. Similarly, three tree-structured formation patterns are considered in [69], and input-to-state stability of a multi-agent system is analysed using the graph theoretic representation of the formation via the adjacency matrix.

2.2.4 Summary

Graph theory is a powerful tool in the analysis of interconnected systems, in terms of describing the interconnection topology as well as capturing system characteristics. In particular, tree-structured interconnected systems have been explored both in literature and in practice.

Despite this, disturbance attenuation analysis for tree-structured interconnected systems has not been considered. Furthermore, most results derived for tree networks cannot be directly applied in systems with non-tree topologies at the current, as the approach to approximate a non-tree network to a tree is still lacking in the literature.

Networked theory for disturbance attenuation has many practical applications, as introduced in Section 2.2.3, but perimeter control in traffic networks is the one used throughout this thesis. Therefore, perimeter control is reviewed in the following section.

2.3 Perimeter control

A vast variety of urban traffic modelling and control strategies have been proposed in the literature [70–73], but still a major challenge is the deployment of advanced and efficient traffic control strategies in large-scale networks, with particular focus on addressing congestion phenomena [74]. It is generally believed that traffic-responsive and coordinated signal control systems are potentially more efficient than fixed-time and isolated control settings. From this perspective, a centralized adaptive control scheme with an elaborated traffic model, in which the traffic flow dynamics of each link and intersection are captured, seems ideal for tackling the congestion problem. However, the detailed modelling for a large-scale urban network would be a very complex task [75]. Moreover, the centralized control with such a high-dimensional model is usually infeasible for

real-time implementation due to the computational complexity [76].

A way to realise a real-time traffic controller for a large-scale urban network, and in particular to alleviate the congestion problem, is to utilize a two-level hierarchical control scheme [77]. The traffic network is partitioned into smaller regions to form the following two levels. At the lower level, the control strategies based on the detailed model are applied to smooth the flows and to improve the mobility within the regions. At the upper level, the overall region dynamics are aggregately described by a simplistic traffic model which captures the main characteristics of congestion and relies on measurable parameters. Based on this aggregate model, the transfer flows between regions are controlled so as to achieve a better utilization of the traffic capacity.

2.3.1 Aggregate modelling method and the control algorithms for a traffic network

At the network-level, the system modelling is based on the conservation equation of vehicles. The conservation equation of a traffic network is given by:

$$\dot{n}(t) = q_{in}(t) - q_{out}(t) + d(t), \tag{2.12}$$

where $n(t)$ is the *accumulation*, i.e. the number of vehicles in the network; $q_{in}(t)$ and $q_{out}(t)$ are the inflow and outflow of the network at time t, respectively; and $d(t)$ is the disturbance flow at time t. The relationship between the number of vehicles $n(t)$ and the outflow $q_{out}(t)$ is required by the model. This relationship can be described by the *macroscopic fundamental diagram* (MFD).

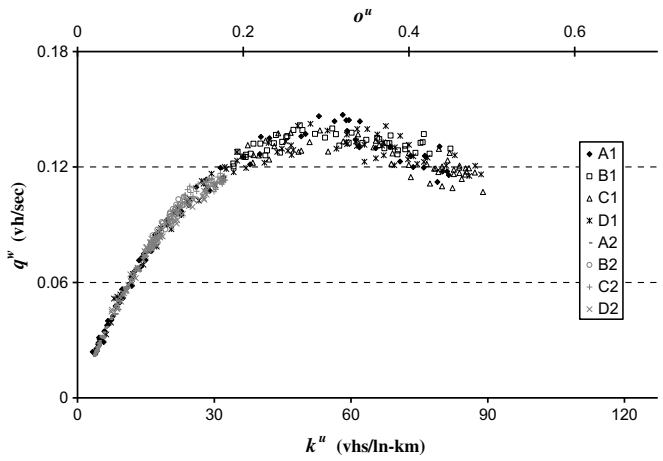


Figure 2.7: A typical macroscopic fundamental diagram (MFD) for a traffic network [78]

MFD abstracts a network as a unimodal, low-scatter relationship between traffic variables. As shown in Figure 2.7, MFD relates the *average flow*[2] q^u, i.e. the averaged number of vehicles passing through a point on the road, and *average occupancy*[3] o^u, i.e. the averaged percent of time a point on the road is occupied by vehicles. Note that each point in Figure 2.7 is obtained by averaging over both time and space. Specifically, each data is averaged over a period of time (5 minutes in this case) and over all lanes in the studied area. In general, the system capacity increases with occupancy until a critical point is reached, and additional vehicles will result in the drop in average flow, and then congestions could follow.

The theoretical physical model of MFD is initially developed by Godfrey [79], while similar works are also done by [80], [81], [82] and [83]. The existence of the MFD is verified by empirical findings in congested urban networks in Yokohama [78]. Nikolas Geroliminis et al also shows there is a robust linear relationship between the average flow and the *trip completion flow*, i.e. the number of vehicles in the network [78]. This property is important for monitoring purposes as the average flow can be monitored by suitable sensors (e.g., inductive loop detectors) while the measurement of trip completion flow requires a wide deployment of GPS.

When incorporating MFD into the modelling of a region, the scattered relationship in Figure 2.7 is often abstracted by a polynomial function, as shown in Figure 2.8. The difference between the real data and the polynomial function is captured by the disturbance term in (2.12). Furthermore, to close the loop in (2.12), the relationship between o^u and q^u is proportionally transformed to the relationship between the accumulation and the trip completion flow $O(n)$.

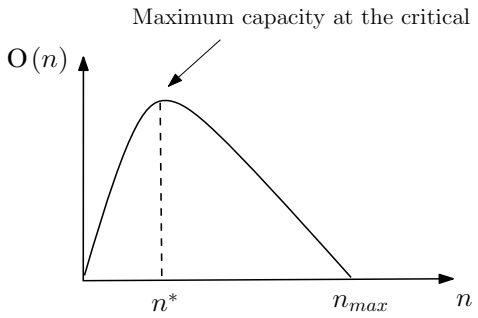


Figure 2.8: A typical macroscopic fundamental diagram (MFD) used for traffic network modelling

[2]The average flow is also called 'circulating flow' or 'space-mean flow' in some papers.

[3]The horizon axis at the bottom of Figure 2.7 is *density* $k^u := o^u/s$, where s is the space-mean effective vehicle length. As s is constant, o^u is proportional to k^u. Therefore, we will focus on o^u only.

An important observation about the MFD is that the spatial distribution of vehicle density in the network is one of the key factors that affects the shape and scatter of an MFD [84–86]. A heterogeneous network[4] may not exhibit a well defined MFD, as the scatter is higher especially for the decreasing part of the MFD, which may introduce large disturbances in the model (2.12) [87]. Therefore a heterogeneous network is partitioned into homogeneous regions[5] with small link densities, and each region has a well-defined MFD. Due to the interactions between the regions in the controlled network, the dynamics of the regions should be modeled to achieve an efficient control. Various modelling methods have been developed for multi-region perimeter controlled traffic networks [74, 88, 89], among which a representative one will be introduced [74, 90, 91].

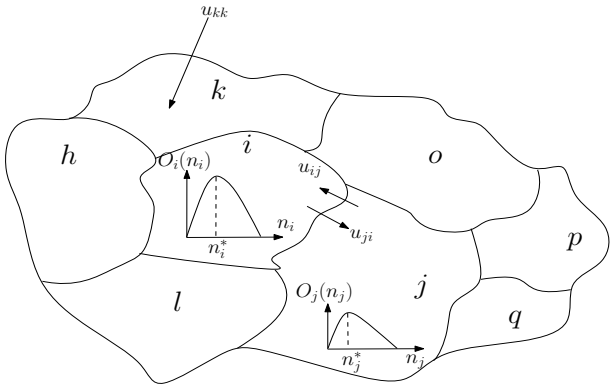


Figure 2.9: A multi-region perimeter controlled traffic network

A traffic network with multi-region perimeter control is schematically shown in Figure 2.9. Let $n_i(t)$ and $\zeta_i(t)$ be the accumulation of vehicles and the disturbance in region i at time t, respectively. Similar to (2.12), the conservation equation for region i is given by:

$$\dot{n}_i(t) = q_{i,in}(t) - q_{i,out}(t) + \zeta_i(t), \tag{2.13}$$

where $q_{i,in}(t)$ and $q_{i,out}(t)$ are the inflow and outflow of region i at time t, respectively.

The trip completion flow in region i is denoted by $O_i(n_i(t))$. In this model, the outflow is

[4] If the density variances between different links in a region is high, then it is called a *heterogeneous region*. Generally speaking, the vehicles are unevenly distributed in a heterogeneous region.

[5] If the density variances between different links in a region is low, then it is called a *homogeneous region*. Generally speaking, the vehicles are almost evenly distributed in a homogeneous region.

assumed to be given by the MFD, i.e.

$$q_{i,out} = O_i(n_i(t)). \tag{2.14}$$

The control variable $\beta_{ih}(t)$ controls the fraction of flow rate in region i that is allowed to enter region h [6]. The inflow is calculated by the sum of transfer flows from the adjacent regions to region i, i.e.

$$q_{i,in}(t) = \sum_{j \in \mathcal{S}_i} \beta_{ij}(t) O_j(n_j(t)), \tag{2.15}$$

where $\mathcal{S}_i$ is the set of source regions of i whose outflows go to region i. Note that the inflow and outflow of the regions located at the periphery of the network, e.g. region k, also include the transfer flows at the perimeter of the network. An assumption is made that the input flow from the perimeter to region k is considered proportional to the trip completion flow of region k, $O_k(n_k)$. Additionally, the model mismatch due to the internal trip completion flow can be treated as disturbance. Thus, the system dynamics is given by

$$\dot{n}_i(t) = \sum_{j \in \mathcal{S}_i} \beta_{ji}(t) O_j(n_j(t)) - O_i(n_i(t)) + \zeta_i(t), \tag{2.16}$$

Given a set of steady state points $\hat{n}_i$, $\hat{\beta}_{ij}$ and $\hat{\zeta}_i$, the nonlinear model (2.16) can be linearised around these points, which yields

$$\Delta\dot{n}_i(t) = \sum_{j \in \mathcal{S}_i} O_j(\hat{n}_j)\Delta\beta_{ji}(t) + \sum_{j \in \mathcal{S}_i} \hat{\beta}_{ji} O_j'(\hat{n}_j)\Delta n_j(t) - O_i'(\hat{n}_i)\Delta n_i(t) + \Delta\zeta_i(t), \tag{2.17}$$

where $O_j'(\hat{n}_j)$ and $O_i'(\hat{n}_i)$ are the derivatives of the trip completion flows at the operating points of region h and i, respectively. Since $\Delta\boldsymbol{n}(t)$ represents the deviation of accumulation from the desired point, it is also called the accumulation error.

The accumulation error dynamics (2.17) can be rewritten in a vector form as follows:

$$\Delta\dot{\boldsymbol{n}}(t) = \boldsymbol{A}\Delta\boldsymbol{n}(t) + \boldsymbol{B}\Delta\boldsymbol{\beta}(t) + \Delta\boldsymbol{\zeta}(t), \tag{2.18}$$

[6] In implementation, the obtained $\beta_{ih}(t)$ values are translated to arriving flows (by multiplying $\beta_{ih}(t)$ by $O_i(n_i(t))$) and used to define the green periods of the traffic lights located at the boundary of region i.

where $\Delta \boldsymbol{n}(t) \in \mathbb{R}^m$, $\Delta \boldsymbol{\beta}(t) \in \mathbb{R}^v$, and $\Delta \boldsymbol{\zeta}(t) \in \mathbb{R}^m$ are the state deviations vector of $\Delta n_i(t)$ ($i = 1, 2, ...m$), the control deviations vector of $\Delta \beta_{ij}(t)$ ($\forall i = 1, 2, ...m$, $j \in \mathcal{S}_i$), and the disturbance deviations vector of $\Delta \zeta_i(t)$ ($i = 1, 2, ...m$), respectively; $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{B} \in \mathbb{R}^{m \times v}$ are the system and input matrices, respectively.

The control objective for the multi-region model is to distribute the accumulation of each region evenly while maintaining the rate of vehicles that are allowed to enter each region around the set point. Therefore, one candidate cost function is given by [74] as

$$J(\boldsymbol{u}) = \frac{1}{2} \int_0^\infty (\Delta \boldsymbol{n}^\top(t) \boldsymbol{Q} \Delta \boldsymbol{n}(t) + \Delta \boldsymbol{\beta}^\top(t) \boldsymbol{R} \Delta \boldsymbol{\beta}(t)), \tag{2.19}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the weighting matrices defined by the controller designer.

A multi-variable state feedback regulator is developed based on the above cost function in [74]. The simulation results show that this control algorithm is capable of alleviating congestions under some tested scenarios. However, the centralised control strategy may experience difficulties due to limited communication resources when applied to networks with a large number of partitions.

2.3.2 Discussion

Multi-region perimeter controlled traffic networks are a suitable example of application of this work for the following reasons.

First, multi-region traffic networks can be seen as an interconnected system, with regions being subsystems, and traffic and communication transfer flows being connections between subsystems.

Second, trees are a representative topology of road networks [92]. Although tree structures have not been considered, it is reasonable to partition some multi-region perimeter control networks into a tree topology, if their road networks have a geometric "tree-like" structure.

Furthermore, a large-scale heterogeneous network may have a great number of regions, thus centralized perimeter control algorithm[7] might be not practical due to the controller size and communication requests. On the other hand, the multi-region perimeter control for such a system can potentially realized through the use of a distributed control scheme[8], where less communication

[7] In a multi-region network, the sum of the information of all the regions in the network is called the *global information*. If the global information can be obtained by the multi-region perimeter controller when it is implemented, then it is called a *centralized control strategy*.

[8] In a multi-region network, the *local information* for a region is the sum of the information of itself and the adjacent

effort is required. This issue has not been directly considered in the existing literature.

Lastly, the multi-region perimeter controlled network would benefit from the application of disturbance attenuation theory. A system with a large number of regions would become congested, if the disturbance in a region is magnified as it propagates to the network. However, the interactions between regions in the presence of disturbances have not been directly considered in the literature.

2.3.3 Summary

In this section, multi-region perimeter control is introduced as a promising method to alleviate congestion, especially in large-scale heterogeneous urban areas. Furthermore, perimeter control is justified as a suitable application of this work in Section 2.3.2.

2.4 Research aims

Following the literature review, it appears that there exist several opportunities that built upon reported literature. This research mainly focuses on the following:

1. **To derive conditions on tree-structured interconnected systems to guarantee that a disturbance is always attenuated when it is propagating in the network.**

   As discussed in Section 2.1, existing literature of disturbance propagation mainly focuses on boundedness of states, whilst the case where a poorly designed controller that may enlarge a disturbance somewhere in the network during its propagation and violate the state constraints has not been considered. The constraints of state are frequently related to safety conditions or operational characteristics of the system, thus it is important to take them into consideration.

   One possible approach to bridge the gap is to enforce disturbance attenuation. As such, the largest effect of disturbance always occurs in the subsystem where it is originally injected, and constraints will not be violated in the overall network as long as it is well controlled in the origin subsystem.

regions. If for each partitioned region, there is a perimeter controller, and only local information can be obtained by these controllers, then it is called a *distributed control scheme*.

Disturbance attenuation requires the identification of disturbance propagation direction, which is ambiguous in cyclic networks. As discussed in Section 2.2, tree networks appears to be an opportunity to simplify the analysis, as their acyclic feature can facilitate the identification of disturbance propagation direction.

To facilitate fast verification, the proposed conditions should have high computational efficiency. Furthermore, it is desired that these conditions depend only on local information, in order that they can be incorporated into distributed controller design.

2. **Based on the conditions proposed in Aim 1, to develop a distributed controller synthesis method for a tree-structured network that guarantees tree topology, disturbance attenuation, and stability.**

   As discussed in Section 2.3, centralised controller design may not be possible for some networks, especially when faced with a large number of subsystems. In general, centralised approaches require extensive communications among subsystems, which may increase the cost dramatically or is even impossible. In addition, the high dimensionality of a large-scale interconnected system may hinder centralised design due to limited computation resources. Therefore, the system may benefit from distributed design.

   An optimisation problem should be formulated to facilitate controller synthesis, where the constraints are tree topology, disturbance attenuation, and stability. The formulated problem is desired be convex, so a global optimal solution can be found. To facilitate distributed controller design, this problem should be able to be broken up into subproblems, which can be solved by subsystems through local information exchange.

3. **To propose an approach to approximate an arbitrarily connected network by a tree, so that the results proposed in Aims 1 and 2 can be extended to non-tree cases.**

   Both Aims 1 and 2 focus on tree-structured networks, whilst non-tree connections widely exist in real-world systems. If a non-tree network behaves close enough to a tree, then the results proposed for tree-structured systems can also be applied in such networks. Hence, a tree approximation method should be proposed to extend the scope of this research.

4. **To validate/demonstrate the theoretical findings in Aims 1, 2, and 3 through applica-**

tion examples.

As discussed in Section 2.3, multi-region perimeter controlled traffic networks are suitable application examples for tree-structured interconnected discrete-time systems. The subsystems in a perimeter controlled transportation system are the partitioned regions, and the interconnections between subsystems are traffic transfer flows and information exchange. Tree-structured partition can be realised in many real-world scenarios, as radius transportation infrastructure layout is widely used in many urban areas. Additionally, an interconnected pumped water supply system is used for the verification of the continuous-time results.

Corresponding to Aims 1 to 3, the simulation should mainly include three aspects. First, a controller should be developed following the proposed synthesis method so that Aim 2 can be validated. Moreover, given the disturbance attenuating controller, the time efficiency of the proposed conditions in Aim 1 should be illustrated. Finally, the performance of the tree approximation method proposed in Aim 3 should be demonstrated.

This page intentionally left blank.

Chapter 3

Disturbance attenuation conditions

From the conclusion in Chapter 2, the vast majority of disturbance propagation analysis focuses on stability of the network, whilst another desired property of the system, i.e. disturbance attenuation, is lacking in the literature.

With that in mind, the aim of this chapter is to propose disturbance attenuation conditions. First, disturbance attenuation is defined for a continuous-time tree-structured network in Section 3.1.1. It is followed by the development of necessary and sufficient disturbance attenuation conditions in Section 3.1.2. The verification of these conditions depends on global information, and they are computationally inefficient. Therefore, sufficient conditions, which can be checked locally and relatively fast, are also proposed in Section 3.1.2. Finally, these results are extended to discrete-time systems in Section 3.2.

3.1 Disturbance attenuation analysis in continuous-time systems

3.1.1 Preliminaries

In this section, a disturbance attenuation definition will be proposed for a tree-structured continuous-time system.

Consider a interconnected continuous-time system with m scalar subsystems, where the interconnection of the subsystems can be represented by a tree $\Gamma(\mathcal{V},\mathcal{E})$. Let $x_i(t)$ denote the state of subsystem i at time t. Suppose the dynamics of the overall system can be captured by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{d}(t), \tag{3.1}$$

where $\boldsymbol{x}(t) := [x_1(t),...,x_m(t)]^\top$ is the state vector; $\boldsymbol{d}(t) \in \mathbb{R}^m$ is the disturbance vector with

The continuous-time results of this chapter has been published as a conference paper [28], and the discrete-time results of this chapter has appeared in a journal paper [93] (under review).

bounded energy; $\boldsymbol{F} \in \mathbb{R}^{m\times m}$ is a Γ-tree-diagonal matrix.

To assess how the impact of a disturbance in one subsystem propagates throughout the network, the following assumption is made.

Assumption 3.1. *A disturbance can only occur in one subsystem at any point in time in system* (3.1)*, i.e., vector* $\boldsymbol{d}(t)$ *only has one non-zero element.*

Let $\boldsymbol{d}^j(t)$ denote a disturbance vector with a single non-zero element at the j-th location. The state of the i-th subsystem under $\boldsymbol{d}^j(t)$ is denoted by $x_i^j(t)$. When a disturbance originally occurs in subsystem j and propagates to the rest of the system, the propagation direction can be represented by the edges pointing away from root j in the rooted tree $\mathcal{T}_j(\mathcal{V},\mathcal{E})$.

Disturbance attenuation can be defined under different measures associated with different physical meanings, such as peak attenuation (under L_∞ norm measure) and energy attenuation (under L_2 norm measure). This research adopts the L_2 norm, which implies energy attenuation of a disturbance. To analyse disturbance propagation, the energy of the states of neighbouring subsystems will be compared. Specifically, in analogy to the bidirectional case in string stability analysis [19], under $\boldsymbol{d}^j(t)$, we treat each pair of neighbouring subsystems i and $o \in \mathcal{C}_j(i)$ in $\mathcal{T}_j$ as a SISO system, where the input and output are the states of the i-th and o-th subsystem, respectively. Under $\boldsymbol{d}^j(t)$, the transfer function between the two is given by:

$$G_{i,o}^j(s) \coloneqq \frac{X_o^j(s)}{X_i^j(s)}, \tag{3.2}$$

where $X_i^j(s)$ and $X_o^j(s)$ are the Laplace transforms of $x_i^j(t)$ and $x_o^j(t)$, respectively.

The energy of the input $x_i^j(t)$ and output $x_o^j(t)$ will be compared for each SISO system, and this comparison is facilitated by the $\mathcal{H}_\infty$ norm of $G_{i,o}^j(s)$, i.e.

$$||G_{i,o}^j(s)||_\infty = \sup_{x_i^j(t)\neq 0} \frac{||x_o^j(t)||_2}{||x_i^j(t)||_2}. \tag{3.3}$$

Given that $G_{i,o}^j(s)$ is a transfer function of a continuous-time SISO system, $\left\|G_{i,o}^j(s)\right\|_\infty$ can be simplified as

$$||G_{i,o}^j(s)||_\infty \coloneqq \sup_\omega |G_{i,o}^j(j\omega)|. \tag{3.4}$$

Based on the string stability definitions given by [10] and [19], disturbance energy attenuation is defined as below.

Definition 3.1 (L_2 disturbance attenuation of a tree-structured continuous-time system)**.** *Consider an interconnected continuous-time system as described by* (3.1)*. Given any $\boldsymbol{d}^j(t)$, if the energy of the output state $x_o^j(t)$ is always smaller than the energy of the input state $x_i^j(t)$ for every pair of neighbouring subsystems i and $o \in \mathcal{C}_j(i)$ in $\mathcal{T}_j$, i.e.*

$$||G_{i,o}^j(s)||_\infty < 1 \ \forall o \in \mathcal{C}_j(i), \ \forall i, o \in \mathcal{V}, \tag{3.5}$$

then the system is subsystem-j L_2 disturbance attenuating. If the system is subsystem-j L_2 disturbance attenuating for all $j \in \mathcal{V}$, then the system is L_2 disturbance attenuating.

