Frequency-Domain Analysis and Design of Distributed Control Systems

Yu-Ping Tian, Southeast University, China

With the rapid development of micro-sensors, micro-motors, sensor networks and communication networks, spatially distributed control systems have attracted increasing attention. Internet congestion control and multi-robot coordination control are two typical examples of distributed control systems. In such systems, global collective behaviors emerge or global targets are reached through a distributed control strategy. In Frequency-Domain Analysis and Design of Distributed Control Systems, Yu-Ping Tian systematically covers distributed control to help readers solve the effects of delays on stability.

- The first book to introduce frequency-domain methods for the analysis of distributed control systems, covering:
  - Scalable stability criteria of networks of distributed control systems
  - Effect of heterogeneous delays on the stability of a network of distributed control systems
  - Stability of Internet congestion control algorithms
  - Consensus in multi-agent systems
  - Introduces control design problems related to frequency-domain analysis
  - PowerPoint slides with book figures available on Wiley Companion Website

This book is ideal for graduate students in control, networking and robotics studies. The book is also geared for researchers in the fields of control theory and networking, who are interested in learning and applying distributed control algorithms or frequency-domain analysis methods.

www.wiley.com/go/tian/fgda
FREQUENCY-DOMAIN ANALYSIS AND DESIGN OF DISTRIBUTED CONTROL SYSTEMS
FREQUENCY-DOMAIN ANALYSIS AND DESIGN OF DISTRIBUTED CONTROL SYSTEMS

Yu-Ping Tian
Southeast University, China
I dedicate this book to Ningning and Ouya for their love, trust and support.
3 Scalability in the Frequency Domain 67
  3.1 How the Scalability Condition is Related with Frequency Responses 67
  3.2 Clockwise Property of Parameterized Curves 71
  3.3 Scalability of First-Order Systems 76
    3.3.1 Continuous-Time System 76
    3.3.2 Discrete-Time System 79
  3.4 Scalability of Second-Order Systems 85
    3.4.1 System of Type I 85
    3.4.2 System of Type II 95
  3.5 Frequency-Sweeping Condition 103
    3.5.1 Stable Quasi-Polynomials 103
    3.5.2 Frequency-Sweeping Test 105
  3.6 Notes and References 108
  References 109

4 Congestion Control: Model and Algorithms 111
  4.1 An Introduction to Congestion Control 111
    4.1.1 Congestion Collapse 112
    4.1.2 Efficiency and Fairness 114
    4.1.3 Optimization-Based Resource Allocation 114
  4.2 Distributed Congestion Control Algorithms 116
    4.2.1 Penalty Function Approach and Primal Algorithm 116
    4.2.2 Dual Approach and Dual Algorithm 117
    4.2.3 Primal-Dual Algorithm 118
    4.2.4 REM: A Second-Order Dual Algorithm 118
  4.3 A General Model of Congestion Control Systems 119
    4.3.1 Framework of End-to-End Congestion Control under Diverse Round-Trip Delays 119
    4.3.2 General Primal-Dual Algorithm 122
    4.3.3 Frequency-Domain Symmetry of Congestion Control Systems 124
  4.4 Notes and References 126
  References 127

5 Congestion Control: Stability and Scalability 129
  5.1 Stability of the Primal Algorithm 129
    5.1.1 Johari–Tan Conjecture 129
    5.1.2 Scalable Stability Criterion for Discrete-Time Systems 131
    5.1.3 Scalable Stability Criterion for Continuous-Time Systems 135
  5.2 Stability of REM 138
    5.2.1 Scalable Stability Criteria 138
7.2 Double Integrator System with Diverse Input Delays and Interconnection Uncertainties 233
  7.2.1 Leader-Following Consensus Algorithm 233
  7.2.2 Consensus Condition under Symmetric Coupling Weights 235
  7.2.3 Robust Consensus under Asymmetric Perturbations 238
  7.2.4 Simulation Study 240
7.3 High-Order Consensus in High-Order Systems 243
  7.3.1 System Model 243
  7.3.2 Consensus Condition 245
  7.3.3 Existence of High-Order Consensus Solutions 249
  7.3.4 Constant Consensus 252
  7.3.5 Consensus in Ideal Networks 254
7.4 Integrator-Chain Systems with Diverse Communication Delays 255
  7.4.1 Matching Condition for Self-Delay 255
  7.4.2 Adaptive Adjustment of Self-Delay 255
  7.4.3 Simulation Study 257
7.5 Notes and References 265
References 266

Index 269
This book is devoted to the study of distributed control systems, a very promising research area given the current rapid development of technologies of micro-sensors, micro-motors, sensor networks and communication networks. The main feature of the book is to adopt a frequency-domain approach to cope with analysis and design problems in distributed control systems. Frequency-domain methods utilize frequency response properties of subsystems (agents) and communication channels to characterize the stability and other performance criteria of the entire system. By comparison with time-domain methods, it is usually assumed to be more convenient to use frequency-domain methods for analyzing the robustness of the system against noise and dynamic perturbations. It will be shown in this book that frequency-domain methods are also powerful in scalability analysis of distributed control system.