Remark: L_2 disturbance attenuation guarantees that the energy of states, which is essentially the effect of the disturbance $\boldsymbol{d}^j(t)$,[1] is always attenuated along the propagation path from the originally disturbed subsystem to the rest of the system.

3.1.2 Disturbance attenuation conditions

This section aims to propose conditions on system (3.1), such that disturbance attenuation property in the sense of Definition 3.1 can be determined.

Problem 3.1. *Given system (3.1), find conditions on matrix $\boldsymbol{F}$, such that the system is L_2 disturbance attenuating in the sense of Definition 3.1.*

Necessary and sufficient conditions

Assume system (3.1) is asymptotically stable. After Laplace transform, define matrix

$$\boldsymbol{M}(s) \coloneqq (s\boldsymbol{I} - \boldsymbol{F})^{-1}. \tag{3.6}$$

Note that $(s\boldsymbol{I} - \boldsymbol{F})^{-1}$ exists under the asymptotic stability assumption. Assume $M_{i,j}(s) \neq 0$, $\forall i, j \in \mathcal{V}$.[2]

[1] We assume the states are at the origin in the absence of disturbance.

[2] It is reasonable to assume $M_{i,j}(s) \neq 0$, otherwise the graph is disconnected.

Define two sets of nodes in Γ as follows:

$$\mathcal{V}^{io} := \{j | j \in \mathcal{V}, \mathcal{P}(i,o) \subseteq \mathcal{P}(j,o)\} \tag{3.7}$$

$$\mathcal{V}^{oi} := \{\bar{j} | \bar{j} \in \mathcal{V}, \mathcal{P}(i,o) \subseteq \mathcal{P}(i,\bar{j})\}. \tag{3.8}$$

Necessary and sufficient conditions are given in the following proposition.

Proposition 3.1 (Necessary and sufficient conditions for continuous-time systems)**.** *Consider an interconnected continuous-time system as described by* (3.1)*. The system is L_2 disturbance attenuating in the sense of Definition 3.1 if and only if $\forall i \in \mathcal{N}(o)$, $\exists f \in \mathcal{V}^{io}$ and $\exists \bar{f} \in \mathcal{V}^{oi}$ such that*

$$\left\| \frac{M_{o,f}(s)}{M_{i,f}(s)} \right\|_\infty < 1, \tag{3.9}$$

$$\left\| \frac{M_{i,\bar{f}}(s)}{M_{o,\bar{f}}(s)} \right\|_\infty < 1. \tag{3.10}$$

To prove for Proposition 3.1, the following lemma is introduced.

Lemma 3.1. *System* (3.1) *is L_2 disturbance attenuating in the sense of Definition 3.1 if and only if for each pair of neighbouring subsystems $i \in \mathcal{N}(o)$,*

$$\left\| \frac{M_{o,j}(s)}{M_{i,j}(s)} \right\|_\infty < 1 \tag{3.11}$$

and

$$\left\| \frac{M_{i,\bar{j}}(s)}{M_{o,\bar{j}}(s)} \right\|_\infty < 1 \tag{3.12}$$

hold for all $j \in \mathcal{V}^{io}$ and $\bar{j} \in \mathcal{V}^{oi}$, where sets $\mathcal{V}^{io}$ and $\mathcal{V}^{oi}$ are defined in (3.7) *and* (3.8)*, respectively.*

Proof. Conditions (3.11) and (3.12) are equivalent to the following condition

$$\left\| \frac{M_{o,j}(s)}{M_{i,j}(s)} \right\|_\infty < 1, \quad o \in \mathcal{C}_j(i), \quad \forall i,j,o \in \mathcal{V}. \tag{3.13}$$

Thus, the proof for conditions (3.11) and (3.12) can be transformed to proving that (3.13) is a necessary and sufficient condition.

The proof includes two steps: the subsystem-j L_2 disturbance attenuation proof and L_2 disturbance attenuation proof of the system.

From (3.1) and (3.6), it follows

$$\boldsymbol{X}(s) = \boldsymbol{M}(s)\boldsymbol{D}(s). \tag{3.14}$$

Suppose the disturbance is $\boldsymbol{d}^j(t)$, where j is an arbitrarily selected subsystem. The Laplace transform of $\boldsymbol{d}^j(t)$ is denoted by

$$\boldsymbol{D}^j(s) := [0, ..., 0, D_j(s), 0, ..., 0]^\top \in \mathbb{C}^{m\times 1}. \tag{3.15}$$

Substituting (3.15) into (3.14), the states of subsystem o and subsystem i are

$$X_o^j(s) = M_{o,j}(s)D_j(s), \tag{3.16}$$

$$X_i^j(s) = M_{i,j}(s)D_j(s), \tag{3.17}$$

respectively. Thus, the transfer function from subsystem i to subsystem o is readily obtained by combining (3.16) and (3.17), i.e.

$$G_{i,o}^j(s) = \frac{M_{o,j}(s)}{M_{i,j}(s)}. \tag{3.18}$$

According to Definition 3.1, the network is subsystem-j L_2 disturbance attenuating if and only if

$$\left\| \frac{M_{o,j}(s)}{M_{i,j}(s)} \right\|_\infty < 1, \quad \forall o \in \mathcal{C}_j(i), \quad i, j, o \in \mathcal{V}. \tag{3.19}$$

Since subsystem j is arbitrarily selected, (3.19) can be proved for all subsystems following the above procedures. Therefore, system (3.1) is L_2 disturbance attenuating according to Definition 3.1. Hence, (3.13) is both necessary and sufficient L_2 disturbance attenuation condition for system (3.1). Thereby, Lemma 3.1 is proved. □

For a system with m subsystems, the number of disturbance attenuation conditions associated with Lemma 3.1 is $m(m-1)$. However, this number can be reduced to $2(m-1)$ by taking advantage of some properties of $\boldsymbol{M}(s)$. Specifically, as $\boldsymbol{M}(s)$ is the inverse of a non-singular tree-diagonal matrix $(s\boldsymbol{I}-\boldsymbol{F})$, it satisfies the treeangle property, which is defined in [94] and is stated as follows.

Definition 3.2 (Treeangle property)**.** *For a given tree $\Gamma(\mathcal{V},\mathcal{E})$ where $|\mathcal{V}|=m$, an m-by-m matrix $\boldsymbol{M}$ satisfies the Γ-treeangle property if*

$$M_{o,j}M_{i,i} = M_{o,i}M_{i,j}, \tag{3.20}$$

for every $j,i,o\in\mathcal{V}$ with $\mathcal{P}(i,o)\subseteq\mathcal{P}(j,o)$.

With $\boldsymbol{M}$ having treeangle property, the proof for Proposition 3.1 can be accomplished by removing some redundant conditions in Lemma 3.1.

Proof of Proposition 3.1. Note that conditions (3.9) and (3.10) are a subset of conditions (3.11) and (3.12). Therefore, the proposition is proved by showing that the redundant conditions in Lemma 3.1 can be guaranteed by (3.9) and (3.10).

Consider a pair of neighbouring subsystems $o,i\in\mathcal{N}(o)$. Suppose i is a leaf in the tree, i.e. $V(i)=1$, then the set $\mathcal{V}^{io}$ only contains one element i, i.e. $\mathcal{V}^{io}=\{i\}$. Therefore, condition (3.9) is equivalent to condition (3.11) since $j=f=i$ in this case.

For the case where i is not a leaf, there must exist at least two elements in the set $\mathcal{V}^{io}$. Suppose there exists $f\in\mathcal{V}^{io}$ satisfying (3.9). By applying treeangle property to subsystems i,o,f, we have

$$M_{o,f}M_{i,i} = M_{o,i}M_{i,f}. \tag{3.21}$$

Now arbitrarily choose another subsystem f' from the set $\mathcal{V}^{io}$, i.e. $f'\in\mathcal{V}^{io}\setminus\{f\}$. Similarly, the treeangle property for subsystems i,o,f' gives

$$M_{o,f'}M_{i,i} = M_{o,i}M_{i,f'}. \tag{3.22}$$

Combining (3.21) and (3.22), the following equation is obtained:

$$\frac{M_{o,f}}{M_{i,f}} = \frac{M_{o,f'}}{M_{i,f'}}. \tag{3.23}$$

Recalling that (3.9) holds for f, the following inequality can be obtained by substituting (3.23) into (3.9):

$$\left\| \frac{P_{o,f'}(z)}{P_{i,f'}(z)} \right\|_\infty < 1. \tag{3.24}$$

Note that f' is a subsystem arbitrarily chosen from the set $\mathcal{V}^{io} \setminus \{f\}$. Therefore, in the case where i is not a leaf, the condition (3.11) is satisfied $\forall j \in \mathcal{V}^{io}$.

To summarize, if we can find a subsystem $f \in \mathcal{V}^{io}$ satisfying (3.9), then condition (3.11) holds $\forall j \in \mathcal{V}^{io}$, irrespective of i being a leaf or otherwise. For the pair (i, o), the other half of the proof, which is to show that condition (3.12) can be implied by condition (3.10), can be readily done by following the same procedure. Furthermore, it can be proven for every pair of neighbouring subsystems $i \in \mathcal{N}(o)$. Therefore, the sufficiency of Proposition 3.1 for the conditions in Lemma 3.1 is established, and Proposition 3.1 is proved thereby. □

Sufficient conditions

The verification of disturbance attenuation using conditions (3.9) and (3.10) requires global information, which may be difficult to obtain in a large-scale interconnected system. Furthermore, checking those conditions takes at least $\mathcal{O}(m^3)$ operations, as it requires the inverse calculation of matrix $z\boldsymbol{I} - \boldsymbol{Q}$. In cases where the computational resources are limited, it is impractical to implement the online verification of these conditions. In this section, we will focus on developing disturbance attenuation conditions that depend on local information and are easily verified.

Lemma 3.2. *Consider an interconnected continuous-time system as described by* (3.1). *Let* $\boldsymbol{d}(t) = \boldsymbol{d}^j(t)$. *The inequality*

$$||G^j_{i,o}(s)||_\infty < 1 \tag{3.25}$$

holds $\forall o \in \mathcal{C}_j(i)$, $\forall i, o \in \mathcal{V}$, *if the system matrix* $\boldsymbol{F}$ *satisfies*

$$|F_{a,a}| > \sum_{\forall b \in \mathcal{N}(a)} |F_{a,b}| \quad \forall a, b \in \mathcal{V}. \tag{3.26}$$

Proof. This lemma can be proven through induction. The induction includes $h(\mathcal{T}_j)$ steps, where at step 1 inequality (3.25) is proven for subsystem o at level $L_j(o) = h(\mathcal{T}_j)$, and at step $h(\mathcal{T}_j)$ it is proven for subsystem o at level $L_j(o) = 1$.

Basis step: As the initial step for induction, we consider a subsystem $o \in \{o|L_j(o) = h(\mathcal{T}_j)\}$. Note that o can only be a leaf. Since $\boldsymbol{F}$ is a Γ-tree-diagonal matrix, the state equation of subsystem o has the form

$$\dot{x}_o(t) = F_{o,o}x_o(t) + F_{o,i}x_i(t), \tag{3.27}$$

where i is the parent of o in $\mathcal{T}_j$.

Thus, the transfer function from i to o under $\boldsymbol{d}^j(k)$ is given by

$$G^j_{i,o}(s) = \frac{F_{o,i}}{s - F_{o,o}}. \tag{3.28}$$

According to (3.4), the $\mathcal{H}_\infty$ norm of the transfer function is

$$\| G^j_{i,o}(s) \|_\infty := \sup_\omega \left| \frac{F_{o,i}}{\jmath\omega - F_{o,o}} \right|. \tag{3.29}$$

The denominator of the right-hand side of (3.29) can be bounded by

$$|\jmath\omega - F_{o,o}| \geq |F_{o,o}|. \tag{3.30}$$

Combining (3.26), (3.29), and (3.30), the inequality (3.25) is proven for subsystems $(i,o) \in \{(i,o)|L_j(o) = h(\mathcal{T}_j), o \in \mathcal{C}_j(i)\}$. Following the above procedure, (3.25) can be proven for all subsystems o at level $L_j(o) = h(\mathcal{T}_j)$.

Induction step: Assume inequality (3.25) holds from step 1 ($L_j(o) = h(\mathcal{T}_j)$) to step $\bar{k}$ ($L_j(o) = h(\mathcal{T}_j) + 1 - \bar{k}$), where $\bar{k} \in [1, h(\mathcal{T}_j) - 1]$. For the current step $\bar{k} + 1$ ($L_j(o) = h(\mathcal{T}_j) - \bar{k}$), (3.25) can be proven as follows.

For all subsystems $o \in \mathcal{S}_{lvs}(h(\mathcal{T}_j) - \bar{k})$, the state equation of o has the same form as in (3.27). Therefore, it can be proven in the same way in the basis step that (3.25) holds.

For a subsystem $o \in \mathcal{S}_{int}(h(\mathcal{T}_j) - \bar{k})$, the state equation has the following form:

$$\dot{x}_o(t) = F_{o,o}x_o(t) + F_{o,i}x_i(t) + \sum_{c\in\mathcal{C}_j(o)} F_{o,c}x_c(t), \tag{3.31}$$

where i is the parent of o.

Taking s-transform of (3.31), we have

$$(s - F_{o,o})X_o(s) = F_{o,i}X_i(s) + \sum_{c\in\mathcal{C}_j(o)} F_{o,c}X_c(s), \tag{3.32}$$

where $X_c(s)$ can be substituted by

$$X_c(s) = G^j_{o,c}(s)X_o(s). \tag{3.33}$$

Therefore, the $\mathcal{H}_\infty$ norm of the transfer function from i to o under $\boldsymbol{d}^j(t)$ can be written as

$$||G^j_{i,o}(s)||_\infty = \frac{|F_{o,i}|}{\inf_\omega \left| \jmath\omega - F_{o,o} - \sum_{c\in\mathcal{C}_j(o)} F_{o,c}G^j_{o,c}(\jmath\omega) \right|}. \tag{3.34}$$

The denominator of the transfer function satisfies

$$\left| \jmath\omega - F_{o,o} - \sum_{c\in\mathcal{C}_j(o)} F_{o,c}G^j_{o,c}(\jmath\omega) \right| \geq |\jmath\omega - F_{o,o}| - \sum_{c\in\mathcal{C}_j(o)} |F_{o,c}G^j_{o,c}(\jmath\omega)|. \tag{3.35}$$

At step $\bar{k}$ it has already been proven from the state equation of subsystem c that

$$||G^j_{o,c}(s)||_\infty < 1. \tag{3.36}$$

Therefore, the second term on the right side of (3.35) can be bounded by:

$$\sum_{c\in\mathcal{C}_j(o)} |F_{o,c}G^j_{o,c}(\jmath\omega)| \leq \sum_{c\in\mathcal{C}(o)} ||G^j_{o,c}(s)||_\infty |F_{o,c}| < \sum_{c\in\mathcal{C}(o)} |F_{o,c}|. \tag{3.37}$$

Combining (3.34), (3.35), (3.30), (3.37), and (3.26), the inequality (3.25) is proven for $(i, o) \in$

$\{o|\mathcal{S}_{int}(h(\mathcal{T}_j)-\bar{k})\cap\mathcal{C}_j(i)\}$. Following the above procedure, (3.25) can be proven for all subsystems o in the set $\mathcal{S}_{int}(h(\mathcal{T}_j)-\bar{k})$.

Since both the basis and the induction step have been performed, by mathematical induction, the inequality (3.25) holds for all levels in the tree $\mathcal{T}_j$. Thereby, the Lemma 3.2 is proved. □

The subsystem-j L_2 disturbance attenuation conclusion can be drawn for system (3.1) following Lemma 3.2, as stated in the following corollary.

Corollary 3.1. *For the continuous-time system as described in* (3.1)*, it is subsystem-j L_2 disturbance attenuating if* (3.26) *holds.*

Proposition 3.2 (Sufficient conditions for continuous-time systems)**.** *For the continuous-time system as described in* (3.1)*, it is L_2 disturbance attenuating if* (3.26) *holds.*

Proof. Subsystem-j disturbance attenuation can be proven for all subsystems $j\in\mathcal{V}$ following the procedure in Lemma 3.2 and Corollary 3.1. According to Definition 3.1, (3.26) is a sufficient disturbance attenuation condition. □

Remark: An advantage of Proposition 3.2 is that each of these conditions can be checked locally for each subsystem. This is because for each subsystem $a\in\mathcal{V}$, only neighbouring information is required to verify the sufficient L_2 disturbance attenuation condition (3.26).

Remark: Compared to the necessary and sufficient L_2 disturbance attenuation conditions (3.9) and (3.10) in Proposition 3.1, the sufficient conditions (3.26) derived in Proposition 3.2 can be verified relatively quickly. Specifically, the number of operations is $\mathcal{O}(m^2)$ for condition (3.26)[3], while it is at least $\mathcal{O}(m^3)$ for conditions (3.9) and (3.10) due to the inverse calculation of the matrix $\boldsymbol{Q}$.

Remark: Compared to conditions (3.9) and (3.10) in Proposition 3.1, the sufficient conditions (3.26) can be more easily integrated into the controller design problem, as they are essentially element-wise inequality constraints.

In the following section, disturbance attenuation conditions are derived for discrete-time systems.

[3]Condition (3.26) is verified independently by each subsystem, and it takes $\mathcal{O}(m)$ operations for each verification.

3.2 Disturbance attenuation analysis in discrete-time systems

3.2.1 Preliminaries

Similar to Definition 3.1, a disturbance attenuation definition will be proposed for a discrete-time system.

Consider a discrete-time interconnected system with m scalar subsystems, where the interconnection of the subsystems can be represented by a tree $\Gamma(\mathcal{V},\mathcal{E})$. Let $x_i(k)$ denote the state of subsystem i at time step k. Suppose the dynamics of the overall system can be described by

$$\boldsymbol{x}(k+1) = \boldsymbol{Q}\boldsymbol{x}(k) + \boldsymbol{d}(k), \tag{3.38}$$

where $\boldsymbol{x}(k) := [x_1(k),...,x_m(k)]^\top$ is the state vector; $\boldsymbol{d}(k) \in \mathbb{R}^m$ is the disturbance vector; $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ is a Γ-tree-diagonal matrix.

To assess how the impact of a disturbance in one subsystem propagates throughout the network, the following assumption is made for system (3.38).

Assumption 3.2. *A disturbance can only occur in one subsystem at any point in time in system* (3.38)*, i.e., vector* $\boldsymbol{d}(k)$ *only has one non-zero element.*

Let $\boldsymbol{d}^j(k)$ denote a disturbance vector with a single non-zero element at the j-th location. The state of the i-th subsystem under $\boldsymbol{d}^j(k)$ is denoted by $x_i^j(k)$. When a disturbance originally occurs in subsystem j and propagates to the rest of the system, the propagation direction can be represented by the edges pointing away from root j in the rooted tree $\mathcal{T}_j(\mathcal{V},\mathcal{E})$.

Similar to continuous-time case, the disturbance attenuation analysis for discrete-time systems will also be conducted under L_2 metric. To analyse disturbance propagation, the energy of the states of neighbouring subsystems will be compared. Under $\boldsymbol{d}^j(k)$, we treat each pair of neighbouring subsystems i and $o \in \mathcal{C}_j(i)$ in $\mathcal{T}_j$ as a SISO system, where the input and output are the states of the i-th and o-th subsystem, respectively. Under $\boldsymbol{d}^j(k)$, the transfer function between the two is given by:

$$H_{i,o}^j(z) := \frac{X_o^j(z)}{X_i^j(z)}, \tag{3.39}$$

where $X_i^j(z)$ and $X_o^j(z)$ are the z-transforms of $x_i^j(k)$ and $x_o^j(k)$, respectively.

The energy of the input and output will be compared for each SISO system, and this comparison is facilitated by the $\mathcal{H}_\infty$ norm of $H^j_{i,o}(z)$:

$$||H^j_{i,o}(z)||_\infty = \sup_{x^j_i \neq 0} \frac{||x^j_o||_2}{||x^j_i||_2}. \tag{3.40}$$

Given that $H^j_{i,o}(z)$ is a transfer function of an SISO system, $\left\|H^j_{i,o}(z)\right\|_\infty$ can be simplified as

$$||H^j_{i,o}(z)||_\infty := \sup_\theta |H^j_{i,o}(e^{\jmath\theta})|. \tag{3.41}$$

Similar to Definition 3.1, disturbance energy attenuation is defined for a discrete-time system as below.

Definition 3.3 (L_2 disturbance attenuation of a discrete-time tree-structured system)**.** *Consider an interconnected discrete-time system as described by* (3.38)*. Given any $\boldsymbol{d}^j(k)$, if the energy of the output state $x^j_o(k)$ is always smaller than the energy of the input state $x^j_i(k)$ for every pair of neighbouring subsystems i and $o \in \mathcal{C}_j(i)$ in $\mathcal{T}_j$, i.e.*

$$||H^j_{i,o}(z)||_\infty < 1 \ \forall o \in \mathcal{C}_j(i),\ \forall i, o \in \mathcal{V}, \tag{3.42}$$

then the system is subsystem-j L_2 disturbance attenuating. If the system is subsystem-j L_2 disturbance attenuating for all $j \in \mathcal{V}$, then the system is L_2 disturbance attenuating.

Remark: L_2 disturbance attenuation guarantees that the energy of states, which is essentially the effect of the disturbance $\boldsymbol{d}^j(k)$,[4] is always attenuated along the propagation path from the originally disturbed subsystem to the rest of the system.

3.2.2 Disturbance attenuation conditions

The aim of this section is to propose conditions on system (3.38), such that disturbance attenuation property in the sense of Definition 3.3 can be determined.

Problem 3.2. *Given system (3.38), find conditions on matrix $\boldsymbol{Q}$, such that the system is L_2 disturbance attenuating.*

[4] We assume the states are at the origin in the absence of disturbance.

Necessary and sufficient conditions

Assume system (3.38) is asymptotically stable. After z-transform, define matrix

$$\boldsymbol{P}(z) := (z\boldsymbol{I} - \boldsymbol{Q})^{-1}, \tag{3.43}$$

where $\boldsymbol{I}$ is the identity matrix. Assume $P_{i,j}(z) \neq 0, \forall i, j \in \mathcal{V}$ [5].

Proposition 3.3 (Necessary and sufficient conditions for discrete-time systems)**.** *Consider an interconnected discrete-time system as described by (3.38). The system is L_2 disturbance attenuating in the sense of Definition 3.3 if and only if $\forall i \in \mathcal{N}(o)$, $\exists f \in \mathcal{V}^{io}$ and $\exists \bar{f} \in \mathcal{V}^{oi}$ such that*

$$\left\| \frac{P_{o,f}(z)}{P_{i,f}(z)} \right\|_\infty < 1, \tag{3.44}$$

$$\left\| \frac{P_{i,\bar{f}}(z)}{P_{o,\bar{f}}(z)} \right\|_\infty < 1. \tag{3.45}$$

Proof. To prove Proposition 3.3, the following lemma is introduced.

Lemma 3.3. *System (3.38) is L_2 disturbance attenuating if and only if for each pair of neighbouring subsystems $i \in \mathcal{N}(o)$,*

$$\left\| \frac{P_{o,j}(z)}{P_{i,j}(z)} \right\|_\infty < 1 \tag{3.46}$$

and

$$\left\| \frac{P_{i,\bar{j}}(z)}{P_{o,\bar{j}}(z)} \right\|_\infty < 1 \tag{3.47}$$

hold for all $j \in \mathcal{V}^{io}$ and $\bar{j} \in \mathcal{V}^{oi}$.

Proof. Notice that conditions (3.46) and (3.47) are equivalent to the following condition

$$\left\| \frac{P_{o,j}(z)}{P_{i,j}(z)} \right\|_\infty < 1, \quad o \in \mathcal{C}_j(i), \quad \forall i, j, o \in \mathcal{V}. \tag{3.48}$$

Thus, the proof for conditions (3.46) and (3.47) can be transformed to proving that (3.48) is a necessary and sufficient condition.

[5] It is reasonable to assume $P_{i,j}(z) \neq 0$, otherwise the graph is disconnected.

The proof includes two steps: the subsystem-j L_2 disturbance attenuation proof and L_2 disturbance attenuation proof of the system.

From (3.38) and (3.43), it follows

$$\boldsymbol{X}(z) = \boldsymbol{P}(z)\boldsymbol{D}(z). \tag{3.49}$$

Suppose the disturbance is $\boldsymbol{d}^j(k)$, for which the z-transform is denoted by

$$\boldsymbol{D}_j^s(z) := [0, ..., 0, D_j(z), 0, ..., 0]^\top \in \mathbb{C}^{m\times 1}. \tag{3.50}$$

Substituting (3.50) into (3.49), the states of subsystem o and subsystem i are

$$X_o^j(z) = P_{o,j}(z)D_j(z), \tag{3.51}$$

$$X_i^j(z) = P_{i,j}(z)D_j(z), \tag{3.52}$$

respectively. Thus, the transfer function from subsystem i to subsystem o is readily obtained by combining (3.51) and (3.52), i.e.

$$H_{i,o}^j(z) = \frac{P_{o,j}(z)}{P_{i,j}(z)}. \tag{3.53}$$

According to Definition 3.3, the network is subsystem-j L_2 disturbance attenuating if and only if

$$\left\| \frac{P_{o,j}(z)}{P_{i,j}(z)} \right\|_\infty < 1, \quad \forall o \in \mathcal{C}_j(i), \quad i, j, o \in \mathcal{V}. \tag{3.54}$$

For the system to be L_2 disturbance attenuating, (3.54) has to be satisfied for all j. Hence, (3.48) is both necessary and sufficient L_2 disturbance attenuation condition for system (3.38). Thereby, Lemma 3.3 is proved. □

For a system with m subsystems, the number of disturbance attenuation conditions associated with Lemma 3.3 is $m(m-1)$. However, this number can be reduced to $2(m-1)$ by taking advantage of some properties of $\boldsymbol{P}(z)$. Specifically, as $\boldsymbol{P}(z)$ is the inverse of a non-singular tree-diagonal matrix $(z\boldsymbol{I} - \boldsymbol{Q})$, it satisfies the treeangle property, which is defined in [94] and is stated

as follows.

With $\boldsymbol{P}$ having treeangle property, the proof for Proposition 3.3 can be accomplished by removing some redundant conditions in Lemma 3.3.

Proof of Proposition 3.3. Note that conditions (3.44) and (3.45) are a subset of conditions (3.46) and (3.47). Therefore, the proposition is proved by showing that the redundant conditions in Lemma 3.3 can be guaranteed by (3.44) and (3.45).

Consider a pair of neighbouring subsystems $o, i \in \mathcal{N}(o)$. Suppose i is a leaf in the tree, i.e. $V(i) = 1$, then the set $\mathcal{V}^{io}$ only contains one element i, i.e. $\mathcal{V}^{io} = \{i\}$. Therefore, condition (3.44) is equivalent to condition (3.46) since $j = f = i$ in this case.

For the case where i is not a leaf, there must exist at least two elements in the set $\mathcal{V}^{io}$. Suppose there exists $f \in \mathcal{V}^{io}$ satisfying (3.44). By applying treeangle property to subsystems i, o, f, we have

$$P_{o,f}P_{i,i} = P_{o,i}P_{i,f}. \tag{3.55}$$

Now arbitrarily choose another subsystem f' from the set $\mathcal{V}^{io}$, i.e. $f' \in \mathcal{V}^{io} \setminus \{f\}$. Similarly, the treeangle property for subsystems i, o, f' gives

$$P_{o,f'}P_{i,i} = P_{o,i}P_{i,f'}. \tag{3.56}$$

Combining (3.55) and (3.56), the following equation is obtained:

$$\frac{P_{o,f}}{P_{i,f}} = \frac{P_{o,f'}}{P_{i,f'}}. \tag{3.57}$$

Recalling that (3.44) holds for f, the following inequality can be obtained by substituting (3.57) into (3.44):

$$\left\| \frac{P_{o,f'}(z)}{P_{i,f'}(z)} \right\|_\infty < 1. \tag{3.58}$$

Note that f' is a subsystem arbitrarily chosen from the set $\mathcal{V}^{io} \setminus \{f\}$. Therefore, in the case where i is not a leaf, the condition (3.46) is satisfied $\forall j \in \mathcal{V}^{io}$.

To summarize, if we can find a subsystem $f \in \mathcal{V}^{io}$ satisfying (3.44), then condition (3.46) holds $\forall j \in \mathcal{V}^{io}$, irrespective of i being a leaf or otherwise. For the pair (i, o), the other half of the proof, which is to show that condition (3.47) can be implied by condition (3.45), can be readily

done by following the same procedure. Furthermore, it can be proven for every pair of neighbouring subsystems $i \in \mathcal{N}(o)$. Therefore, the sufficiency of Proposition 3.3 for the conditions in Lemma 3.3 is established, and Proposition 3.3 is proved thereby. □

□

Sufficient conditions

It can be easily observed that necessary and sufficient conditions (3.44) and (3.45) have similar problems as their continuous-time counterparts (3.9) and (3.10). Specifically, conditions (3.44) and (3.45) cannot be directly used for distributed control, as global information is required. Moreover, at least $\mathcal{O}(m^3)$ operations are required for the verification of conditions (3.44) and (3.45) due to the calculation of the inverse of $z\boldsymbol{I} - \boldsymbol{Q}$, which may not be practical for large-scale systems. Hence, it is desirable to develop conditions which are suitable for distributed control and more computationally advanced.

Lemma 3.4. *Consider an interconnected discrete-time system as described by* (3.38)*. Let* $\boldsymbol{d}(k) = \boldsymbol{d}^j(k)$*. The inequality*

$$||H^j_{i,o}(z)||_\infty < 1 \tag{3.59}$$

holds $\forall o \in \mathcal{C}_j(i)$*,* $\forall i, o \in \mathcal{V}$*, if the system matrix* $\boldsymbol{Q}$ *satisfies*

$$|1 - |Q_{a,a}|| > \sum_{\forall b \in \mathcal{N}(a)} |Q_{a,b}| \quad \forall a, b \in \mathcal{V}. \tag{3.60}$$

Proof. This lemma can be proven through induction. The induction includes $h(\mathcal{T}_j)$ steps, where at step 1 inequality (3.59) is proven for subsystem o at level $L_j(o) = h(\mathcal{T}_j)$, and at step $h(\mathcal{T}_j)$ it is proven for subsystem o at level $L_j(o) = 1$.

Basis step: As the initial step for induction, we consider a subsystem $o \in \{o | L_j(o) = h(\mathcal{T}_j)\}$. Note that o can only be a leaf. Since $\boldsymbol{Q}$ is a Γ-tree-diagonal matrix, the state equation of subsystem o has the form

$$x_o(k+1) = Q_{o,o}x_o(k) + Q_{o,i}x_i(k), \tag{3.61}$$

where i is the parent of o in $\mathcal{T}_j$.

Thus, the transfer function from i to o under $\boldsymbol{d}^j(k)$ is given by

$$H^j_{i,o}(z) = \frac{Q_{o,i}}{z - Q_{o,o}}. \tag{3.62}$$

According to (3.41), the $\mathcal{H}_\infty$ norm of the transfer function is

$$\| H^j_{i,o}(z) \|_\infty := \sup_\theta \left| \frac{Q_{o,i}}{e^{j\theta} - Q_{o,o}} \right|. \tag{3.63}$$

The denominator of the right-hand side of (3.63) can be bounded by

$$\left| e^{j\theta} - Q_{o,o} \right| \geq \left| |e^{j\theta}| - |Q_{o,o}| \right| = |1 - |Q_{o,o}|| \,. \tag{3.64}$$

Given condition (3.60), (3.64) can be further bounded by

$$|1 - |Q_{o,o}|| > \sum_{\forall v \in \mathcal{N}(o)} |Q_{o,v}| = |Q_{o,i}|. \tag{3.65}$$

Combining (3.63), (3.64), and (3.65), the inequality (3.59) is proven for subsystems $(i,o) \in \{(i,o)|L_j(o) = h(\mathcal{T}_j), o \in \mathcal{C}_j(i)\}$. Following the above procedure, (3.59) can be proven for all subsystems o at level $L_j(o) = h(\mathcal{T}_j)$.

Induction step: Assume inequality (3.59) holds from step 1 ($L_j(o) = h(\mathcal{T}_j)$) to step $\bar{k}$ ($L_j(o) = h(\mathcal{T}_j) + 1 - \bar{k}$), where $\bar{k} \in [1, h(\mathcal{T}_j) - 1]$. For the current step $\bar{k}+1$ ($L_j(o) = h(\mathcal{T}_j) - \bar{k}$), (3.59) can be proven as follows.

Let the set of leaves and the set of interior (non-leaf) subsystems at level l be denoted by $\mathcal{S}_{lvs}(l) := \{o|L_j(o) = l, V(o) = 1\}$ and $\mathcal{S}_{int}(l) := \{o|L_j(o) = l, V(o) > 1\}$, respectively.

For all subsystems $o \in \mathcal{S}_{lvs}(h(\mathcal{T}_j) - \bar{k})$, the state equation of o has the same form as in (3.61). Therefore, it can be proven in the same way in the basis step that (3.59) holds.

For a subsystem $o \in \mathcal{S}_{int}(h(\mathcal{T}_j) - \bar{k})$, the state equation has the following form:

$$x_o(k+1) = Q_{o,o}x_o(k) + Q_{o,i}x_i(k) + \sum_{c \in \mathcal{C}_j(o)} Q_{o,c}x_c(k), \tag{3.66}$$

where i is the parent of o.

Taking z-transform of (3.66), we have

$$(z - Q_{o,o})X_o(z) = Q_{o,i}X_i(z) + \sum_{c \in \mathcal{C}_j(o)} Q_{o,c}X_c(z), \tag{3.67}$$

where $X_c(z)$ can be substituted by

$$X_c(z) = H^j_{o,c}(z)X_o(z). \tag{3.68}$$

Therefore, the $\mathcal{H}_\infty$ norm of the transfer function from i to o under $\boldsymbol{d}^j$ can be written as

$$||H^j_{i,o}(z)||_\infty = \frac{|Q_{o,i}|}{\inf\limits_\theta \left| e^{\jmath\theta} - Q_{o,o} - \sum\limits_{c \in \mathcal{C}_j(o)} Q_{o,c}H^j_{o,c}(e^{\jmath\theta}) \right|}. \tag{3.69}$$

The denominator of the transfer function satisfies

$$\left| e^{\jmath\theta} - Q_{o,o} - \sum_{c \in \mathcal{C}_j(o)} Q_{o,c}H^j_{o,c}(e^{\jmath\theta}) \right| \quad \geq \quad \left| \left| e^{\jmath\theta} - Q_{o,o} \right| - | \sum_{c \in \mathcal{C}_j(o)} Q_{o,c}H^j_{o,c}(e^{\jmath\theta})| \right|, \tag{3.70}$$

where $\left| e^{\jmath\theta} - Q_{o,o} \right|$ has been proven in (3.64) and (3.65) to be bounded by $\sum\limits_{\forall v \in \mathcal{N}(o)} |Q_{o,v}|$. Note that for subsystem $o \in \mathcal{S}_{int}(h(\mathcal{T}_j) - \bar{k})$, the neighbour set $\mathcal{N}(o)$ contains the parent vertex set and children vertex set, i.e. $\mathcal{N}(o) = \{i | o \in \mathcal{C}_j(i)\} \cup \{c | c \in \mathcal{C}_j(o)\}$. Thus, the bound for $\left| e^{\jmath\theta} - Q_{o,o} \right|$ in this case is given by

$$\left| e^{\jmath\theta} - Q_{o,o} \right| > \sum_{\forall v \in \mathcal{N}(o)} |Q_{o,v}| = |Q_{o,i}| + \sum_{\forall c \in \mathcal{C}_j(o)} |Q_{o,c}|. \tag{3.71}$$

At step $\bar{k}$ it has already been proven from the state equation of subsystem c that

$$||H^j_{o,c}(z)||_\infty < 1. \tag{3.72}$$

Therefore, the second term on the right side of (3.70) can be bounded by:

$$\left| \sum_{c \in \mathcal{C}(o)} Q_{o,c}H^j_{o,c}(e^{\jmath\theta}) \right| \leq \sum_{c \in \mathcal{C}(o)} ||H^j_{o,c}(z)||_\infty |Q_{o,c}| < \sum_{c \in \mathcal{C}(o)} |Q_{o,c}|. \tag{3.73}$$

Combining (3.69), (3.70), (3.71), and (3.73), the inequality (3.59) is proven for $(i,o) \in \{o|\mathcal{S}_{int}(h(\mathcal{T}_j) - \bar{k}) \cap \mathcal{C}_j(i)\}$. Following the above procedure, (3.59) can be proven for all subsystems o in the set $\mathcal{S}_{int}(h(\mathcal{T}_j) - \bar{k})$.

Since both the basis and the induction step have been performed, by mathematical induction, the inequality (3.59) holds for all levels in the tree $\mathcal{T}_j$. Thereby, the Lemma 3.4 is proved. □

The subsystem-j L_2 disturbance attenuation conclusion can be drawn for system (3.38) following Lemma 3.4, as stated in the following corollary.

Corollary 3.2. *For the system as described in* (3.38)*, it is subsystem-j L_2 disturbance attenuating in the sense of Definition 3.3 if* (3.60) *holds.*

Proposition 3.4 (Sufficient conditions for discrete-time systems)**.** *For the discrete-time system as described in* (3.38)*, it is L_2 disturbance attenuating in the sense of Definition 3.3 if* (3.60) *holds.*

Proof. Subsystem-j disturbance attenuation can be proven for all subsystems $j \in \mathcal{V}$ following the procedure in Lemma 3.4 and Corollary 3.2. According to Definition 3.3, (3.60) is a sufficient disturbance attenuation condition. □

3.3 Conclusions

This chapter has focused on proposing disturbance attenuation conditions for a network with m scalar subsystems. Continuous-time systems are first discussed, and then similar results are obtained for discrete-time systems.

The determination of disturbance attenuation is facilitated by comparing the states of every pair of neighbouring subsystems, and L_2 norm metric is adopted, implying energy attenuation during disturbance propagation. Using frequency-domain techniques, $2(m-1)$ necessary and sufficient disturbance attenuation conditions are proposed. However, it is found that these conditions are based on global information, and the verification for large-scale systems may be difficult due to computational complexity (the number of operations is at least $\mathcal{O}(m^3)$). To address this problem, sufficient conditions are proposed by making further use of the tree topology. Only neighbouring information is required for the verification of these sufficient conditions, which enables the local check by each subsystem. Moreover, the computational operations for the verification is reduced

to $\mathcal{O}(m^2)$. These two advantages of the sufficient conditions enable them to be used for large-scale interconnected systems with distributed control policy.

The derived conditions can be used to determine disturbance attenuation for given systems of the described type. However, it is still unclear how to develop a controller for an open-loop system such that the resulted closed-loop system attenuates disturbance. In Chapter 4, a controller synthesis method will be proposed, in which the sufficient conditions derived in Chapter 3 will be incorporated to guarantee disturbance attenuation.

This page intentionally left blank.

Chapter 4

Disturbance attenuating controller design and tree approximation

Although the determination of disturbance attenuation has been accomplished by the conditions proposed in Chapter 3, how to develop a controller to guarantee disturbance attenuation is so far still an open question. Hence, the first goal of this chapter is to propose a controller synthesis method for a tree-structured system such that the resulting closed-loop system satisfies the following requirements: (a) the closed-loop system also has a tree topology; (b) the overall system is asymptotically stable; (c) disturbance is attenuating when propagating in the network.

The second half of this chapter deals with tree approximation problems. Tree-structured networks have been focused so far, but non-tree networks widely exist in many real-world applications. To broaden the scope of this research, it is important to discuss how to approximate an arbitrarily connected network by a tree, so the preceding results can also be applied.

Corresponding to the two problems above, this chapter includes 2 sections. In Section 4.1, two controller synthesis methods are proposed for a continuous-time and discrete-time system, respectively. In Section 4.2, a tree approximation method is developed.

4.1 Disturbance attenuating controller synthesis method

Based on the sufficient conditions proposed in Chapter 3, a disturbance-attenuating controller synthesis method will be introduced, and both centralised and distributed design will be discussed in this section.

The main content of this chapter has been submitted as a journal paper [93] (under review).

4.1.1 Controller synthesis for continuous-time systems

Problem formulation

Consider an interconnected continuous-time system with m first-order subsystems and ν inputs. Suppose the open-loop dynamics are captured by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{H}\boldsymbol{x}(t) + \boldsymbol{J}\boldsymbol{u}(t) + \boldsymbol{d}(t), \tag{4.1}$$

where system matrix $\boldsymbol{H} \in \mathbb{R}^{m\times m}$ is a Γ-tree-diagonal matrix; $\boldsymbol{J} \in \mathbb{R}^{m\times\nu}$ is an input matrix. The aim is to find a linear state feedback controller

$$\boldsymbol{u}(t) = \boldsymbol{R}\boldsymbol{x}(t), \tag{4.2}$$

where $\boldsymbol{R} \in \mathbb{R}^{\nu\times m}$ is the feedback gain, such that the resulting closed-loop system (3.1)[1]

1. remains a tree topology,
2. is asymptotically stable, and
3. attenuates disturbance in the sense of Definition 3.1.

These requirements are mathematically described by the following problem formulation.

Problem 4.1 (Controller synthesis for continuous-time systems)**.** *The objective is to find a controller* (4.2) *for a tree structured continuous-time system* (4.1) *such that the closed-loop system* (3.1) *satisfies*

(a) $\boldsymbol{J}\boldsymbol{R} \circ S(\Gamma) = \boldsymbol{J}\boldsymbol{R}$,

(b) $\exists \boldsymbol{P} = \boldsymbol{P}^\top > 0$ *such that* $\boldsymbol{F}^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{F} < 0$, *and*

(c) $|F_{a,a}| > \sum_{\forall b\in\mathcal{N}(a)} |F_{a,b}|$, $\forall a, b \in \mathcal{V}$.

[1] It implies that $\boldsymbol{F} = \boldsymbol{H} + \boldsymbol{J}\boldsymbol{R}$.

Centralised controller synthesis

The three conditions in Problem 4.1 can be seen as the minimum requirements on the controller to be developed. Depending on the specific requirements of the system, other conditions can be added, and various types of control objectives can be selected to finalize the design process. However, this section mainly focuses on how to design a controller such that the overall system has the property of disturbance attenuation in the sense of Definition 3.1, while guaranteeing asymptotic stability with tree topology information exchange. Thereby, we will restrict ourselves to these three requirements (a)-(c), and a convex objective function $\mathcal{J}(\boldsymbol{R})$ will be used for discussion.

The control design is formulated as an optimisation problem with the decision variable $\boldsymbol{R}$:

$$\min_{\boldsymbol{R}} \quad \mathcal{J}(\boldsymbol{R}) \tag{4.3a}$$

$$\text{s.t.} \quad \boldsymbol{F} = \boldsymbol{H} + \boldsymbol{J}\boldsymbol{R}, \tag{4.3b}$$

$$\boldsymbol{J}\boldsymbol{R} \circ S(\Gamma) = \boldsymbol{J}\boldsymbol{R}, \tag{4.3c}$$

$$\boldsymbol{P} = \boldsymbol{P}^\top > 0, \quad \boldsymbol{F}^\top \boldsymbol{P} + \boldsymbol{P}\boldsymbol{F} < 0, \tag{4.3d}$$

$$|F_{a,a}| > \sum_{\forall b \in \mathcal{N}(a)} |F_{a,b}|, \quad \forall a, b \in \mathcal{V}. \tag{4.3e}$$

Note that the optimisation problem (4.3) is non-convex, and the structural requirement on $\boldsymbol{R}$ generally makes it even more difficult to solve [95]. One possible approach is to employ augmented Lagrangian method. For more details, we refer the reader to [95, 96], in which augmented Lagrangian method is used to handle the structural constraints in distributed controller design problems. Noticing that the non-convexity of our problem is due to constraint (4.3e), in this research we will tackle this problem by considering a convex approximation of the original problem. By observation, constraint (4.3e) can be expanded as

$$0 > -F_{a,a} + \sum_{\forall b \in \mathcal{N}(a)} |F_{a,b}|, \tag{4.4}$$

and

$$0 > F_{a,a} + \sum_{\forall b \in \mathcal{N}(a)} |F_{a,b}|. \tag{4.5}$$

Note that (4.4) is non-convex, whilst (4.5) is convex, which is proven in the following lemma and corollary.