The book consists of three parts. The first part includes Chapters 1, 2 and 3, and describes common features and mathematical models of distributed control systems; it also introduces basic tools that are useful for further analysis and design of the underlying systems, such as graph theory, frequency-domain stability criteria, scalability analysis based on differential geometric properties of frequency response plots, etc. The second part includes Chapters 4 and 5, and the third part includes Chapters 6 and 7. The second part focuses on the distributed congestion control of communication networks while the third part studies the consensus control of multi-agent systems. The second and third parts can be considered as the application of the general theory introduced in the first part. However, the theory is also developed in the last two parts for two particular types of distributed control systems. Moreover, many interesting and beneficial results are obtained when general theory is applied to concrete systems.

It is quite common to choose the congestion control and the consensus control for the study of distributed control systems. There are at least two reasons. Firstly, both types of control schemes emphasize cooperation in a group of agents although the starting points of the cooperation are somehow different. For the congestion control, the cooperation is realized through distributed real-time optimization of some common performance indexes, i.e., allocation of limited resources in networks. For the consensus control, the basis of the cooperation is to manipulate states (or output) of agents to reach some common value. The two cooperation approaches are widely used in many other practical distributed control systems, such as multi-robot control systems, localization and information fusion sensor networks, etc. Secondly, many theoretical problems such as stability, scalability and delay effect encountered in these two kinds of systems can be treated by using a unified frequency-domain method. Although distributed real-time optimization problems in congestion control usually lead to nonlinear models, the local dynamics around an equilibrium of a congestion control system can be described as a linear distributed feedback control system with a bipartite interconnection topology graph, which essentially has the same structure as a multi-agent system controlled by
a consensus protocol. Nevertheless, consensus problems are different from congestion control problems in many aspects. For example, a multi-agent system driven by a consensus protocol has a continuum of equilibriums instead of an isolated equilibrium. This is perhaps why the Laplacian matrix of the interconnection topology graph plays a key role in consensus problems.

The time-delayed feedback control is introduced in this book as one of the basic design methods for distributed control systems. We study not only the effect of time delays, in particular various communication delays, on stability and scalability, but also the potential of time-delayed feedback control for enhancing the stability. This looks somewhat illogical. But in fact it is possible for time delay to have a dual character in feedback control systems, as in many other things in nature.

Some notions and results appearing in this book are new. They are mostly related to bipartite systems, symmetric systems, symmetric communication delays, commensurate self-delays, semi-stability test in the frequency domain, and high-order consensus. However, most of the material is based on the research results of the author with his collaborators and students, notably Jiandong Zhu, Hong-Yong Yang, Cheng-Lin Liu and Ya Zhang, to whom the author would like to express his sincere thanks. The author would also like to acknowledge the continuous support of the National Natural Science Foundation of China in the research topics of this book and in other related areas.

Parts of the book were taught on short courses listed below.

2. “Internet Congestion Control”, Summer school lecture at the School of Mechanics and Engineering Sciences, Peking University, Beijing, 2008.

The author would also like to thank Prof. Zhong-Ping Jiang of New York University, Prof. Lin Huang of Peking University and Prof. Guangrong Chen of City University of Hong Kong for their invitations to present these lectures, which impelled the author to collect related materials scattered in his own and others' research publications.

Yu-Ping Tian
Southeast University, Nanjing
Glossary of Symbols

:= "defined as"
≜ "denoted as"
A \ B Exclude set B from set A, where B ⊂ A
A ⊗ B Kronecker product of matrices A and B
A ⊕ B Direct sum of set A and B
(z)⁺ A piece-wise function of x ≥ 0, which takes value z if x > 0, or max(z, 0) if x = 0
R, Rⁿ, Rⁿ×ⁿ The set of real numbers, n component real vectors, and n by n real matrices
R⁺ The set of nonnegative real numbers
C, Cⁿ, Cⁿ×ⁿ The set of complex numbers, n component complex vectors, and n by n complex matrices
Cⁿ Space of functions which are n-times differentiable
C(ω) Curvature of a parametric curve at parameter ω
Re(s) The real part of complex number s
Im(s) The imaginary part of complex number s
C⁻ The open left half of the complex plane (briefly as LHP), {s ∈ C : Re(s) < 0}
C⁻ The closed left half of the complex plane (briefly as closed LHP), {s ∈ C : Re(s) ≤ 0}
C⁺ The closed right half of the complex plane (briefly as RHP), {s ∈ C : Re(s) ≥ 0}
C⁺ The open right half of the complex plane (briefly as open RHP), {s ∈ C : Re(s) > 0}
D The interior of the unit disc (briefly as IUD), {s ∈ C : |s| < 1}
GLOSSARY OF SYMBOLS