Lemma 4.1. *Given any two m-dimensional matrices* $\boldsymbol{F}^1$ *and* $\boldsymbol{F}^2$ *satisfying* (4.5)*, matrix* $\boldsymbol{F}^3 := \theta\boldsymbol{F}^1 + (1-\theta)\boldsymbol{F}^2$ *also satisfies* (4.5)*.*

Proof. The following inequality can be readily derived:

$$\begin{aligned} F^3_{a,a} + \sum_{\forall b\in\mathcal{N}(a)} |F^3_{a,b}| &= \theta F^1_{a,a} + (1-\theta)F^2_{a,a} + \sum_{\forall b\in\mathcal{N}(a)} |\theta F^1_{a,b} + (1-\theta)F^2_{a,b}| \\ &\leq \theta F^1_{a,a} + (1-\theta)F^2_{a,a} + \sum_{\forall b\in\mathcal{N}(a)} \theta|F^1_{a,b}| + (1-\theta)\sum_{\forall b\in\mathcal{N}(a)} |F^2_{a,b}| \\ &\leq \theta(F^1_{a,a} + \sum_{\forall b\in\mathcal{N}(a)} |F^1_{a,b}|) + (1-\theta)(F^2_{a,a} + \sum_{\forall b\in\mathcal{N}(a)} |F^2_{a,b}|) \\ &< 0. \end{aligned}$$

□

Corollary 4.1. *Constraint* (4.5) *is convex.*

We use $\epsilon < 0$ to indicate the desired convergence rate. Therefore, constraint (4.3e) is replaced by

$$\epsilon > F_{a,a} + \sum_{\forall b\in\mathcal{N}(a)} |F_{a,b}|, \forall a, b \in \mathcal{V}, \tag{4.6}$$

which is a convex subset of (4.5).

Note that constraint (4.6) guarantees disturbance attenuation of the system, since it is a sufficient condition for (4.3e). The optimisation problem becomes convex after this substitution. Moreover, from Gershgorin disk theorem, constraint (4.6) automatically enforces the eigenvalues of $\boldsymbol{F}$ to the left half s-plane, guaranteeing Lyapunov stability of the resulting closed-loop system. Thereby, constraint (4.3d), which is included to ensure stability, can now be eliminated.

Distributed controller synthesis

Centralised controller design may not be possible for large-scale networks due to limited communication and computation resources, thus a distributed synthesis method is proposed under the following assumption.

Assumption 4.1. *Each subsystem only has access to the information from itself and its neighbouring subsystems.*

To realise distributed design, the objective function $J(\boldsymbol{R})$ should be able to express as a sum of elements depending on local information. For example, $J(\boldsymbol{R})$ can be $\sum_{a=1}^{\nu}\sum_{b=1}^{m}|R_{a,b}|$ or the Frobenius norm $||\boldsymbol{R}||_F$, both of which are approximations of cardinality of $\boldsymbol{R}$ [97, 98]. Here, we consider Frobenius norm as an example to realise distributed design. Note that minimising $||\boldsymbol{R}||_F$ is equivalent to minimising $||\boldsymbol{R}||_F^2$, thus the objective is given as follows:

$$\min_{\boldsymbol{R}} \qquad ||\boldsymbol{R}||_F^2. \tag{4.7}$$

By breaking up the centralised optimisation problem (4.7), (4.3a), (4.3b), (4.6) to sub-problems, subsystem controllers can be designed locally. Given (4.1), the dynamics of subsystem i can be captured by

$$\dot{x}_i(t) = \sum_{j\in(\mathcal{N}(i)\cup\{i\})} H_{i,j}x_j(t) + \sum_{a\in\mathcal{U}_i} J_{i,a}u_a(t) + d_i(t), \tag{4.8}$$

where $\mathcal{U}_i$ is the set of inputs indices of subsystem i. An input u_a is termed as a *public input*, if it affects more than one subsystem, i.e. $a \in (\mathcal{U}_i \cap \mathcal{U}_j \cap ... \cap \mathcal{U}_g), i \neq j \neq ... \neq g$. Let $\mathcal{X}(a) \coloneqq \{i, j, ..., g\}$ be the set of indices of subsystems that share input u_a[2]. Let $\mathcal{U}_i^p$ denote the set of public inputs indices of subsystem i, i.e. $\mathcal{U}_i^p \coloneqq \{a | i \in \mathcal{X}(a)\}$. On the other hand, if an input u_a affects one subsystem only, then it is termed as a *private input*. The set of private inputs indices of subsystem i is given by $\mathcal{U}_i \setminus \mathcal{U}_i^p$. Thus, state equation (4.8) can be reposed in terms of public and private inputs as follows:

$$\dot{x}_i(t) = \sum_{j\in(\mathcal{N}(i)\cup\{i\})} H_{i,j}x_j(t) + \sum_{a\in\mathcal{U}_i^p} J_{i,a}u_a(t) + \sum_{a\in\mathcal{U}_i\setminus\mathcal{U}_i^p} J_{i,a}u_a(t) + d_i(t). \tag{4.9}$$

By taking an insight of the optimisation problem, one can find that the only coupling between subsystems comes from the feedback gains associated with public inputs. Hence, to decompose the problem, we introduce new inputs $\tilde{u}_{a_i}, \tilde{u}_{a_j}, ..., \tilde{u}_{a_g}$, which are local versions of public input u_a

[2] $\mathcal{X}(a)$ is defined exclusively for public inputs. Therefore, it is assumed that $\mathcal{X}(a)$ has at least two elements when it is used.

in subsystems $i, j, ..., g$, respectively. Note that

$$\tilde{u}_{a_i} = \tilde{u}_{a_j} = ... = \tilde{u}_{a_g}. \tag{4.10}$$

Let $\boldsymbol{card}(\cdot)$ denote the cardinality of a set. Construct $\tilde{\boldsymbol{u}} \in \mathbb{R}^{(\sum_{i=1}^{m} \boldsymbol{card}(\mathcal{U}_i))\times 1}$ from the private inputs and local versions of public inputs of each subsystem. Corresponding to $\mathcal{U}_i^p$, denote the set of local versions of public inputs indices of subsystem i in vector $\tilde{\boldsymbol{u}}$ by $\tilde{\mathcal{U}}_i^p$. Similarly, the set of inputs indices in $\tilde{\boldsymbol{u}}$ corresponding to the private inputs in $\mathcal{U}_i$ is given by $\tilde{\mathcal{U}}_i \setminus \tilde{\mathcal{U}}_i^p$.

The state equation (4.1) can be rewritten in terms of $\tilde{\boldsymbol{u}}$ as in

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{H}\boldsymbol{x}(t) + \tilde{\boldsymbol{J}}\tilde{\boldsymbol{u}}(t) + \boldsymbol{d}(t), \tag{4.11}$$

where $\tilde{\boldsymbol{B}} \in \mathbb{R}^{m\times\sum_{i=1}^{m} \boldsymbol{card}(\mathcal{U}_i)}$ is the input matrix, and (4.10) holds for $\tilde{\boldsymbol{u}}$.

The control law for (4.11) is given by

$$\tilde{\boldsymbol{u}}(t) = \tilde{\boldsymbol{R}}x(t), \tag{4.12}$$

where $\tilde{\boldsymbol{R}} \in \mathbb{R}^{(\sum_{i=1}^{m} \boldsymbol{card}(\mathcal{U}_i))\times m}$ is the state feedback matrix.

$$\begin{aligned}
\min_{\tilde{\boldsymbol{R}}} \quad & \sum_{i=1}^{m} \tilde{\mathcal{J}}_i(\tilde{\boldsymbol{R}}) && \text{(4.13a)}\\
\text{subject to} \quad & \tilde{\boldsymbol{F}}_{(i,:)} = \boldsymbol{H}_{(i,:)} + \tilde{\boldsymbol{J}}_{(i,:)}\tilde{\boldsymbol{R}}, && \text{(4.13b)}\\
& (\tilde{\boldsymbol{J}}_{(i,:)}\tilde{\boldsymbol{R}}) \circ S_{(i,:)}(\Gamma) = \tilde{\boldsymbol{J}}_{(i,:)}\tilde{\boldsymbol{R}}, && \text{(4.13c)}\\
& \epsilon > F_{a,a} + \sum_{\forall b\in\mathcal{N}(a)} |F_{a,b}|, && \text{(4.13d)}\\
& \tilde{\boldsymbol{R}}_{(a_i,:)} = \tilde{\boldsymbol{R}}_{(a_j,:)},\ \forall i, j \in \mathcal{X}(a), && \text{(4.13e)}
\end{aligned}$$

where $i = 1, 2, ...m$, and

$$\tilde{\mathcal{J}}_i(\tilde{\boldsymbol{R}}) := \sum_{a_i\in\tilde{\mathcal{U}}_i^p} \frac{1}{\boldsymbol{card}(\mathcal{X}(a))} \tilde{\boldsymbol{R}}_{(a_i,:)}\tilde{\boldsymbol{R}}_{(a_i,:)}^{\top} + \sum_{a\in\tilde{\mathcal{U}}_i\setminus\tilde{\mathcal{U}}_i^p} \tilde{\boldsymbol{R}}_{(a,:)}\tilde{\boldsymbol{R}}_{(a,:)}^{\top}.$$

Note that (4.13e) is the consensus constraint corresponding to the coupling (4.10) between subsystems.

Define a cost function for subsystem i as follows:

$$\hat{\mathcal{J}}_i(\tilde{\boldsymbol{R}}) := \sum_{b\in(\mathcal{N}(i)\cup\{i\})}\Big(\sum_{a_i\in\tilde{\mathcal{U}}_i^p}\frac{1}{\boldsymbol{card}(\mathcal{X}(a))}\tilde{R}_{a_i,b}^2+\sum_{a\in\tilde{\mathcal{U}}_i\setminus\tilde{\mathcal{U}}_i^p}\tilde{R}_{a,b}^2\Big). \tag{4.14}$$

Note that problem (4.13a)-(4.13e) is still described in terms of global information. To facilitate distributed design under Assumption 4.1, the following proposition will show that it only requires local information to solve this problem.

Proposition 4.1. *The optimisation problem (4.13a)-(4.13e) is equivalent to*

$$\min_{\tilde{R}} \quad \sum_{i=1}^{m}\hat{\mathcal{J}}_i(\tilde{\boldsymbol{R}}) \tag{4.15a}$$

$$\textit{subject to} \quad \tilde{F}_{i,b} = H_{i,b}+\sum_{a\in\tilde{\mathcal{U}}(i)}\tilde{J}_{i,a}\tilde{R}_{a,b},\quad b\in(\mathcal{N}(i)\cup\{i\}), \tag{4.15b}$$

$$\tilde{R}_{a,b}=0, a\in\tilde{\mathcal{U}}_i, b\notin(\mathcal{N}(i)\cup\{i\}), \tag{4.15c}$$

$$\tilde{R}_{a_i,b}=0, i\in\mathcal{X}(a),\quad b\notin\bigcap_{j\in\mathcal{X}(a)}(\mathcal{N}(j)\cup\{j\}), \tag{4.15d}$$

$$\epsilon > F_{a,a}+\sum_{\forall b\in\mathcal{N}(a)}|F_{a,b}| \tag{4.15e}$$

$$\tilde{R}_{(a_i,b)}=\tilde{R}_{(a_j,b)},\quad i,j\in\mathcal{X}(a),\quad b\in\bigcap_{j\in\mathcal{X}(a)}(\mathcal{N}(j)\cup\{j\}), \tag{4.15f}$$

for $i=1,2,...,m$.

Proof. The proof includes three parts. Specifically, the first two parts show the equivalence of the constraints (4.13b), (4.13c), (4.13e) and (4.15b)-(4.15d), (4.15f), and the last part proves the equivalence of the objectives $\tilde{\mathcal{J}}_i(\tilde{\boldsymbol{R}})$ and $\hat{\mathcal{J}}_i(\tilde{\boldsymbol{R}})$.

First, suppose a matrix $\tilde{\boldsymbol{R}}$ satisfies conditions (4.13b), (4.13c), (4.13e). It will show that conditions (4.15b)-(4.15d), (4.15f) must be satisfied for $\tilde{\boldsymbol{R}}$ as well.

Proof of $\tilde{R}$ satisfying (4.15c)**:**

The j-th element of the one-dimensional vector $\tilde{\boldsymbol{J}}_{(i,:)}\tilde{\boldsymbol{R}}$ is given by

$$(\tilde{\boldsymbol{J}}_{(i,:)}\tilde{\boldsymbol{R}})_j := \sum_{a=1}^{\sum_{h=1}^{m} \boldsymbol{card}(\mathcal{U}_h)} \tilde{J}_{i,a}\tilde{R}_{a,j}, \quad j = 1,2,...,m. \tag{4.16}$$

Note that $\tilde{J}_{i,a} = 0$ if $a \notin \tilde{\mathcal{U}}_i$. Thus, (4.16) is equivalent to

$$\tilde{\boldsymbol{J}}_{(i,:)}\tilde{\boldsymbol{R}}_{(:,j)} := \sum_{a\in\tilde{\mathcal{U}}_i} \tilde{J}_{i,a}\tilde{R}_{a,j}, \quad j = 1,2,...,m. \tag{4.17}$$

Also, notice that the j-th element of vector $S_{(i,:)}(\Gamma)$ is zero, if j is not a neighbour of i, i.e.

$$S(\Gamma)_{(i,j)} = 0, \quad \forall j \notin (\mathcal{N}(i) \cup \{i\}). \tag{4.18}$$

Combining (4.13c), (4.17), and (4.18), condition (4.15c) can be guaranteed.

Proof of $\tilde{R}$ satisfying (4.15b)**:** From (4.15c) and (4.13b), constraint (4.15b) is satisfied.

Proof of $\tilde{R}$ satisfying (4.15d)**:** Given $i, j \in \mathcal{X}(a)$, the following equations should be satisfied based on (4.15c):

$$\tilde{R}_{a_i,b} = 0, \quad b \notin (\mathcal{N}(i) \cup \{i\}), \tag{4.19}$$

$$\tilde{R}_{a_j,b} = 0, \quad b \notin (\mathcal{N}(j) \cup \{j\}). \tag{4.20}$$

Combining (4.19), (4.20), and (4.13e), condition (4.15d) is satisfied.

Proof of $\tilde{R}$ satisfying (4.15f)**:** Notice that (4.15f) is a subset of (4.13e), thus it holds naturally.

Second, the other direction of the proof is given as follows.

Suppose $\tilde{\boldsymbol{R}}$ satisfies conditions (4.15b)-(4.15d), and (4.15f).

Proof of $\tilde{R}$ satisfying (4.13c)**:** Combining (4.17) and (4.15c), condition (4.13c) can be obtained.

Proof of $\tilde{R}$ satisfying (4.13b)**:** From (4.13c), it holds that

$$(\boldsymbol{H}_{(i,:)} + \tilde{\boldsymbol{J}}_{(i,:)}\tilde{\boldsymbol{R}})_b = 0, \quad b \notin (\mathcal{N}(i) \cup \{i\}). \tag{4.21}$$

Combining (4.21) and (4.15b), constraint (4.13b) is satisfied.

Proof of $\tilde{R}$ satisfying (4.13e)**:** From (4.15d), it stands that

$$\tilde{R}_{a_j,b} = 0, \quad j \in \mathcal{X}(a), b \notin \bigcap_{i \in \mathcal{X}(a)} (\mathcal{N}(i) \cup \{i\}). \tag{4.22}$$

Combining (4.15d), (4.22), and (4.15f), constraint (4.13e) is satisfied.

Finally, the proof of the equivalence of objectives $\tilde{\mathcal{J}}_i(\tilde{\boldsymbol{R}})$ and $\hat{\mathcal{J}}_i(\tilde{\boldsymbol{R}})$ is given as follows.

From conditions (4.15c) and (4.15d), it stands that

$$\sum_{b \notin (\mathcal{N}(i) \cup \{i\})} \Big(\sum_{a_i \in \tilde{\mathcal{U}}_i^p} \frac{1}{\boldsymbol{card}(\mathcal{X}(a))} \tilde{R}_{a_i,b}^2 + \sum_{a \in \tilde{\mathcal{U}}_i \setminus \tilde{\mathcal{U}}_i^p} \tilde{R}_{a,b}^2 \Big) = 0. \tag{4.23}$$

Therefore, $\tilde{J}_i(\tilde{\boldsymbol{R}})$ and $\hat{J}_i(\tilde{\boldsymbol{R}})$ are equivalent. □

From Proposition 4.1, the centralised optimisation problem (4.7), (4.3a), (4.3b), (4.6), is reposed in (4.15a)-(4.15f), where both the objective and constraints are expressed in terms of local information of each subsystem. Therefore, dual decomposition method can be employed to solve this problem in a distributed manner.

The controller synthesis for discrete-time systems follows the same idea of that for continuous-time systems, which is presented in the following section.

4.1.2 Controller synthesis for discrete-time systems

Problem formulation

Consider an interconnected system with m first-order subsystems and ν inputs. Suppose the open-loop dynamics are captured by

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{d}(k), \tag{4.24}$$

where system matrix $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ is a Γ-tree-diagonal matrix; $\boldsymbol{B} \in \mathbb{R}^{m\times \nu}$ is an input matrix. The aim is to find a linear state feedback controller

$$\boldsymbol{u}(k) = \boldsymbol{K}\boldsymbol{x}(k), \tag{4.25}$$

where $\boldsymbol{K} \in \mathbb{R}^{\nu\times m}$ is the feedback gain, such that the resulting closed-loop system (3.38) [3]

1. remains a tree topology,
2. is asymptotically stable, and
3. attenuates disturbance in the sense of Definition 3.3.

These requirements are mathematically described by the following problem formulation.

Problem 4.2 (Controller synthesis for discrete-time systems)**.** *The objective is to find a controller (4.25) for a tree structured system (4.24) such that the closed-loop system (3.38) satisfies*

(a) $\boldsymbol{BK} \circ S(\Gamma) = \boldsymbol{BK}$,

(b) $\exists \boldsymbol{P} = \boldsymbol{P}^\top > 0$ *such that* $\boldsymbol{Q}^\top \boldsymbol{P}\boldsymbol{Q} - \boldsymbol{P} < 0$, *and*

(c) $|1 - |Q_{a,a}|| > \sum_{\forall b\in\mathcal{N}(a)} |Q_{a,b}|$, $\forall a, b \in \mathcal{V}$.

Centralised controller synthesis

Similar to continuous-time cases, and a convex objective function $J(\boldsymbol{K})$ will be used for discussion. The control design is formulated as an optimisation problem with the decision variable $\boldsymbol{K}$:

$$\begin{aligned}
\min_{\boldsymbol{K}} \quad & \mathcal{J}(\boldsymbol{K}) && \text{(4.26a)}\\
\text{subject to} \quad & \boldsymbol{Q} = \boldsymbol{A} + \boldsymbol{BK}, && \text{(4.26b)}\\
& \boldsymbol{BK} \circ S(\Gamma) = \boldsymbol{BK}, && \text{(4.26c)}\\
& \boldsymbol{P} = \boldsymbol{P}^\top > 0, \quad \boldsymbol{Q}^\top \boldsymbol{P}\boldsymbol{Q} - \boldsymbol{P} < 0, && \text{(4.26d)}\\
& |1 - |Q_{a,a}|| > \sum_{\forall b\in\mathcal{N}(a)} |Q_{a,b}|, \forall a, b \in \mathcal{V}. && \text{(4.26e)}
\end{aligned}$$

[3] It implies that $\boldsymbol{Q} = \boldsymbol{A} + \boldsymbol{BK}$.

The above optimisation problem is non-convex due to constraint (4.26e). Here, we tackle this problem by substituting constraint (4.26e) by a convex constraint

$$\alpha \geq \sum_{\forall b \in \mathcal{N}(a) \cup a} |Q_{a,b}| \quad \forall a, b \in \mathcal{V}, \tag{4.27}$$

where $0 < \alpha < 1$ indicates the desired convergence rate.

Note that (4.27) is a sufficient condition for (4.26e). After this substitution, the resulting problem immediately becomes convex. Furthermore, from Gershgorin disk theorem [99], this tightened condition (4.27) automatically enforces the eigenvalues of $\boldsymbol{Q}$ to be within the unit disk, implying that the interconnected linear system is stable. Therefore, constraint (4.26d), which is included to ensure stability, can now be eliminated.

Distributed controller synthesis

Centralised controller design may not be possible for large-scale networks due to limited communication and computation resources, thus a distributed synthesis method is proposed under the following assumption.

To realise distributed design, the objective function $\mathcal{J}(\boldsymbol{K})$ should be able to express as a sum of elements depending on local information. For example, $\mathcal{J}(\boldsymbol{K})$ can be $\sum_{a=1}^{v}\sum_{b=1}^{m}|K_{a,b}|$ or the Frobenius norm $||\boldsymbol{K}||_F$, both of which are approximations of cardinality of $\boldsymbol{K}$ [97, 98]. Here, we consider Frobenius norm as an example to realise distributed design. Note that minimising $||\boldsymbol{K}||_F$ is equivalent to minimising $||\boldsymbol{K}||_F^2$, thus the objective is given as follows:

$$\min_{\boldsymbol{K}} \qquad ||\boldsymbol{K}||_F^2. \tag{4.28}$$

By breaking up the centralised optimisation problem (4.28), (4.26b), (4.26c), (4.27) to subproblems, subsystem controllers can be designed locally. Given (4.24), the dynamics of subsystem i can be captured by

$$x_i(k+1) = \sum_{j \in (\mathcal{N}(i) \cup \{i\})} A_{i,j} x_j(k) + \sum_{a \in \mathcal{U}_i} B_{i,a} u_a(k) + d_i(k), \tag{4.29}$$

where $\mathcal{U}_i$ is the set of inputs indices of subsystem i. An input u_a is termed as a *public input*,

if it affects more than one subsystem, i.e. $a \in (\mathcal{U}_i \cap \mathcal{U}_j \cap ... \cap \mathcal{U}_g), i \neq j \neq ... \neq g$. Let $\mathcal{X}(a) := \{i, j, ..., g\}$ be the set of indices of subsystems that share input u_a[4]. Let $\mathcal{U}_i^p$ denote the set of public inputs indices of subsystem i, i.e. $\mathcal{U}_i^p := \{a | i \in \mathcal{X}(a)\}$. On the other hand, if an input u_a affects one subsystem only, then it is termed as a *private input*. The set of private inputs indices of subsystem i is given by $\mathcal{U}_i \setminus \mathcal{U}_i^p$. Thus, state equation (4.29) can be reposed in terms of public and private inputs as follows:

$$x_i(k+1) = \sum_{j \in (\mathcal{N}(i) \cup \{i\})} A_{i,j} x_j(k) + \sum_{a \in \mathcal{U}_i^p} B_{i,a} u_a(k) + \sum_{a \in \mathcal{U}_i \setminus \mathcal{U}_i^p} B_{i,a} u_a(k) + d_i(k). \tag{4.30}$$

By taking an insight of the optimisation problem, one can find that the only coupling between subsystems comes from the feedback gains associated with public inputs. Hence, to decompose the problem, we introduce new inputs $\tilde{u}_{a_i}, \tilde{u}_{a_j}, ..., \tilde{u}_{a_g}$, which are local versions of public input u_a in subsystems $i, j, ..., g$, respectively. Note that

$$\tilde{u}_{a_i} = \tilde{u}_{a_j} = ... = \tilde{u}_{a_g}. \tag{4.31}$$

Let $\boldsymbol{card}(\cdot)$ denote the cardinality of a set. Construct $\tilde{\boldsymbol{u}} \in \mathbb{R}^{(\sum_{i=1}^m \boldsymbol{card}(\mathcal{U}_i)) \times 1}$ from the private inputs and local versions of public inputs of each subsystem. Corresponding to $\mathcal{U}_i^p$, denote the set of local versions of public inputs indices of subsystem i in vector $\tilde{\boldsymbol{u}}$ by $\tilde{\mathcal{U}}_i^p$. Similarly, the set of inputs indices in $\tilde{\boldsymbol{u}}$ corresponding to the private inputs in $\mathcal{U}_i$ is given by $\tilde{\mathcal{U}}_i \setminus \tilde{\mathcal{U}}_i^p$.

The state equation (4.24) can be rewritten in terms of $\tilde{\boldsymbol{u}}$ as in

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \tilde{\boldsymbol{B}}\tilde{\boldsymbol{u}}(k) + \boldsymbol{d}(k), \tag{4.32}$$

where $\tilde{\boldsymbol{B}} \in \mathbb{R}^{m \times \sum_{i=1}^m \boldsymbol{card}(\mathcal{U}_i)}$ is the input matrix, and (4.31) holds for $\tilde{\boldsymbol{u}}$.

The control law for (4.32) is given by

$$\tilde{\boldsymbol{u}}(k) = \tilde{\boldsymbol{K}} x(k), \tag{4.33}$$

where $\tilde{\boldsymbol{K}} \in \mathbb{R}^{(\sum_{i=1}^m \boldsymbol{card}(\mathcal{U}_i)) \times m}$ is the state feedback matrix.

[4] $\mathcal{X}(a)$ is defined exclusively for public inputs. Therefore, it is assumed that $\mathcal{X}(a)$ has at least two elements when it is used.

$$\min_{\tilde{\boldsymbol{K}}} \quad \sum_{i=1}^{m} \tilde{\mathcal{J}}_i(\tilde{\boldsymbol{K}}) \tag{4.34a}$$

$$\text{subject to} \quad \tilde{\boldsymbol{Q}}_{(i,:)} = \boldsymbol{A}_{(i,:)} + \tilde{\boldsymbol{B}}_{(i,:)}\tilde{\boldsymbol{K}}, \tag{4.34b}$$

$$(\tilde{\boldsymbol{B}}_{(i,:)}\tilde{\boldsymbol{K}}) \circ S_{(i,:)}(\Gamma) = \tilde{\boldsymbol{B}}_{(i,:)}\tilde{\boldsymbol{K}}, \tag{4.34c}$$

$$1 > \sum_{\forall j \in \mathcal{N}(i)\cup\{i\}} |\tilde{Q}_{i,j}|, \tag{4.34d}$$

$$\tilde{\boldsymbol{K}}_{(a_i,:)} = \tilde{\boldsymbol{K}}_{(a_j,:)}, \ \forall i, j \in \mathcal{X}(a), \tag{4.34e}$$

where $i = 1, 2, ...m$, and

$$\tilde{\mathcal{J}}_i(\tilde{\boldsymbol{K}}) := \sum_{a_i \in \tilde{\mathcal{U}}_i^p} \frac{1}{\boldsymbol{card}(\mathcal{X}(a))} \tilde{\boldsymbol{K}}_{(a_i,:)} \tilde{\boldsymbol{K}}_{(a_i,:)}^\top + \sum_{a \in \tilde{\mathcal{U}}_i \setminus \tilde{\mathcal{U}}_i^p} \tilde{\boldsymbol{K}}_{(a,:)} \tilde{\boldsymbol{K}}_{(a,:)}^\top.$$

Note that (4.34e) is the consensus constraint corresponding to the coupling (4.31) between subsystems.

Define a cost function for subsystem i as follows:

$$\hat{\mathcal{J}}_i(\tilde{\boldsymbol{K}}) := \sum_{b \in (\mathcal{N}(i)\cup\{i\})} \Big(\sum_{a_i \in \tilde{\mathcal{U}}_i^p} \frac{1}{\boldsymbol{card}(\mathcal{X}(a))} \tilde{K}_{a_i,b}^2 + \sum_{a \in \tilde{\mathcal{U}}_i \setminus \tilde{\mathcal{U}}_i^p} \tilde{K}_{a,b}^2 \Big). \tag{4.35}$$

Note that problem (4.34a)-(4.34e) is still described in terms of global information. To facilitate distributed design under Assumption 4.1, the following proposition will show that it only requires local information to solve this problem.

Proposition 4.2. *The optimisation problem (4.34a)-(4.34e) is equivalent to*

$$\min_{\tilde{\boldsymbol{K}}} \quad \sum_{i=1}^{m} \hat{\mathcal{J}}_i(\tilde{\boldsymbol{K}}) \tag{4.36a}$$

$$\textit{subject to} \quad \tilde{Q}_{i,b} = A_{i,b} + \sum_{a \in \tilde{\mathcal{U}}(i)} \tilde{B}_{i,a}\tilde{K}_{a,b}, \quad b \in (\mathcal{N}(i)\cup\{i\}), \tag{4.36b}$$

$$\tilde{K}_{a,b} = 0, a \in \tilde{\mathcal{U}}_i, b \notin (\mathcal{N}(i)\cup\{i\}), \tag{4.36c}$$

$$\tilde{K}_{a_i,b} = 0, i \in \mathcal{X}(a), \quad b \notin \bigcap_{j \in \mathcal{X}(a)} (\mathcal{N}(j)\cup\{j\}), \tag{4.36d}$$

$$1 > \sum_{\forall j \in \mathcal{N}(i) \cup \{i\}} |\tilde{Q}_{i,j}|, \tag{4.36e}$$

$$\tilde{K}_{(a_i,b)} = \tilde{K}_{(a_j,b)}, \quad i,j \in \mathcal{X}(a), \quad b \in \bigcap_{j \in \mathcal{X}(a)} (\mathcal{N}(j) \cup \{j\}), \tag{4.36f}$$

for $i = 1, 2, ..., m$.

Proof. The proof includes three parts. Specifically, the first two parts show the equivalence of the constraints (4.34b), (4.34c), (4.34e) and (4.36b)-(4.36d), (4.36f), and the last part proves the equivalence of the objectives $\tilde{\mathcal{J}}_i(\tilde{\boldsymbol{K}})$ and $\hat{\mathcal{J}}_i(\tilde{\boldsymbol{K}})$.

First, suppose a matrix $\tilde{\boldsymbol{K}}$ satisfies conditions (4.34b), (4.34c), (4.34e). It will show that conditions (4.36b)-(4.36d), (4.36f) must be satisfied for $\tilde{\boldsymbol{K}}$ as well.

Proof of $\tilde{K}$ satisfying (4.36c)

The j-th element of the one-dimensional vector $\tilde{\boldsymbol{B}}_{(i,:)}\tilde{\boldsymbol{K}}$ is given by

$$(\tilde{\boldsymbol{B}}_{(i,:)}\tilde{\boldsymbol{K}})_j := \sum_{a=1}^{\sum_{h=1}^{m} \boldsymbol{card}(\mathcal{U}_h)} \tilde{B}_{i,a}\tilde{K}_{a,j}, \quad j = 1, 2, ..., m. \tag{4.37}$$

Note that $\tilde{B}_{i,a} = 0$ if $a \notin \tilde{\mathcal{U}}_i$. Thus, (4.37) is equivalent to

$$\tilde{\boldsymbol{B}}_{(i,:)}\tilde{\boldsymbol{K}}_{(:,j)} := \sum_{a \in \tilde{\mathcal{U}}_i} \tilde{B}_{i,a}\tilde{K}_{a,j}, \quad j = 1, 2, ..., m. \tag{4.38}$$

Also, notice that the j-th element of vector $S_{(i,:)}(\Gamma)$ is zero, if j is not a neighbour of i, i.e.

$$S(\Gamma)_{(i,j)} = 0, \quad \forall j \notin (\mathcal{N}(i) \cup \{i\}). \tag{4.39}$$

Combining (4.34c), (4.38), and (4.39), condition (4.36c) can be guaranteed.

Proof of $\tilde{K}$ satisfying (4.36b) From (4.36c) and (4.34b), constraint (4.36b) is satisfied.

Proof of $\tilde{K}$ satisfying (4.36d) Given $i, j \in \mathcal{X}(a)$, the following equations should be satisfied based on (4.36c):

$$\tilde{K}_{a_i,b} = 0, \quad b \notin (\mathcal{N}(i) \cup \{i\}), \tag{4.40}$$

$$\tilde{K}_{a_j,b} = 0, \quad b \notin (\mathcal{N}(j) \cup \{j\}). \tag{4.41}$$

Combining (4.40), (4.41), and (4.34e), condition (4.36d) is satisfied.

Proof of $\tilde{K}$ satisfying (4.36f) Notice that (4.36f) is a subset of (4.34e), thus it holds naturally.

Second, the other direction of the proof is given as follows.

Suppose $\tilde{\boldsymbol{K}}$ satisfies conditions (4.36b)-(4.36d), and (4.36f).

Proof of $\tilde{K}$ satisfying (4.34c) Combining (4.38) and (4.36c), condition (4.34c) can be obtained.

Proof of $\tilde{K}$ satisfying (4.34b) From (4.34c), it holds that

$$(\boldsymbol{A}_{(i,:)} + \tilde{\boldsymbol{B}}_{(i,:)}\tilde{\boldsymbol{K}})_b = 0, \quad b \notin (\mathcal{N}(i) \cup \{i\}). \tag{4.42}$$

Combining (4.42) and (4.36b), constraint (4.34b) is satisfied.

Proof of $\tilde{K}$ satisfying (4.34e) From (4.36d), it stands that

$$\tilde{K}_{a_j,b} = 0, \quad j \in \mathcal{X}(a), b \notin \bigcap_{i \in \mathcal{X}(a)} (\mathcal{N}(i) \cup \{i\}). \tag{4.43}$$

Combining (4.36d), (4.43), and (4.36f), constraint (4.34e) is satisfied.

Finally, the proof of the equivalence of objectives $\tilde{\mathcal{J}}_i(\tilde{\boldsymbol{K}})$ and $\hat{\mathcal{J}}_i(\tilde{\boldsymbol{K}})$ is given as follows.

From conditions (4.36c) and (4.36d), it stands that

$$\sum_{b \notin (\mathcal{N}(i) \cup \{i\})} \Big(\sum_{a_i \in \tilde{\mathcal{U}}_i^p} \frac{1}{\boldsymbol{card}(\mathcal{X}(a))} \tilde{K}_{a_i,b}^2 + \sum_{a \in \tilde{\mathcal{U}}_i \setminus \tilde{\mathcal{U}}_i^p} \tilde{K}_{a,b}^2 \Big) = 0. \tag{4.44}$$

Therefore, $\tilde{\mathcal{J}}_i(\tilde{\boldsymbol{K}})$ and $\hat{\mathcal{J}}_i(\tilde{\boldsymbol{K}})$ are equivalent. □

From Proposition 4.2, the centralised optimisation problem (4.28), (4.26b), (4.26c), (4.27), is reposed in (4.36a)-(4.36f), where both the objective and constraints are expressed in terms of local information of each subsystem. Therefore, dual decomposition method can be employed to solve this problem in a distributed manner.

4.2 Tree approximation of an interconnected system

In this section, a tree approximation approach will be proposed so that the states of an arbitrarily connected network are close to those of a tree-structured network. With this method, the results in Chapter 3 and Section 4.1 can be applied to more general cases.

4.2.1 Tree approximation for continuous-time systems

We start by considering an interconnected system with a connection topology $\mathcal{G}(\mathcal{V},\mathcal{E}')$. Suppose there are ν' inputs, and the dynamic equation is given by

$$\dot{\boldsymbol{\varphi}}(t) = \boldsymbol{H}'\boldsymbol{\varphi}(t) + \boldsymbol{J}'\boldsymbol{\varsigma}(t) + \boldsymbol{d}(t) \tag{4.45}$$

where $\boldsymbol{\varphi}(t) \in \mathbb{R}^{m\times 1}$, and $\boldsymbol{\varsigma}(t) \in \mathbb{R}^{\nu'\times 1}$ are the state and input vector, respectively; $\boldsymbol{H}' \in \mathbb{R}^{m\times m}$ and $\boldsymbol{J}' \in \mathbb{R}^{m\times\nu'}$ are the system and input matrix, respectively; In particular, matrix $\boldsymbol{H}'$ is a $\mathcal{G}$-structured matrix, i.e. $\boldsymbol{H}' \circ S(\mathcal{G}) = \boldsymbol{H}'$. The control law is given by

$$\boldsymbol{\varsigma}(t) = \boldsymbol{R}'\boldsymbol{\varphi}(t). \tag{4.46}$$

The objective is to find a spanning tree $\Gamma(\mathcal{V},\mathcal{E})$ of $\mathcal{G}(\mathcal{V},\mathcal{E}')$, i.e. $\mathcal{E} \subseteq \mathcal{E}'$, such that for any given $\boldsymbol{R}'$ satisfying $\boldsymbol{J}'\boldsymbol{R}' \circ S(\Gamma) = \boldsymbol{J}'\boldsymbol{R}'$, the state trajectories of a tree-structured system

$$\dot{\boldsymbol{x}}(t) = (\boldsymbol{H} + \boldsymbol{J}'\boldsymbol{R}')\boldsymbol{x}(t) + \boldsymbol{d}(t), \tag{4.47}$$

are close to the original closed-loop system

$$\dot{\boldsymbol{\varphi}}(t) = (\boldsymbol{H}' + \boldsymbol{J}'\boldsymbol{R}')\boldsymbol{\varphi}(t) + \boldsymbol{d}(t), \tag{4.48}$$

where $\boldsymbol{H} \coloneqq \boldsymbol{H}' \circ S(\Gamma)$.

Dynamics (4.48) of the original system can be rewritten in terms of $\boldsymbol{H}$ as in

$$\dot{\boldsymbol{\varphi}}(t) = (\boldsymbol{H} + \boldsymbol{J}'\boldsymbol{R}')\boldsymbol{\varphi}(t) + \Delta\boldsymbol{H}\boldsymbol{\varphi}(t) + \boldsymbol{d}(t), \tag{4.49}$$

where $\Delta \boldsymbol{H} \coloneqq \boldsymbol{H}' - \boldsymbol{H}$.

By comparison, the original system (4.49) can be seen as the closed-loop tree approximation model (4.47) perturbed by $\Delta \boldsymbol{H}\boldsymbol{\varphi}(k)$. Therefore, (4.49) can be well approximated by (4.47) when $||\Delta \boldsymbol{H}||$ is small. Particularly, when $||\Delta \boldsymbol{H}|| = 0$,[5] model (4.47) becomes exactly the same as the original model (4.49).

From this viewpoint, tree approximation problem can be transformed to the following problem.

Problem 4.3. *Given a $\mathcal{G}(\mathcal{V}, \mathcal{E}')$-structured matrix $\boldsymbol{H}'$, i.e. $\boldsymbol{H}' \circ S(\mathcal{G}) = \boldsymbol{H}'$, find a spanning tree Γ in $\mathcal{G}$ such that $\sum_{i=1}^{m}\sum_{j=1}^{m} |\Delta H_{i,j}|$ is minimised, i.e.*

$$\min_{\boldsymbol{H}} \quad \sum_{i=1}^{m}\sum_{j=1}^{m} |\Delta H_{i,j}| \tag{4.50}$$

$$s.t. \quad \Delta \boldsymbol{H} = \boldsymbol{H}' - \boldsymbol{H}, \tag{4.51}$$

$$\boldsymbol{H}' \circ S(\Gamma(\mathcal{V}, \mathcal{E})) = \boldsymbol{H}, \tag{4.52}$$

$$\mathcal{E} \subseteq \mathcal{E}'. \tag{4.53}$$

To tackle this problem, the definition of a maximum spanning tree (MST) problem is introduced as follows.

Definition 4.1 (MST problem)**.** *Given an undirected weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$, the MST problem is to find the spanning tree $\Gamma(\mathcal{V}', \mathcal{E}', \mathcal{A}')$ in $\mathcal{G}$ with maximum possible total edge weight, i.e.*

$$\max_{\mathcal{E}'} \quad \sum_{i=1}^{m}\sum_{j=1}^{m} |\mathcal{A}'_{i,j}| \tag{4.54a}$$

$$s.t. \quad \mathcal{V}' = \mathcal{V}, \tag{4.54b}$$

$$\mathcal{E}' \subseteq \mathcal{E}, \tag{4.54c}$$

$$\mathcal{A}'_{i,j} = \mathcal{A}_{i,j},\ \forall \mathcal{A}'_{i,j} \neq 0. \tag{4.54d}$$

Two undirected weighted graphs $\bar{\mathcal{G}}(\mathcal{V}, \mathcal{E}', \mathcal{A}_1)$ and $\bar{\Gamma}(\mathcal{V}, \mathcal{E}, \mathcal{A}_2)$ are constructed from the unweighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}')$ and $\Gamma(\mathcal{V}, \mathcal{E})$, respectively. The edge weighting function $\mathcal{A}_i$, $i = 1, 2$ is defined as $\mathcal{A}_1(i, j) \coloneqq |H'_{i,j}| + |H'_{j,i}|$, $\forall (i, j) \in \mathcal{E}'$ and $\mathcal{A}_2(i, j) \coloneqq |H_{i,j}| + |H_{j,i}|$, $\forall (i, j) \in \mathcal{E}$, respectively.

[5] Due to norm properties, $||\Delta \boldsymbol{H}|| = 0$ implies that $\boldsymbol{H} = \boldsymbol{H}'$. In such cases, the original graph $\mathcal{G}$ is a tree already.

With above definitions, the solution to Problem 4.3 is presented in the following proposition.

Proposition 4.3. *The problem (4.50)-(4.53) is equivalent to the MST problem in* $\bar{\mathcal{G}}(\mathcal{V}, \mathcal{E}', \mathcal{A}_1)$. *Furthermore, the MST is* $\bar{\Gamma}(\mathcal{V}, \mathcal{E}, \mathcal{A}_2)$.

Proof. From conditions (4.51) to (4.53), $\sum_{i=1}^{m}\sum_{j=1}^{m}|\Delta H_{i,j}|$ can be broken into two components, i.e.

$$\sum_{i=1}^{m}\sum_{j=1}^{m}|\Delta H_{i,j}| = \sum_{i=1}^{m}\sum_{j=1}^{m}|H'_{i,j}| - \sum_{i=1}^{m}\sum_{j=1}^{m}|H_{i,j}|. \tag{4.55}$$

Note that given a physical system, $\sum_{i=1}^{m}\sum_{j=1}^{m}|H'_{i,j}|$ is a constant number. Consequently, the problem of minimising $\sum_{i=1}^{m}\sum_{j=1}^{m}|\Delta H_{i,j}|$ is transformed to maximizing $\sum_{i=1}^{m}\sum_{j=1}^{m}|H_{i,j}|$. Furthermore, the sum $\sum_{i\in\mathcal{V}}|H_{i,i}|$ is also a constant number, since $H_{i,i} = H'_{i,i}$. Thus, the optimisation problem (4.50) is equivalent to

$$\max_{A} \quad \sum_{(i,j)\in\mathcal{E}}|H_{i,j}| \tag{4.56}$$

subject to constraints (4.52) and (4.53). Note that this is exactly the mathematical formulation of the MST problem in Proposition 4.3. Thereby, Proposition 4.3 is proven. □

4.2.2 Tree approximation for discrete-time systems

The tree approximation for discrete-time systems follows the same procedure as in Section 4.2.1.

We start by considering an interconnected system with a connection topology $\mathcal{G}(\mathcal{V}, \mathcal{E}')$. Suppose there are ν' inputs, and the dynamic equation is given by

$$\boldsymbol{\xi}(k+1) = \boldsymbol{A}'\boldsymbol{\xi}(k) + \boldsymbol{B}'\boldsymbol{\mu}(k) + \boldsymbol{d}(k), \tag{4.57}$$

where $\boldsymbol{\xi}(k) \in \mathbb{R}^{m\times 1}$, and $\boldsymbol{\mu}(k) \in \mathbb{R}^{\nu'\times 1}$ are the state and input vector, respectively; $\boldsymbol{A}' \in \mathbb{R}^{m\times m}$ and $\boldsymbol{B}' \in \mathbb{R}^{m\times\nu'}$ are the system and input matrix, respectively; In particular, matrix $\boldsymbol{A}'$ is a $\mathcal{G}$-structured matrix, i.e. $\boldsymbol{A}' \circ S(\mathcal{G}) = \boldsymbol{A}'$. The control law is given by

$$\boldsymbol{\mu}(k) = \boldsymbol{K}'\boldsymbol{\xi}(k). \tag{4.58}$$

The objective is to find a spanning tree $\Gamma(\mathcal{V}, \mathcal{E})$ of $\mathcal{G}(\mathcal{V}, \mathcal{E}')$, i.e. $\mathcal{E} \subseteq \mathcal{E}'$, such that for any

given $\boldsymbol{K}'$ satisfying $\boldsymbol{B}'\boldsymbol{K}' \circ S(\Gamma) = \boldsymbol{B}'\boldsymbol{K}'$, the state trajectories of a tree-structured system

$$\boldsymbol{x}(k+1) = (\boldsymbol{A} + \boldsymbol{B}'\boldsymbol{K}')\boldsymbol{x}(k) + \boldsymbol{d}(k), \tag{4.59}$$

are close to the original closed-loop system

$$\boldsymbol{\xi}(k+1) = (\boldsymbol{A}' + \boldsymbol{B}'\boldsymbol{K}')\boldsymbol{\xi}(k) + \boldsymbol{d}(k), \tag{4.60}$$

where $\boldsymbol{A} \coloneqq \boldsymbol{A}' \circ S(\Gamma)$.