\( \tilde{D} \)  The closure of \( D \) (briefly as closed IUD), \( \{s \in \mathbb{C} : |s| \leq 1 \} \)

\( D^0 \)  The closed outer part of the unit disc (briefly as OUD), \( \{s \in \mathbb{C} : |s| \geq 1 \} \)

\( \tilde{D}^0 \)  The open outer part of the unit disc (briefly as open OUD), \( \{s \in \mathbb{C} : |s| > 1 \} \)

\( \mathbb{H}_{n \times n} \)  The set of \( n \) by \( n \) transfer function matrices \( G(s) \), with \( \|G\|_\infty < \infty \)

\( \mathbb{RH}_{n \times n} \)  The set of \( n \) by \( n \) rational transfer function matrices \( G(s) \), with \( \|G\|_\infty < \infty \)

\( \text{Co}(r_1, \ldots, r_n) \)  Convex hull of elements \( r_1 \cdots, r_n \)

\( n_{1\leq i \leq 2} \)  The set of integers from \( n_1 \) to \( n_2 \) satisfying \( n_1 \leq n_2 \), \( \{n_1, \ldots, n_2\} \)

\( \mathcal{S} \)  The index set of source nodes in a communication network, \( \{1, 2, \ldots, S\} \)

\( \mathcal{L} \)  The index set of link nodes in a communication network, \( \{1, 2, \ldots, L\} \)

\( A^\top \)  Transpose of matrix (vector) \( A \)

\( A^* \)  Conjugate transpose of matrix (vector) \( A \)

\( j \)  Unit of imaginary numbers, \( \sqrt{-1} \)

\( 1_n \)  \( n \times 1 \) vector \([1, \ldots, 1]^\top\)

\( e_1 \)  \( n \times 1 \) vector \([1, 0, \ldots, 0]^\top\)

\( \text{diag}(t_i, i \in 1, n) \)  Diagonal matrix with diagonal entries \( t_i, i \in 1, n \)

\( \lambda_i(A) \)  \( i \)-th eigenvalue of matrix \( A \in \mathbb{C}^{n \times n} \).

\( \sigma(A) \)  Spectrum of matrix \( A \in \mathbb{C}^{n \times n} \), \( \{\lambda_1(A), \ldots, \lambda_n(A)\} \)

\( \rho(A) \)  Spectral radius of matrix \( A \in \mathbb{C}^{n \times n} \), \( \max_{i \in 1, n} |\lambda_i(A)| \)

\( \bar{\sigma}(A) \)  Maximum singular value of matrix \( A \in \mathbb{C}^{n \times m} \), \( \left( \max_{i \in 1, n} \lambda_i(AA^*) \right)^{\frac{1}{2}} \)

\( \text{rank}(A) \)  Rank of matrix \( A \)

\( \text{span}(A) \)  The space spanned by all the columns of matrix \( A \)

\( [A](:, i) \)  The \( i \)-th column of matrix \( A \)

\( [A](:, i_1, i_2) \)  The matrix formed by \( i_1 \)-th to \( i_2 \)-th columns of matrix \( A \), where \( i_1 \leq i_2 \)
1

Introduction

Ruling a large country is like cooking a small fish.
—Lao Dan (580–500 BC), Tao Te Ching

This chapter is concerned with some common characteristics of distributed control systems, such as distributive interconnections, local control rules and scalability. Basic notions and results of algebraic graph theory are introduced as a theoretic foundation for modeling interconnections of subsystems (agents) in distributed control systems. Coordination control systems and end-to-end congestion control systems are also introduced as two typical kinds of distributed control systems. The main topics of this book are also highlighted in the introduction of these two kinds of systems.

1.1 Network-Based Distributed Control System

From the “flyball” speed governor of Watt’s steam engine to the control of communication systems including the telephone system, cell phones and the Internet, the control mechanism in industrial, social and many other real-world systems has experienced a development process from centralized control to distributed control. In the past decade, this process has been significantly speeded up thanks to rapid advances of communication techniques and their application to control systems.

Recently, workshops, seminars, short courses and even regular courses on distributed control systems emerge in the control society just like bamboo shoots after spring rain. But the following questions, which are often raised in seminars or lectures by post-graduate students, may also puzzle the reader of this book. What kind of control systems can be called distributed control system? Are there any differences between a distributed control system and a so-called large-scale control system or a decentralized control systems? What’s the advantage of distributed control over centralized control? etc. We will not attempt to answer all the questions; rather, we will try to grasp some common features of distributed control systems.