Dynamics (4.60) of the original system can be rewritten in terms of $\boldsymbol{A}$ as in

$$\boldsymbol{\xi}(k+1) = (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})\boldsymbol{\xi}(k) + \Delta\boldsymbol{A}\boldsymbol{\xi}(k) + \boldsymbol{d}(k), \tag{4.61}$$

where $\Delta\boldsymbol{A} \coloneqq \boldsymbol{A}' - \boldsymbol{A}$.

By comparison, the original system (4.61) can be seen as the closed-loop tree approximation model (4.59) perturbed by $\Delta\boldsymbol{A}\boldsymbol{\xi}(k)$. Therefore, (4.61) can be well approximated by (4.59) when $||\Delta\boldsymbol{A}||$ is small. Particularly, when $||\Delta\boldsymbol{A}|| = 0$,[6] model (4.59) becomes exactly the same as the original model (4.61).

From this point of view, tree approximation problem for continuous-time systems can be transformed to Problem 4.3, with $\boldsymbol{H}' \coloneqq \boldsymbol{A}'$, $\boldsymbol{H} \coloneqq \boldsymbol{A}$. Furthermore, the solution to this problem is provided in Proposition 4.3.

4.3 Conclusions

A controller synthesis method is proposed for a tree-structured open-loop system. The controller is designed under requirements that the resulting closed-loop system is asymptotically stable, disturbance attenuating, and retains a tree topology. The synthesis of the controller is formulated as an optimisation problem with an objective of minimising cardinality of the state feedback matrix, subject to constraints which are mathematical descriptions of the three requirements. The convexity of the problem is obtained by replacing the non-convex constraint of disturbance attenuation

[6] Due to norm properties, $||\Delta\boldsymbol{A}|| = 0$ implies that $\boldsymbol{A} = \boldsymbol{A}'$. In such cases, the original graph $\mathcal{G}$ is a tree already.

by a convex approximation. From Gershgorin disk theorem, it is shown that the convex approximation also ensures asymptotic stability. Therefore, Lyapunov equation, which is introduced to guarantee stability, is eliminated. The controller is then ready to be designed in a distributed manner, as only local information is needed for the rest of the problem. The distributed design of the controller is facilitated by breaking up the overall optimisation problem into local problems for each subsystem. To handle the coupling between subsystems, dual decomposition method is employed to solve this problem.

To extend the scope of this research, non-tree networks are focused on and a tree approximation method is developed in the second half of this chapter. Given an arbitrarily connected system, the approximation is realised by removing some weak connections, and the objective is to minimise the difference of the closed-loop state trajectories. From the analysis, tree approximation is transformed to an MST problem, for which many algorithms are available in the literature [55, 100, 101].

The theoretical results in Chapters 3 and 4 will be validated through simulation in the next chapter.

Chapter 5
Application

In this chapter, the main theoretical results proposed in Chapters 3 and 4 are verified through simulation. Two application examples are considered here. Specifically, an interconnected pumped water supply system is adopted for the validation of the continuous-time results, and a multi-region perimeter control system is simulated for discrete-time verification. Both continuous and discrete-time simulations contain three aspects: validation of disturbance attenuating controller design method proposed in Section 4.1, demonstration of time efficiency for evaluation of sufficient disturbance attenuation conditions proposed in Chapter 3, and demonstration of the efficacy of tree approximation method proposed in Section 4.2.

The continuous-time simulation is presented in Section 5.1 with subsections of modelling, set-ups, and results. Later, the discrete-time simulations are given in Section 5.2.

5.1 Simulation for a continuous-time system: a water supply system

5.1.1 Model description of the water supply system

We consider a pumped water supply system for the simulation of continuous-time systems. This model can represent the water supply system in buildings, or approximate the pumped irrigation systems, of which the full model can be described by Saint-Venant equations, see reference [102].

A substantial proportion of the first half (continuous-time system simulation) of this chapter has been published in [28], and the second half (discrete-time system simulation) of this chapter has been submitted as a journal paper [93] (under review).

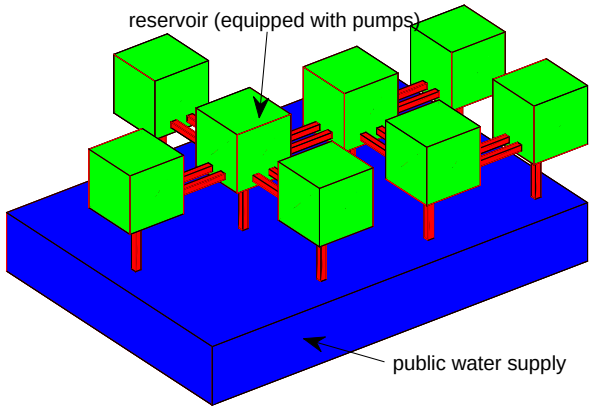


Figure 5.1: A water supply network with a tree topology.

As shown in Figure 5.1, a group of reservoirs (green) are connected through tubes (red) to form a tree-structured water supply network, where the transfer flows in tubes are controlled by pumps equipped in each reservoir. Leakage and evaporation occur in tubes, which can be seen as a disturbance to the system. These reservoirs are also connected to a public supply system (blue), which supplies water to these reservoirs. The objective is to maintain the water level of each reservoir at a desired operating point.

Neighbouring reservoirs i and j are connected through two unidirectional tubes ij and ji, where the sequence indicates the direction of the water flow. These flows are driven by pumps equipped inside those reservoirs. The flow $q_{ij}(t)$ from reservoir i to j at time t is given by

$$q_{ij}(t) = \varpi_{ij}(t)m_i(t), \tag{5.1}$$

where $\varpi_{ij}(t)$ is the control input corresponding to tube ij, and $m_i(t)$ is the mass of water in reservoir i at time t.

The flow $q_{pi}(t)$ from the public supply system to a reservoir i at time t is given by

$$q_{pi}(t) = \varpi_{pi}(t)\hat{m}_i, \tag{5.2}$$

where $\hat{m}_i$ is the desired amount of water in reservoir i.

The conservation equation of water for reservoir i is given by

$$\dot{m}_i(t) = q_{in,i}(t) - q_{out,i}(t) + \varrho_i(t), \tag{5.3}$$

where $q_{in,i}(t)$, $q_{out,i}(t)$ are the inflow and outflow of reservoir i, respectively; $\varrho_i(t)$ is the disturbance of subsystem i.

By breaking inflow $q_{in,i}(t)$ and outflow $q_{out,i}(t)$ into components, dynamics (3.38) can be rewritten as

$$\dot{m}_i(t) = \sum_{j\in\mathcal{N}(i)} q_{ji}(t) - \sum_{j\in\mathcal{N}(i)} q_{ij}(t) + q_{pi}(t) + \varrho_i(t). \tag{5.4}$$

Substituting (5.1) and (5.2) into equation (5.4), we have

$$\dot{m}_i(t) = \sum_{j\in\mathcal{N}(i)} \varpi_{ji}(t)m_j(t) - \sum_{j\in\mathcal{N}(i)} \varpi_{ij}(t)m_i(t) + \varpi_{pi}(t)\hat{m}_i + \varrho_i(t). \tag{5.5}$$

Given desired operating points $\hat{m}_i$, $\hat{\varpi}_{hi}$ and $\hat{\varrho}_i$ for each reservoir i, the non-linear model (5.5) can be linearised around the operating point, which yields

$$\begin{aligned}\Delta\dot{m}_i(t) = \quad & \sum_{j\in\mathcal{N}(i)} \hat{\varpi}_{ji}\Delta m_j(t) + \sum_{j\in\mathcal{N}(i)} \Delta\varpi_{ji}(t)\hat{m}_j - \sum_{j\in\mathcal{N}(i)} \hat{\varpi}_{ij}\Delta m_i(t) \\ & - \sum_{j\in\mathcal{N}(i)} \Delta\varpi_{ij}(t)\hat{m}_i + \Delta\varpi_{pi}(t)\hat{m}_i + \Delta\varrho_i(t).\end{aligned} \tag{5.6}$$

The state equation (5.6) can be written in a vector form as follows:

$$\Delta\dot{\boldsymbol{m}}(t) = \tilde{\boldsymbol{H}}\Delta\boldsymbol{m}(t) + \tilde{\boldsymbol{J}}\Delta\boldsymbol{\varpi}(t) + \Delta\boldsymbol{\varrho}(t), \tag{5.7}$$

where $\Delta\boldsymbol{m}(t) \in \mathbb{R}^m$ is the state vector; $\Delta\boldsymbol{\varpi}(t) \in \mathbb{R}^v$ is the control input vector; $\Delta\boldsymbol{\varrho}(t) \in \mathbb{R}^m$ is the disturbance vector; $\tilde{\boldsymbol{H}} \in \mathbb{R}^{m\times m}$ is a tree-diagonal system matrix; $\tilde{\boldsymbol{J}} \in \mathbb{R}^{m\times v}$ is the input matrix.

Considering that the reservoirs have different capacities, the analysis of disturbance propaga-

tion should be conducted upon normalised mass $x(t)$, i.e.

$$\boldsymbol{x}(t) \coloneqq \tilde{\boldsymbol{L}}\Delta\boldsymbol{m}(t), \tag{5.8}$$

where $\tilde{\boldsymbol{L}}$ is a diagonal matrix with diagonal elements defined as

$$\tilde{L}_{i,i} \coloneqq \frac{1}{\hat{m}_i}. \tag{5.9}$$

Combining (5.7) and (5.8), the dynamics of the reservoirs can be represented by (4.1), where $\boldsymbol{H} = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{H}}\tilde{\boldsymbol{L}}^{-1}$, $\boldsymbol{J} = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{J}}$, $\boldsymbol{u}(t) = \Delta\boldsymbol{\varpi}(t)$, and $\boldsymbol{d}(t) = \tilde{\boldsymbol{L}}\Delta\boldsymbol{\varrho}(t)$. Therefore, the pumped water supply system design problem has been translated to a controller design problem as described in Problem 4.1.

5.1.2 Simulation set-up of the water supply system

For the numerical results in this section, the mass of water in each reservoir at the desired point is set to be uniformly distributed random numbers between 100 kilograms and 150 kilograms. The control input $\hat{\varpi}$ at the desired point is selected through trial and error.

It is assumed for all simulation scenarios that the initial states are zero, i.e. $\boldsymbol{x}(0) = 0$.

5.1.3 Simulation results of the water supply system

Validation of continuous-time disturbance attenuating controller synthesis method

For this simulation scenario, a network with 50 reservoirs is considered. As depicted in Figure 5.2, the topology of the network can be represented by a tree $\hat{\Gamma}$.[1]

[1] It follows that the system matrix $\boldsymbol{H}$ is a $\hat{\Gamma}$ tree-diagonal matrix.

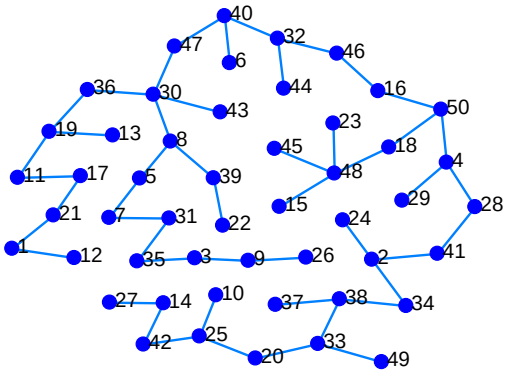


Figure 5.2: A 50-reservoir water supply network with a tree topology $\hat{\Gamma}$.

A controller is developed following the controller synthesis method proposed in Section 4.1.1. Specifically, the control law is obtained by solving a convex optimisation problem with objective (4.3a) subject to constraints (4.3b), (4.3c), and (4.6), where ϵ is chosen to be -0.2.

The disturbance attenuation property of the developed controller is demonstrated via injecting an uniformly distributed random disturbance between -1 and 1 in a subsystem for 18000 seconds. Two scenarios are considered for simulation. In the first scenario, the disturbance occurs in an internal node 30, while in the second scenario it occurs in a leaf node 15.

Let $\hat{\mathcal{T}}_{30}$ be the rooted tree by treating node 30 in $\hat{\Gamma}$ as the root. In this simulation scenario, there exist 16 disturbance propagation paths from root 30 to all the leaves in rooted tree $\mathcal{T}_{30}$. The energy of a state x_i is represented by $\hat{\gamma} := \log_{10}(\int_{t=0}^{18000} x_i^2(t)dt)$, where the logarithmic scale is used due to the rapid decrease of the energy along propagation paths. The results for the energy following injection of the uniformly distributed disturbance in node 30 are plotted in Figure 5.3, where the x-axis is the level of a node i in the rooted tree $\hat{\mathcal{T}}_{31}$, and the y-axis is the energy $\hat{\gamma}$.

Let $\hat{\mathcal{T}}_{15}$ be the rooted tree by treating node 15 in $\hat{\Gamma}$ as the root. In the second scenario, there exist 15 disturbance propagation paths from root 15 to all the leaves in rooted tree $\hat{\mathcal{T}}_{15}$. Like Figure 5.3, the energy of the states along these disturbance propagation paths are shown in Figure 5.4.

It can be seen that the energy of the states is attenuated through propagation along all paths in both scenarios, which demonstrates the disturbance attenuation property of the developed continuous-time controller. Thereby, the continuous-time disturbance attenuating controller synthesis method in Section 4.1.1 is validated.

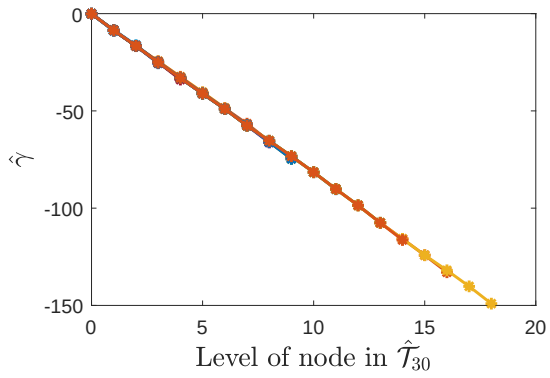


Figure 5.3: The energy of the states along all paths in $\hat{\mathcal{T}}_{30}$ starting from node 30.

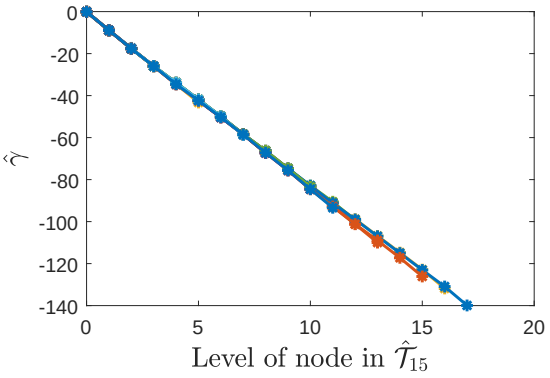


Figure 5.4: The energy of the states along all paths in $\hat{\mathcal{T}}_{15}$ starting from node 15.

Required computation time for evaluation of continuous-time sufficient disturbance attenuation conditions

Having established that disturbance attenuation can be guaranteed, we now change focus to the earlier results on efficiency of computation. Specifically, the time efficiency of the sufficient conditions in Proposition 3.2 is demonstrated through a comparison to conditions in Proposition 3.1.

We consider five groups of water supply networks with different number of subsystems: 10, 20, 30, 40, and 50, and each group includes ten different arbitrarily generated tree topologies. Similar to previous section, disturbance attenuating controllers are designed for each network following the method introduced in Section 4.1.1.

The disturbance attenuation property of the resulting closed-loop systems is verified through

Table 5.1: Mean verification times (t) and standard deviation (σ) for continuous-time systems

Network size	Prop. 3.1		Prop. 3.2	
	t [s]	σ	t [s]	σ ($\times 10^{-6}$)
10	1.5×10^{-1}	0.009	0.8×10^{-5}	0.8
20	7.6×10^{-1}	0.027	1.7×10^{-5}	1.0
30	2.4×10^{0}	0.048	2.6×10^{-5}	2.6
40	6.1×10^{0}	0.127	3.8×10^{-5}	9.6
50	2.4×10^{1}	0.863	4.4×10^{-5}	3.1

Propositions 3.1 and 3.2, respectively. Verification time is obtained using `timeit` function in MATLAB, and each verification is repeated for ten times.[2] The verification results show that disturbance attenuation conclusions can be drawn for all tested networks both by Propositions 3.1 and 3.2, whereas the time consumption for verification is quite different.

To investigate the influence of network size, the mean verification time t are calculated by averaging over both different topologies of each group and ten repetitions of each network. The mean verification times and corresponding standard deviation σ under each network size are given in Table 5.1. It can be observed in Table 5.1 that for each network size, disturbance attenuation verification is more time-efficient with Proposition 3.2. It also shows that verification time with Proposition 3.1 increases rapidly as the number of region grows, whilst time for Proposition 3.2 changes slowly. This numerical result demonstrates an advantage of the conditions in Proposition 3.2, that is, they are more time-efficient than conditions in Proposition 3.1, and this advantage becomes more obvious when the network size is larger.

Simulation of tree approximation method in continuous-time systems

The tree approximation method proposed in Section 4.2.1 is verified for continuous-time systems through comparing a series of different tree approximations of a non-tree structured network.

[2]The configuration of the laptop used for disturbance attenuation verification is as follows. Processor: Intel(R) Core(TM) i5-6200U CPU @2.3GHz; RAM: 8GB; System: 64-bit Windows 10.

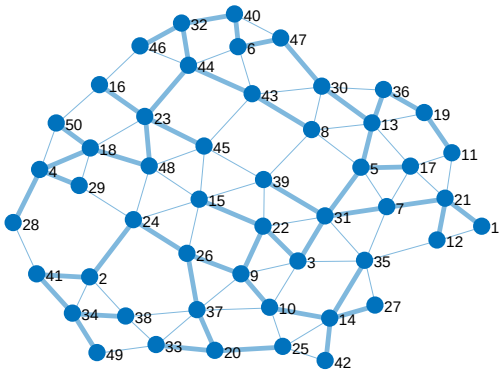


Figure 5.5: A 50-reservoir network $\hat{\mathcal{G}}$ with all interconnections (fine) and the maximum spanning tree (bold).

The tree approximation method proposed in Section 4.2.1 is verified through comparing a series of different tree approximations of a non-tree structured water supply network.

Consider a pumped water supply system with 50 reservoirs and its topology $\hat{\mathcal{G}}$ is depicted in Figure 5.5. A tree-structured system is developed following the tree approximation method introduced in Section 4.2.1. According to Proposition 4.3, the tree approximation problem can be transformed into an MST problem in $\hat{\mathcal{G}}$. In this paper, Prim's algorithm [55] is utilised to find the MST solving Problem 4.3, which is highlighted in Figure 5.5.

To test the efficacy of the proposed method for generating a tree approximation, we now choose 22 other candidate trees, each of which is an arbitrarily generated spanning tree of $\hat{\mathcal{G}}$. Disturbance attenuating controllers are designed for these tree networks following the controller synthesis method proposed in Section 4.1.1.

The developed controllers are applied to both the original system and the tree-structured systems, and the closed-loop models are built in MATLAB/SIMULINK. We consider continuous injection of 50 independent and uniformly distributed random disturbances between -1 and 1 into every node of the network. Each simulation is run for 18000 seconds.

We use the difference between the original system state $\boldsymbol{\varphi}(t)$ and tree-structured system state $\boldsymbol{x}(t)$ to indicate the performance of the approximation. Specifically, the difference is measured by $\hat{\delta} := \sqrt{\int_{t=0}^{18000} \sum_{i=1}^{50} |\varphi_i(t) - x_i(t)|^2 dt}$, where t is the simulation time. The relationship between the system matrix difference $\sum_{i=1}^{50} \sum_{j=1}^{50} |\Delta H_{i,j}|$ and the state difference $\hat{\delta}$ is depicted in Figure 5.6, where each '$\circ$' corresponds to a pair of closed-loop tree approximation system and the

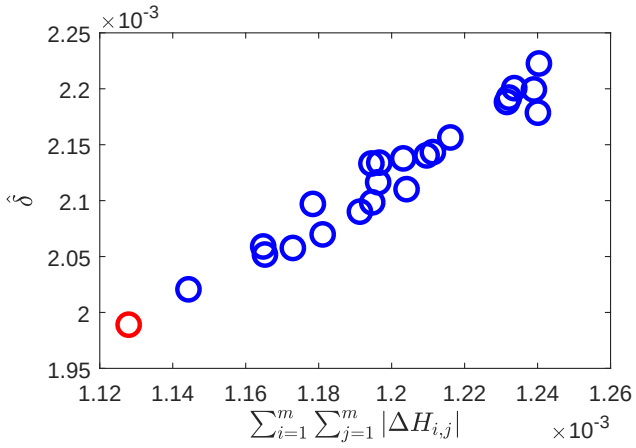


Figure 5.6: The relationship between the system matrix difference and the state difference in continuous-time systems.

corresponding original system. In particular, the system developed following the proposed tree approximation method in Section 4.2.1, is in red colour.

It can be seen in Figure 5.6 that there is a general trend of increasing difference of states between the original system and the tree approximation system when $\sum_{i=1}^{m}\sum_{j=1}^{m}|\Delta H_{i,j}|$ grows. This observation is consistent with an important hypothesis of the proposed tree approximation method in Section 4.2.1, which is that the smaller the system matrix difference is, the better the approximation is.

Furthermore, it also can be seen from Figure 5.6 that the system developed following Section 4.2.1 has the minimum state difference among all the tested tree approximations.

To demonstrate the application of tree approximation in disturbance attenuation, the disturbance propagation characteristics of the original system is studied via injecting a uniformly distributed random disturbance between -1 and 1 into subsystem 30. The states energy $\hat{\gamma}$ along all propagation paths in the best tree approximation under this disturbance is plotted in Figure 5.7. It is observed that the disturbance is not amplified, even though it is not monotonically decreasing along those paths when it is propagating in the network. Specifically, the peak value of the disturbance during propagation is more than 8 orders of magnitude smaller than that in the origin region.

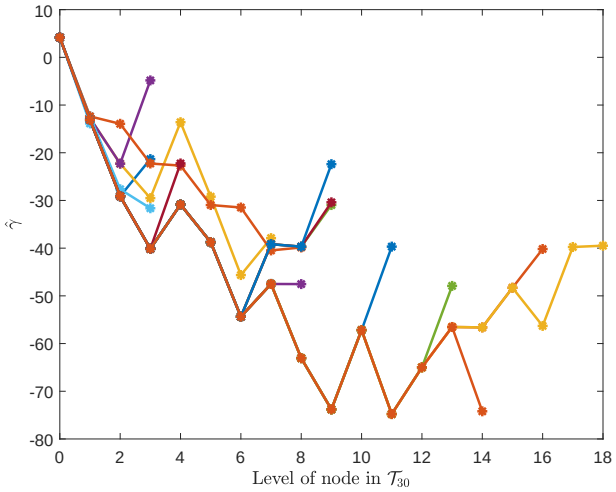


Figure 5.7: The states energy of the original system along all propagation paths in the best approximated tree (continuous-time systems).

5.2 Simulation for a discrete-time system: multi-region perimeter control

5.2.1 Model description of the traffic network

Multi-region perimeter control has been introduced in Section 2.3. In particular, a typical modelling method [74, 90, 91] is described in Section 2.3.1.

The perimeter control model used for this research is based on [74, 90, 91] with slight modifications. Particularly, the modifications are given as follows:

1. A partitioned traffic network with a geographical tree topology $\Gamma(\mathcal{V}, \mathcal{E})$ is considered here.

2. If a region is located at the periphery of the network, it is assumed that the exterior demand flow q_{ei} of region i is a constant value and is controlled by $\beta_{ei}(t)$.

3. Instead of assuming that the outflow is given by the MFD (2.14), a more realistic outflow description is adopted here:

$$q_{i,out}(t) = \sum_{h \in \mathcal{N}(i)} \beta_{ih}(t) O_i(n_i(t)), \tag{5.10}$$

where $\mathcal{N}(i)$ is the set of neighbouring regions of region i.

With the above modifications, the inflow of region i is given by

$$q_{i,in}(t) = r(i)\beta_{ei}(t)q_{ei} + \sum_{h\in\mathcal{N}(i)} \beta_{hi}(t)O_h(n_h(t)), \tag{5.11}$$

where $r(i) = 1$ if i is located at the periphery of the network, and $r(i) = 0$ otherwise.

By substituting (5.11) and (5.10) into (2.13), the system dynamics are presented in the following non-linear state equation:

$$\dot{n}_i(t) = r(i)\beta_{ei}(t)q_{ei} + \sum_{h\in\mathcal{N}(i)} \beta_{hi}(t)O_h(n_h(t)) - \sum_{h\in\mathcal{N}(i)} \beta_{ih}(t)O_i(n_i(t)) + \zeta_i(t), \tag{5.12}$$

where $\zeta_i(t) = \zeta_{i,in}(t) - \zeta_{i,out}(t)$.

Given desired operating points $\hat{\zeta}_i$, $\hat{\beta}_{hi}$, and $\hat{d}_i$ for each region $i \in \mathcal{V}$, the non-linear model (5.12) can be linearised around the operating point, which yields

$$\begin{aligned}\Delta\dot{n}_i(t) =& r(i)\Delta\beta_{ei}(t)q_{ei} + \sum_{h\in\mathcal{N}(i)} O_h(\hat{n}_h)\Delta\beta_{hi}(t) + \sum_{h\in\mathcal{N}(i)} \hat{\beta}_{hi}O_h'(\hat{n}_h)\Delta n_h(t) \\ & - O_i(\hat{n}_i)\sum_{h\in\mathcal{N}(i)} \Delta\beta_{ih}(t) - O_i'(\hat{n}_i)\Delta n_i(t)\sum_{h\in\mathcal{N}(i)} \hat{\beta}_{ih} + \Delta\zeta_i(t).\end{aligned} \tag{5.13}$$

The state equation (5.13) can be written in a vector form as follows

$$\Delta\dot{\boldsymbol{n}}(t) = \bar{\boldsymbol{A}}\Delta\boldsymbol{n}(t) + \bar{\boldsymbol{B}}\Delta\boldsymbol{\beta}(t) + \Delta\boldsymbol{\zeta}(t), \tag{5.14}$$

where $\Delta\boldsymbol{n}(t) \in \mathbb{R}^m$ is the state vector; $\Delta\boldsymbol{\beta}(t) \in \mathbb{R}^v$ is the control input vector; $\Delta\boldsymbol{\zeta}(t) \in \mathbb{R}^m$ is the disturbance vector; $\bar{\boldsymbol{A}} \in \mathbb{R}^{m\times m}$ is a tree-diagonal system matrix; $\bar{\boldsymbol{B}} \in \mathbb{R}^{m\times v}$ is the input matrix.

When multi-region perimeter controllers are applied to a traffic network, the control inputs $\Delta\boldsymbol{\beta}(t)$ are translated into signal settings of the traffic lights at the border of each region. The evolution of the real system cannot be faster than the cycle time of the traffic light. Thus, the continuous-time system description (5.14) of the multi-region traffic network may be translated to a discrete-time description, through first-order Euler discretization with sample time T, as follows

$$\Delta\boldsymbol{n}(k+1) = \tilde{\boldsymbol{A}}\Delta\boldsymbol{n}(k) + \tilde{\boldsymbol{B}}\boldsymbol{u}(k) + T\Delta\boldsymbol{\zeta}(k), \tag{5.15}$$

where k is the discrete-time index; $\tilde{\boldsymbol{A}} := \boldsymbol{I} + T\bar{\boldsymbol{A}}$ is the system matrix; $\tilde{\boldsymbol{B}} := T\bar{\boldsymbol{B}}$ is the input matrix; $\boldsymbol{u}(k) := \Delta\boldsymbol{\beta}(k)$ is the input vector for the discrete-time system. Note that the first-order Euler approximation is valid when $||\bar{\boldsymbol{A}}T^2|| \ll 1$.

Considering that the regions have different sizes and capacities, the analysis of disturbance propagation should be conducted upon the normalized variable, i.e. the relative accumulation error. The relative accumulation error $x_i(k)$ in region i is defined as the ratio of the accumulation error to the maximum accumulation error, i.e.

$$x_i(k) := \frac{\Delta n_i(k)}{n_{i,max} - \widehat{n}_i}. \tag{5.16}$$

The above linear transformation can be written in the following vector form

$$\boldsymbol{x}(k) := \boldsymbol{L}\Delta\boldsymbol{n}(k), \tag{5.17}$$

where $\boldsymbol{L} \in \mathbb{R}^{m\times m}$ is a diagonal matrix with non-zero diagonal elements $L_{ii} = \frac{1}{n_{i,max}-\widehat{n}_i}$, $\forall i = \mathcal{V}$. By substituting (5.17) into (5.15), the dynamics of the relative accumulation error can be captured by (4.24), where $\boldsymbol{A} = \boldsymbol{L}\tilde{\boldsymbol{A}}\boldsymbol{L}^{-1}$, $\boldsymbol{B} = \boldsymbol{L}\tilde{\boldsymbol{B}}$, and $\boldsymbol{d}(k) = T\boldsymbol{L}\Delta\boldsymbol{\zeta}(k)$. Therefore, the multi-region perimeter control problem for a large-scale traffic network is translated to a controller design problem as described in Problem 4.2.

Both tree-structured and arbitrarily connected transportation systems will be simulated, and a common set-up is given in the following section.

5.2.2 Simulation set-up of the traffic network

For the numerical results in this section, the MFD is based on Yokohama data [78]: $O(n) = (1.4877 \cdot 10^{-7}n^3 - 2.9815 \cdot 10^{-3}n^2 + 15.0912n)/3600$, and the traffic light cycle time T is set to be 60 seconds. To introduce different capacities of these regions, the MFD of each region i is randomly scaled from $O(n)$ as $O_i(n_i) = k_o(i)O(n_i/k_n(i))$, where $k_o(i)$ and $k_n(i)$ are random coefficients between 0.3 to 0.5.

The accumulation at the operating point is chosen to be 80% of the critical accumulation, i.e. $\hat{n}_i = 0.8n_i^*$. The exterior demand flow is assumed to be $q_{ei} = 0.4O_i(\hat{n}_i)$, and the control input $\hat{\beta}_{ei}$ is chosen to be 0.5 for all i. Furthermore, the control input at the operating point $\hat{\beta}_{ij}$ is selected by

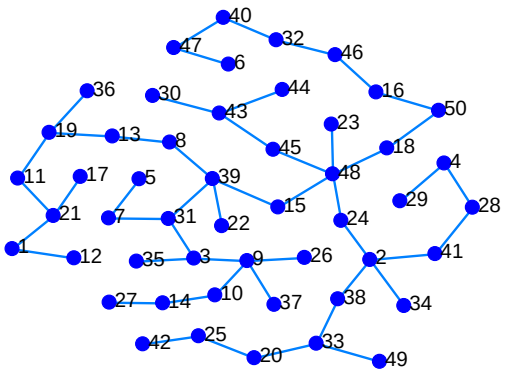


Figure 5.8: A 50-region traffic network with a geographical topology Γ'.

trial and error.

With the above parameters, system matrix $\boldsymbol{A}$ and input matrix $\boldsymbol{B}$ are determined.

It is assumed for all cases that the initial states are zero, i.e. $\boldsymbol{x}(0) = 0$, which implies that the accumulation of all the regions are $\hat{\boldsymbol{n}}$ at $k = 0$.

5.2.3 Simulation results of the traffic network

Validation of discrete-time disturbance attenuating controller design method

A traffic network partitioned into 50 regions is considered for the first simulation scenario. As shown in Figure 5.8, the network can be represented by a tree Γ'.[3]

A controller is designed following the controller synthesis method proposed in Section 4.1.2. Specifically, the control law is obtained by solving a convex optimisation problem with objective (4.26a) subject to constraints (4.26b), (4.26c), and (4.27), where α is chosen to be 0.8.

The disturbance attenuation property of the developed controller is demonstrated via injecting a uniformly distributed random disturbance between -1 and 1 in a region for 18000 seconds. Two scenarios are simulated here. In the first scenario, the disturbance occurs in an internal node 39, whilst in the second scenario it is injected in a leaf node 36.

Let $\mathcal{T}'_{39}$ be the rooted tree by treating node 39 in Γ' as the root. There exist 18 paths from root 39 to all the leaves in $\mathcal{T}'_{39}$. The energy of a state x_i is represented by $\gamma := log_{10}(\sum_{k=1}^{18000/T} x_i^2(k))$,

[3] It follows that the system matrix $\boldsymbol{A}$ is Γ' tree-diagonal.

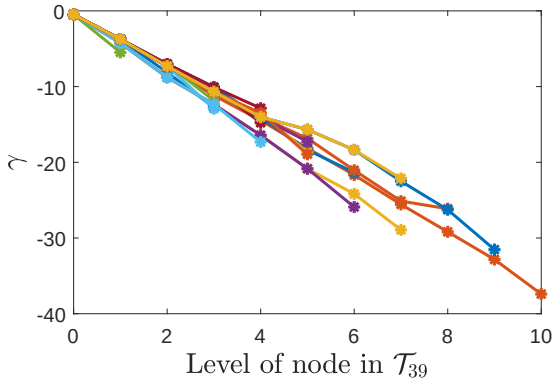


Figure 5.9: The energy of the states along all 18 paths in $\mathcal{T}'_{39}$ starting from node 39.

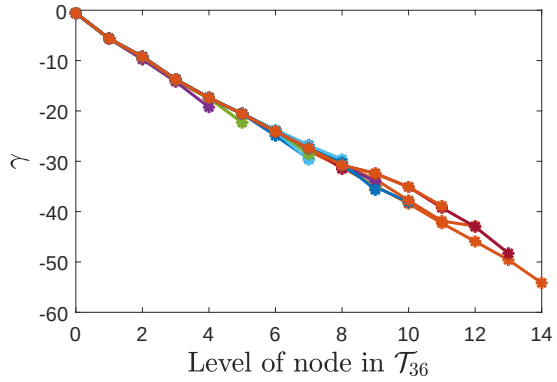


Figure 5.10: The energy of the states along all 17 paths in $\mathcal{T}'_{36}$ starting from node 36.

where logarithmic scale is used due to the rapid decrease of the states energy along each propagation path. The results for the energy following injection of the uniformly distributed disturbance in node 39 are plotted in Figure 5.9, where the x-axis is the level of a node i in the rooted tree $\mathcal{T}'_{39}$, and the y-axis is the energy γ.

Let $\mathcal{T}'_{36}$ be the rooted tree by treating node 36 in Γ' as the root. There are 17 paths from root 36 to all the leaves in $\mathcal{T}'_{36}$. Like Figure 5.9, the energy of the states along all disturbance propagation paths is shown in Figure 5.10.

Disturbance attenuation along all paths is demonstrated in both Figures 5.9 and 5.10. Thereby, the disturbance attenuating controller synthesis method in Section 4.1.2 is validated.

Table 5.2: Mean verification times (t) and standard deviation (σ) for discrete-time systems

Network size	Prop. 3.3		Prop. 3.4	
	t [s]	σ	t [s]	σ ($\times 10^{-5}$)
10	1.6×10^{-1}	0.01	8.6×10^{-6}	0.1
20	8.5×10^{-1}	0.07	1.8×10^{-5}	0.2
30	2.5×10^{0}	0.14	2.6×10^{-5}	0.4
40	6.5×10^{0}	0.54	4.0×10^{-5}	1.2
50	1.7×10^{1}	1.74	5.0×10^{-5}	1.4

Required computation time for evaluation of discrete-time sufficient disturbance attenuation conditions

Similar to continuous-time simulations, the time efficiency of the sufficient conditions in Proposition 3.4 is demonstrated through a comparison to conditions in Proposition 3.3.

We consider five groups of traffic networks with different number of partitions: 10, 20, 30, 40, and 50, and each group includes ten different arbitrarily generated tree topologies. Discrete-time disturbance attenuating controllers are designed for each network following the method proposed in Section 4.1.2. The disturbance attenuation property of the resulting closed-loop systems is verified through Propositions 3.3 and 3.4, respectively. The verification procedures are the same as in continuous-time simulation, except that they are tested by Propositions 3.3 and 3.4.

The verification results show that disturbance attenuation conclusions can be drawn for all tested networks by both Propositions. The mean verification times and corresponding standard deviation σ under each network size are given in Table 5.2. It can be observed in Table 5.2 that for each network size, disturbance attenuation verification is more time-efficient with Proposition 3.4. It also shows that verification time with Proposition 3.3 increases rapidly as the number of region grows, whilst time for Proposition 3.4 changes slowly. This numerical result demonstrates an advantage of the conditions in Proposition 3.4, as described in Remark 3.1.2, that they are more time-efficient than conditions in Proposition 3.3, and this advantage becomes more obvious when the network size is larger.

Simulation of tree approximation method in discrete-time systems

The tree approximation method proposed in Section 4.2 is verified through comparing a series of different tree approximations of a non-tree structured traffic network.

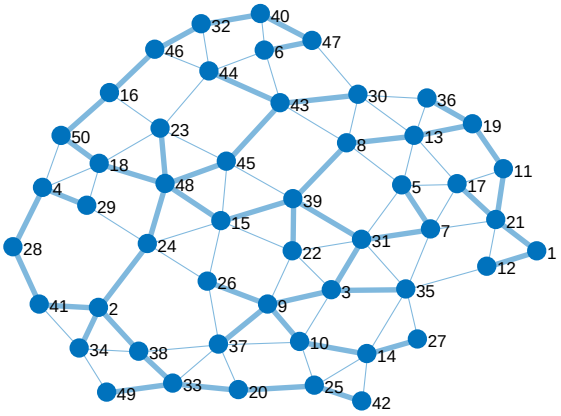


Figure 5.11: A 50-region traffic network $\mathcal{G}'$ with all interconnections (fine) and the maximum spanning tree (bold).

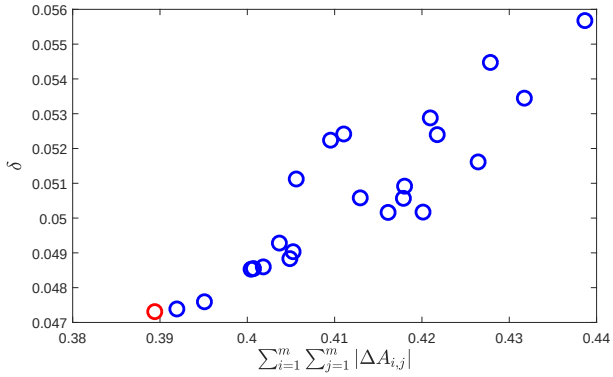


Figure 5.12: The relationship between the system matrix difference and the state difference in discrete-time systems.

Consider a perimeter controlled transportation system partitioned into 50 regions and the geographical topology $\mathcal{G}'$ is depicted in Figure 5.11. A tree-structured system is developed following the tree approximation method introduced in Section 4.2.2. According to Proposition 4.3, the tree approximation problem can be transformed into an MST problem in $\mathcal{G}'$. Here, we also use Prims algorithm [55] to find the MST, which is highlighted in Figure 5.11.

To test the efficacy of the proposed tree approximation method, we extract 22 other candidate trees and repeat the procedures and simulation set-ups in Section 5.1.3.

The performance of approximation is still indicated by the difference between the original system state $\boldsymbol{\xi}(k)$ and tree-structured system state $\boldsymbol{x}(k)$. Specifically, the difference is measured

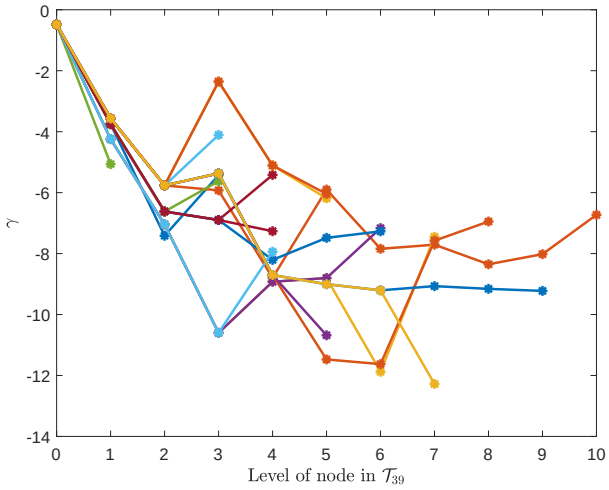


Figure 5.13: The states energy of the original system along all propagation paths in the best approximated tree (discrete-time systems).

by $\delta := \sqrt{\sum_{k=1}^{18000/T} \sum_{i=1}^{m} |\xi_i(k) - x_i(k)|^2}$. The relationship between the system matrix difference $\sum_{i=1}^{m} \sum_{j=1}^{m} |\Delta A_{i,j}|$ and the state difference δ is depicted in Figure 4.2.1, where each '$\circ$' corresponds to a pair of closed-loop tree approximation system and the corresponding original system. In particular, the system developed following the proposed tree approximation method in Section 4.2.2, is in red colour.

Similar conclusions can be made from Figure 5.12 as for Figure 5.6. Specifically, it can be seen in Figure 5.12 that the state difference between the original system and the tree approximation system increases with $\sum_{i=1}^{m} \sum_{j=1}^{m} |\Delta A_{i,j}|$. Moreover, it also can be observed that the system developed following Section 4.2.2 has the minimum state difference among all the tested tree approximations.

To investigate the application of tree approximation in disturbance attenuation, the disturbance propagation characteristics of the original system using the controller designed for the best approximated tree is studied. A uniformly distributed random disturbance between -1 and 1 is injected into node 30, and the results are plotted in Figure 5.13. It is shown that the disturbance is not amplified, even though it is not monotonically decreasing along those paths when it is propagating in the network. Specifically, the peak value of the disturbance during propagation is more than 3 orders of magnitude smaller than that in the origin subsystem.

5.3 Conclusions

In this chapter, simulations are conducted to validate the theoretical results in Chapters 3 and 4. The continuous-time results are verified through an interconnected pumped water supply system, and the discrete-time results is based on a traffic application.

Both continuous-time and discrete-time simulations include three aspects. First, a controller is designed for a 50-subsystem network following the synthesis method proposed in Section 4.1. By injecting a disturbance into a subsystem, it is shown that the system is able to attenuate disturbance along all propagation paths. Therefore, the attenuating controller synthesis method developed in Section 4.1 is validated. Second, the required computation time for evaluation of sufficient disturbance attenuation conditions is compared to that of necessary and sufficient conditions. The results show that sufficient conditions are more time efficient. Finally, the efficacy of the tree approximation method proposed in Section 4.2 is demonstrated via a 50-subsystem non-tree network. The performance of approximation is indicated by the state difference between the original non-tree system and the tree approximation. The tree-structured system obtained by following the method in Section 4.2 is compared to 22 other candidate spanning trees. It is observed from the simulation results that the system developed following Section 4.2 has the best approximation performance among all tested candidates. Furthermore, a disturbance attenuating controller is developed for the approximation tree and applied in the original non-tree network. The results show that the disturbance is not amplified in the non-tree network, and the peak value of the disturbance is sufficiently small compared to the subsystem where the disturbance originally occurs.

Chapter 6
Contributions and further work

6.1 Summary of contributions

This research has investigated disturbance attenuation in tree-structured interconnected systems, and an approach to relax the requirements of network topology to "tree-like" systems is proposed. In particular, the four research aims outlined in Section 2.4 have been achieved and lead to the following contributions:

1. Development of disturbance attenuation conditions

   (a) Necessary and sufficient conditions

   Based on Definitions 3.1 and 3.3, necessary and sufficient disturbance attenuation conditions are derived for an interconnected system with a tree topology. These conditions guarantee that a disturbance is attenuated when it is propagating in the network.

   (b) Sufficient conditions

   It is found that the verification of the proposed necessary and sufficient conditions requires global information. Furthermore, the number of operations with those conditions for a network of size m is at least $\mathcal{O}(m^3)$. When network size increases, the verification may become difficult due to limited communication and computation resources.

   Therefore, sufficient conditions are developed by taking advantage of the tree topology of the network. These sufficient conditions can be checked locally by each subsystem. Moreover, the verification is more time efficient, as only $\mathcal{O}(m^2)$ operations need to be taken.

2. Development of disturbance attenuating controllers

   The proposed sufficient disturbance attenuation conditions are incorporated into a controller design problem, in order to guarantee that the resulting closed-loop system attenuates disturbance.

   (a) Centralised controller design

   A centralised controller synthesis method is developed by formulating a convex optimisation problem. This problem considers minimum number of requirements for a tree-structured disturbance attenuating system: tree topology, disturbance attenuation, and stability. In particular, the disturbance attenuation constraint is guaranteed by a convex approximation of the proposed sufficient disturbance attenuation conditions.

   (b) Distributed controller design

   As discussed in Section 2.4, centralised design may not be possible, when the network is in large-scale and computation resources are limited. Therefore, the optimisation problem formulated for centralised controller design is broken up to subproblems, which can be solved locally by each subsystem through communication between neighbouring subsystems. The decentralisation of the problem is mainly facilitated by taking advantage of the sufficient disturbance attenuation conditions. Specifically, it is found that the stability condition requiring global information can be removed, as disturbance attenuation conditions which depend on local information can also guarantee asymptotic stability according to Gershgorin theorem.

3. Development of a tree approximation method

   This work proposes a tree approximation method, which is facilitated by extracting the MST from a non-tree network, such that given any controllers that accepts the selected tree structure, the state trajectories of the resulting non-tree closed-loop system are close to the tree system. This method can widen the scope of this research by applying Contributions 1 and 2 to more general cases.

4. Validation of results through numerical examples

   Water supply systems and multi-region perimeter controlled transportation networks are

simulated to verify the theoretical results proposed for continuous-time and discrete-time systems in this thesis, respectively. The simulation mainly includes three aspects, and similar conclusions are drawn for both examples:

(a) Validation of the attenuating controller design method

A controller is designed for a 50-subsystem network following the synthesis method in Contribution 2. It is shown that the resulting system is able to attenuate disturbance along all disturbance propagation paths.

(b) Demonstration of time efficiency of sufficient disturbance attenuation conditions

The time efficiency of the sufficient conditions is demonstrated via a comparison to necessary and sufficient conditions. The simulation results show that the sufficient conditions outperform the necessary and sufficient conditions in terms of verification time consumption, especially when the network size is large.

(c) Performance demonstration of the tree approximation method

A tree approximation system is obtained from a non-tree network following the tree approximation method, and its approximation performance is compared to various arbitrarily selected tree approximation candidates. The performance is indicated by the state difference between the original system and the approximation system. The simulation results show that the tree approximation system following the developed method outperforms other candidates under this performance metric.

6.2 Publications

The outcomes of this thesis are published or under review for publication in the following journals and conferences.

– **Y. Zhu**, I. Shames, C. Manzie "Stability analysis for distributed perimeter controlled traffic networks" In *American Control Conference (ACC), 2016*, pp. 809–814, Boston, USA, July 2016.

– Z. Hao, R. Boel, **Y. Zhu**, C. Manzie, S. Iman, Z. Li "Flexible hierarchical feedback control of urban traffic" In *EUROCAST*, pp. 294.

– **Y. Zhu**, I. Shames, C. Manzie "Disturbance Attenuation in Discrete-time Tree-Structured Interconnected Systems" In *Automatica*, under review.

6.3 Further work

Opportunities for further work, both theoretical and applied, are outlined in the following discussion.

1. Disturbance propagation analysis for high-order subsystems

   This research considers a network with first-order subsystems. The disturbance attenuation analysis is substantially simplified by the first-order state equation. Although such systems can be found in a large variety of applications, the disturbance attenuation results in this work can be extended to more general cases with high-order descriptions by generalising some of the ideas introduced in this research.

2. Further investigation of tree approximation

   In this research an approach is proposed to provide the best tree approximation for a non-tree network. From the simulation results, it is observed that the disturbance is not amplified in the original non-tree network for a given example. Further research could investigate that how to extend the proposed controller synthesis method to "tree-like" systems such that the non-tree system can attenuate disturbance.

3. Realistic traffic simulation environment

   In this work, the simulations are conducted in MATLAB/SIMULINK. In particular, the MFD is based on the observations from Yokohama and is captured by a polynomial function. However, as introduced in Section 2.3.1, the shape of MFD can be affected by signal settings and the infrastructure characteristics, etc. Therefore, it would be beneficial to implement the disturbance attenuation controllers in realistic traffic simulators such as SUMO or

AIMSUN. The performance of the controller could be demonstrated under more complex and realistic scenarios.

This page intentionally left blank.

Bibliography

[1] Feng Gao, DF Dang, SS Huang, and Shengbo Eben Li. Decoupled robust control of vehicular platoon with identical controller and rigid information flow. *International Journal of Automotive Technology*, 18(1):157–164, 2017.

[2] Shengbo Eben Li, Yang Zheng, Keqiang Li, Le-Yi Wang, and Hongwei Zhang. Platoon control of connected vehicles from a networked control perspective: Literature review, component modeling, and controller synthesis. *IEEE Transactions on Vehicular Technology*, 2017.

[3] Ionela Prodan, Laurent Lefevre, and Denis Genon-Catalot. Distributed model predictive control of irrigation systems using cooperative controllers. *IFAC-PapersOnLine*, 50(1):6564–6569, 2017.

[4] A Pawlowski, JA Sánchez-Molina, JL Guzmán, F Rodríguez, and S Dormido. Evaluation of event-based irrigation system control scheme for tomato crops in greenhouses. *Agricultural Water Management*, 183:16–25, 2017.

[5] Aniruddha Pant, Peter Seiler, and Karl Hedrick. Mesh stability of look-ahead interconnected systems. *IEEE Transactions on Automatic Control*, 47(2):403–407, 2002.

[6] Qi Chen and U Ozguner. A hybrid system model and overlapping decomposition for vehicle flight formation control. In *Proceedings of 42nd IEEE Conference on Decision and Control*, volume 1, pages 516–521, 2003.

[7] S Chang. Temporal stability of n-dimensional linear processors and its applications. *IEEE Transactions on Circuits and Systems*, 27(8):716–719, 1980.

[8] Paolo Massioni and Michel Verhaegen. Distributed control for identical dynamically coupled systems: A decomposition approach. *IEEE Transactions on Automatic Control*, 54(1):124–135, 2009.

[9] Peter Seiler, Aniruddha Pant, and Karl Hedrick. Disturbance propagation in vehicle strings. *IEEE Transactions on Automatic Control*, 49(10):1835–1842, 2004.

[10] Elaine Shaw and J Karl Hedrick. String stability analysis for heterogeneous vehicle strings. In *American Control Conference, 2007. ACC'07*, pages 3118–3125. IEEE, 2007.

[11] Gerrit JL Naus, Rene PA Vugts, Jeroen Ploeg, Marinus JG van de Molengraft, and Maarten Steinbuch. String-stable cacc design and experimental validation: A frequency-domain approach. *IEEE Transactions on Vehicular Technology*, 59(9):4268–4279, 2010.

[12] Laven Soltanian and Michael Cantoni. Achieving string stability in irrigation channels under distributed distant-downstream control. In *52nd Annual Conference on Decision and Control (CDC)*, pages 2169–2174. IEEE, 2013.

[13] Randal W Beard, Jonathan Lawton, and Fred Y Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9(6):777–790, 2001.

[14] Ben Yun, Ben M Chen, Kai-Yew Lum, and Tong H Lee. A leader-follower formation flight control scheme for uav helicopters. In *IEEE International Conference on Automation and Logistics*, pages 39–44, 2008.

[15] Herbert G Tanner, George J Pappas, and Vijay Kumar. Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3):443–455, 2004.

[16] Kai-ching Chu. Decentralized control of high-speed vehicular strings. *Transportation Science*, 8(4):361–384, 1974.

[17] Raffaello D'Andrea and Geir E Dullerud. Distributed control design for spatially interconnected systems. *IEEE Transactions on Automatic Control*, 48(9):1478–1495, 2003.

[18] D Swaroop and J Karl Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, 1996.

[19] J Eyre, D Yanakiev, and I Kanellakopoulos. A simplified framework for string stability analysis of automated vehicles. *Vehicle System Dynamics*, 30(5):375–405, 1998.

[20] Miguel Damas, AM Prados, F Gómez, and Gonzalo Olivares. Hidrobus® system: fieldbus for integrated management of extensive areas of irrigated land. *Microprocessors and Microsystems*, 25(3):177–184, 2001.

[21] Soledad Escolar Díaz, Jesús Carretero Pérez, Alejandro Calderón Mateos, Maria-Cristina Marinescu, and Borja Bergua Guerra. A novel methodology for the monitoring of the agricultural production process based on wireless sensor networks. *Computers and Electronics in Agriculture*, 76(2):252–265, 2011.

[22] Min-Hsiung Hung, DE Grin, and Kenneth J Waldron. Force distribution equations for general tree-structured robotic mechanisms with a mobile base. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 2711–2716, 1999.

[23] Karl Iagnemma, Adam Rzepniewski, Steven Dubowsky, and Paul Schenker. Control of robotic vehicles with actively articulated suspensions in rough terrain. *Autonomous Robots*, 14(1):5–16, 2003.

[24] Jinyan Shao, Guangming Xie, Junzhi Yu, and Long Wang. Leader-following formation control of multiple mobile robots. In *Proceedings of the 2005 IEEE International Symposium on Intelligent Control*, pages 808–813. IEEE, 2005.

[25] Dimos V Dimarogonas and Karl H Johansson. On the stability of distance-based formation control. In *In Proceedings of 47th IEEE Conference on Decision and Control*, pages 1200–1205, 2008.

[26] J Shao, G Xie, and L Wang. Leader-following formation control of multiple mobile vehicles. *IET Control Theory & Applications*, 1(2):545–552, 2007.

[27] Hiroki Kawakami and Toru Namerikawa. Cooperative target-capturing strategy for multivehicle systems with dynamic network topology. In *American Control Conference*, pages 635–640. IEEE, 2009.

[28] Yaqi Zhu, Iman Shames, and Chris Manzie. Stability analysis for distributed perimeter controlled traffic networks. In *American Control Conference (ACC), 2016*, pages 809–814. IEEE, 2016.

[29] Ali Ghasemi, Reza Kazemi, and Shahram Azadi. Stable decentralized control of a platoon of vehicles with heterogeneous information feedback. *IEEE Transactions on Vehicular Technology*, 62(9):4299–4308, 2013.

[30] Diana Yanakiev and Ioannis Kanellakopoulos. Nonlinear spacing policies for automated heavy-duty vehicles. *IEEE Transactions on Vehicular Technology*, 47(4):1365–1377, 1998.

[31] Fabio Morbidi, Patrizio Colaneri, and Thomas Stanger. Decentralized optimal control of a car platoon with guaranteed string stability. In *European Control Conference (ECC)*, pages 3494–3499. IEEE, 2013.

[32] Shengbo Li, Keqiang Li, Rajesh Rajamani, and Jianqiang Wang. Model predictive multi-objective vehicular adaptive cruise control. *IEEE Transactions on Control Systems Technology*, 19(3):556–566, 2011.

[33] Ronald Theodoor Van Katwijk. Multi-agent look-ahead traffic-adaptive control. *Doctoral dissertation, Delft University of Technology*, 2008.

[34] Lingyun Xiao and Feng Gao. Practical string stability of platoon of adaptive cruise control vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1184–1194, 2011.

[35] Cédric Langbort, Ramu Sharat Chandra, and Raffaello D'Andrea. Distributed control design for systems interconnected over an arbitrary graph. *IEEE Transactions on Automatic Control*, 49(9):1502–1519, 2004.

[36] Geir E Dullerud and Raffaello D'Andrea. Distributed control of heterogeneous systems. *IEEE Transactions on Automatic Control*, 49(12):2113–2128, 2004.

[37] Richard H Middleton and Julio H Braslavsky. String instability in classes of linear time invariant formation control with limited communication range. *IEEE Transactions on Automatic Control*, 55(7):1519–1530, 2010.

[38] Zheng Yang Gao Feng, Li Shengbo Eben and Kum Dongsuk. Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay. *IET Intelligent Transport Systems*, 10(7):503–513, 2016.

[39] Chehardoli Hossein and Ghasemi Ali. Adaptive centralized/decentralized control and identification of 1-d heterogeneous vehicular platoons based on constant time headway policy. *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[40] DV Swaroop. String stability of interconnected systems: An application to platooning in automated highway systems. *California Partners for Advanced Transit and Highways (PATH)*, 1997.

[41] Yang Zheng, Shengbo Eben Li, Jianqiang Wang, Dongpu Cao, and Keqiang Li. Stability and scalability of homogeneous vehicular platoon: Study on the influence of information flow topologies. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):14–26, 2016.

[42] van de Wouw Nathan Ploeg Jeroen, Shukla Dipan P and Nijmeijer Henk. Controller synthesis for string stability of vehicle platoons. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):854–865, 2014.

[43] Michael Cantoni, Erik Weyer, Yuping Li, Su Ki Ooi, Iven Mareels, and Matthew Ryan. Control of large-scale irrigation networks. *Proceedings of the IEEE*, 95(1):75–91, 2007.

[44] SM Melzer and BC Kuo. Optimal regulation of systems described by a countably infinite number of objects. *Automatica*, 7(3):359–366, 1971.

[45] Nader Motee and Ali Jadbabaie. Optimal control of spatially distributed systems. *IEEE Transactions on Automatic Control*, 53(7):1616–1629, 2008.

[46] Florian Dörfler, John W Simpson-Porco, and Francesco Bullo. Electrical networks and algebraic graph theory: Models, properties, and applications. *Proceedings of the IEEE*, 106(5):977–1005, 2018.

[47] Douglas Brent West. *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001.

[48] George N Korres, Peter J Katsikas, Kevin A Clements, and Paul W Davis. Numerical observability analysis based on network graph theory. *IEEE Transactions on Power Systems*, 18(3):1035–1045, 2003.

[49] Reinhard Diestel. *Graph theory {graduate texts in mathematics; 173}*. Springer-Verlag Berlin and Heidelberg GmbH & amp, 2000.

[50] Chris Godsil and Gordon F Royle. *Algebraic graph theory*, volume 207. Springer Science & Business Media, 2013.

[51] Abraham Berman and Robert J Plemmons. *Nonnegative matrices in the mathematical sciences*. SIAM, 1994.

[52] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 1990.

[53] Jaroslav Nešetřil, Eva Milková, and Helena Nešetřilová. Otakar borůvka on minimum spanning tree problem translation of both the 1926 papers, comments, history. *Discrete mathematics*, 233(1-3):3–36, 2001.

[54] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[55] Robert Clay Prim. Shortest connection networks and some generalizations. *Bell Labs Technical Journal*, 36(6):1389–1401, 1957.

[56] David R Karger, Philip N Klein, and Robert E Tarjan. A randomized linear-time algorithm to find minimum spanning trees. *Journal of the ACM (JACM)*, 42(2):321–328, 1995.

[57] Bernard Chazelle. The soft heap: an approximate priority queue with optimal error rate. *Journal of the ACM (JACM)*, 47(6):1012–1027, 2000.

[58] Jean-Pierre Corfmat and A Stephen Morse. Decentralized control of linear multivariable systems. *Automatica*, 12(5):479–495, 1976.

[59] Fabrizio Giulietti, Lorenzo Pollini, and Mario Innocenti. Autonomous formation flight. *IEEE Control Systems*, 20(6):34–44, 2000.

[60] Sinisa Todorovic and Michael C Nechyba. A vision system for intelligent mission profiles of micro air vehicles. *IEEE Transactions on Vehicular Technology*, 53(6):1713–1725, 2004.

[61] Zhongkui Li, Zhisheng Duan, Guanrong Chen, and Lin Huang. Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(1):213–224, 2010.

[62] Wei Ren. Consensus strategies for cooperative control of vehicle formations. *IET Control Theory & Applications*, 1(2):505–512, 2007.

[63] Yen-Chen Liu. Distributed synchronization for heterogeneous robots with uncertain kinematics and dynamics under switching topologies. *Journal of the Franklin Institute*, 352(9):3808–3826, 2015.

[64] Nandhini Sayeekumar, Shafeeque Ahmed, S Prabhakar Karthikeyan, Sarat Kumar Sahoo, and I Jacob Raglend. Graph theory and its applications in power systems-a review. In *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies*, pages 154–157. IEEE, 2015.

[65] Yang Wang, Liusheng Huang, Junmin Wu, and Hongli Xu. Wireless sensor networks for intensive irrigated agriculture. In *Consumer Communications and Networking Conference*, pages 197–201. IEEE, 2007.

[66] D Anurag, Siuli Roy, and Somprakash Bandyopadhyay. Agro-sense: Precision agriculture using sensor-based wireless mesh networks. In *Innovations in NGN: Future Network and Services. K-INGN 2008. First ITU-T Kaleidoscope Academic Conference*, pages 383–388. IEEE, 2008.

[67] Lars B Cremean and Richard M Murray. Stability analysis of interconnected nonlinear systems under matrix feedback. In *Proceedings of 42nd IEEE Conference on Decision and Control*, volume 3, pages 3078–3083. IEEE, 2003.

[68] Eddie Cheng and Marc J Lipman. Basic structures of some interconnection networks. *Electronic Notes in Discrete Mathematics*, 11:140–156, 2002.

[69] Herbert G Tanner, George J Pappas, and Vijay Kumar. Input-to-state stability on formation graphs. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 3, pages 2439–2444, 2002.

[70] Markos Papageorgiou, Christina Diakaki, Vaya Dinopoulou, Apostolos Kotsialos, and Yibing Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.

[71] Michael AP Taylor and Peter W Bonsall. *Understanding traffic systems: data analysis and presentation*. Routledge, 2017.

[72] Ana LC Bazzan and Franziska Klügl. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29(3):375–403, 2014.

[73] Tung Le, Péter Kovács, Neil Walton, Hai L Vu, Lachlan LH Andrew, and Serge SP Hoogendoorn. Decentralized signal control for urban road networks. *Transportation Research Part C: Emerging Technologies*, 58:431–450, 2015.

[74] Konstantinos Aboudolas and Nikolas Geroliminis. Perimeter and boundary flow control in multi-reservoir heterogeneous networks. *Transportation Research Part B: Methodological*, 55:265–281, 2013.

[75] Nikolas Geroliminis, Jack Haddad, and Mohsen Ramezani. Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):348–359, 2013.

[76] Mohammad Hajiahmadi, Jack Haddad, Bart De Schutter, and Nikolas Geroliminis. Optimal hybrid perimeter and switching plans control for urban traffic networks. *IEEE Transactions on Control Systems Technology*, 23(2):464–478, 2015.

[77] Haddad Jack Ramezani Mohsen and Geroliminis Nikolas. Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control. *Transportation Research Part B: Methodological*, 74:1–19, 2015.

[78] Nikolas Geroliminis and Carlos F Daganzo. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, 42(9):759–770, 2008.

[79] J. Godfrey. The mechanism of a road network. *Traffic Engineering and Control*, 1969.

[80] Robert Herman and Ilya Prigogine. A two-fluid approach to town traffic. *Science*, 204(4389):148–151, 1979.

[81] Siamak Ardekani and Robert Herman. Urban network-wide traffic variables and their relations. *Transportation Science*, 21(1):1–16, 1987.

[82] H Mahmassani, James C Williams, and R Herman. Performance of urban traffic networks. In *Transportation and Traffic Theory (Proceedings of the Tenth International on Transportation and Traffic Theory Symposium, Cambridge, Massachusetts), NH Gartner, NHM Wilson, editors, Elsevier*, pages 1–18, 1987.

[83] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: theory and practicea survey. *Automatica*, 25(3):335–348, 1989.

[84] Dirk Helbing. Derivation of a fundamental diagram for urban traffic flow. *The European Physical Journal B-Condensed Matter and Complex Systems*, 70(2):229–241, 2009.

[85] Amin Mazloumian, Nikolas Geroliminis, and Dirk Helbing. The spatial variability of vehicle densities as determinant of urban network capacity. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 368(1928):4627–4647, 2009.

[86] Nikolas Geroliminis and Jie Sun. Properties of a well-defined macroscopic fundamental diagram for urban traffic. *Transportation Research Part B: Methodological*, 45(3):605–617, 2011.

[87] Christine Buisson and Cyril Ladier. Exploring the impact of homogeneity of traffic measurements on the existence of macroscopic fundamental diagrams. *Transportation Research Record: Journal of the Transportation Research Board*, (2124):127–136, 2009.

[88] Jack Haddad and Arie Shraiber. Robust perimeter control design for an urban region. *Transportation Research Part B: Methodological*, 68:315–332, 2014.

[89] Carlos F Daganzo. Urban gridlock: Macroscopic modeling and mitigation approaches. *Transportation Research Part B: Methodological*, 41(1):49–62, 2007.

[90] Konstantinos Ampountolas and Anastasios Kouvelas. Real-time estimation of critical values of the macroscopic fundamental diagram for maximum network throughput. In *Transportation Research Board 94th Annual Meeting*, number 15-1779, 2015.

[91] Konstantinos Aboudolas and Nikolas Geroliminis. Feedback perimeter control for multi-region large-scale congested networks. In *2013 European Control Conference (ECC)*, pages 3506–3511. IEEE, 2013.

[92] Behnam Amini, Farideddin Peiravian, Morteza Mojarradi, and Sybil Derrible. Comparative traffic performance analysis of urban transportation network structures. *arXiv:1507.03612*, 2015.

[93] Yaqi Zhu, Iman Shames, and Chris Manzie. Disturbance attenuation in discrete-time tree-structured interconnected systems. *Automatica*, under review, 2018.

[94] DJ Klein. Treediagonal matrices and their inverses. *Linear Algebra and its Applications*, 42:109–117, 1982.

[95] Fu Lin, Makan Fardad, and Mihailo R Jovanovic. Augmented lagrangian approach to design of structured optimal state feedback gains. *IEEE Transactions on Automatic Control*, 56(12):2923–2929, 2011.

[96] Fu Lin, Makan Fardad, and Mihailo R Jovanović. Sparse feedback synthesis via the alternating direction method of multipliers. In *Proceeding of the 2012 American Control Conference*, pages 4765–4770, 2012.

[97] Francis R Bach. Structured sparsity-inducing norms through submodular functions. In *Advances in Neural Information Processing Systems*, pages 118–126, 2010.

[98] A. Zare, N. K. Dhingra, M. R. Jovanović, and T. T. Georgiou. Structured covariance completion via proximal algorithms. In *Proceedings of the 56th IEEE Conference on Decision and Control*, pages 3775–3780, Melbourne, Australia, 2017.

[99] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*, volume 2. Wiley New York, 2007.

[100] Ronald L Graham and Pavol Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7(1):43–57, 1985.

[101] Michael Held and Richard M Karp. The traveling-salesman problem and minimum spanning trees: Part ii. *Mathematical programming*, 1(1):6–25, 1971.

[102] Marzia Cescon and Erik Weyer. Modeling and identification of irrigation channel dynamics affected by wind. *IFAC-PapersOnLine*, 50(1):5386–5391, 2017.