Cooperation among agents

Cooperation is perhaps the soul of a distributed control system. Since there is no centralized control unit, the only way through which all the agents in the system can work in harmony
is to conduct some kind of cooperation. Ignoring cooperation in biology might be one of a few shortcomings by which one could rebuke the great theory of Darwin (1859). Indeed, cooperation has been observed in many biological colonies, such as in birds, fish, bacilli and so on. Through cooperation they increase the probability of discovering food, get rid of prey and other dangers. Breder (1954) proposed a simple mathematic model to characterize the attracting and excluding actions in schools of fish. To describe the flocks and aggregations of schools more effectively, Reynolds (1987) proposed three simple rules: (1) collision avoidance, (2) agreement on velocity and (3) approaching center, which actually implies approaching any neighbor. He successfully simulated the motion of fish by using these rules. These investigations stimulate researchers to develop artificial systems that make decisions and take actions in distributive but cooperative manners. Nowadays, distributed control mechanisms have been widely used in engineering systems, such as aircraft traffic control (Tomlin, Pappas and Sastry 1998), multi-robot control (Rekleitis, Dudek and Milios 2000), coverage control of sensor networks (Qi, Iyengar and Chakrabarty 2001), formation control of unmanned aerial vehicles (UAVs) (Giulietti, Pollini and Innocenti 2000) etc.

**Spatially distributed interconnection**

In a classic feedback control system, the controller gets the output signal of the plant, compares it with some reference signal and makes decisions on how to act. Such a system also serves as one of basic units (subsystems) in a distributed control system. However, to cooperate with other units in the entire system, it should be equipped with a sensor/communication modular besides the classic decision/action modular (controller), as shown in Figure 1.1. Such a subsystem is sometimes called an agent. In a distributed control system, therefore, each subsystem gets not only the information of the output of itself but also the information of some other subsystems via sensor or/and communication networks (Figure 1.2). A distributed control system interconnected with multiple agents is also called a multi-agent system (MAS).

![Figure 1.1 An agent in distributed control system.](image)

For many distributed control systems such as the Internet, power grids, traffic control systems etc., subsystems (agents) are often distributed across multiple computational units in an immense space and connected through long-distance packet-based communications. In these system packet loss and delay are unavoidable, and hence, computational and communication
constraints cannot be ignored. New formalism to ensure stability, performance and robustness is required in the analysis and design of distributed control systems (Murray et al. 2003).

**Local control rule**

A distributed control law should be subject to the local control rule, which implies that in most cases there is no kind of centralized supervision or control unit in the system, and each agent makes decisions based only on the information received by its own sensor or from its neighboring agents through communication. The action rules proposed by Reynolds (1987) are typically local control rules. The local control rule is the most important feature of distributed control systems and makes the distributed control distinguished from the decentralized control of large-scale systems, which were extensively studied in the 1970s and 1980s (see, e.g., Sastry (1999) for a survey on large-scale systems). The decentralized control mechanism allows each agent to communicate perfectly with any other agent in the system, ignoring the computational and communication constraints. Moreover, the plant in decentralized control systems is usually a single tightly connected unit and not a rather loosely interconnected group of complete systems, which is the typical case in distributed control systems. Therefore, cooperation among agents in distributed control systems is much more important than in large-scale systems, and hence the coupling between a pair of subsystems can not be dealt with as a disturbance as in some decentralized control designs.

A remarkable advantage of the local control rule over the non-local control rules is its higher fault-tolerance capability. This is extremely important for many large-scale engineering systems which must operate continuously even when some individual units fail. Therefore, building a very reliable distributed control system with fault-tolerance ability from unreliable parts is a very promising research direction (Murray et al. 2003), although the topic is beyond the scope of this book.

**Scalability**

Scalability is perhaps another very important reason why distributed control systems prefer the local control rule. In this book the scalability of a distributed control system implies that the controller of the system and its maintenance utilize only local information around each agent and rarely depends on the scale of the system. In other words, by scalability we mean that not only the control law but also most important properties of the system rely on the local information. For example, in checking the stability of the entire distributed control system the
scalability requires that the stability criterion does not need to use information about the global interconnection topology of the system because such global information is usually unavailable to individual agents. A scalable system allows new applications to be designed and deployed without requiring changes to the underlying system.

The scalability of heterogeneous distributed control systems has drawn much more attention of researchers because most practical networked systems such as the Internet are heterogeneous. Diverse communication and/or input delays, different channel capacities, non-identical agent dynamics and so on, can make a system heterogeneous. Analysis and design of heterogeneous systems are much more difficult in comparison with homogeneous systems. One reason for that is the analysis and/or design of a large but homogenous systems can usually be treated as a task for a small system through diagonalization of the original system. But in most cases such a simplification method is not applicable to heterogeneous systems.

1.2 Graph Theory and Interconnection Topology

1.2.1 Basic Definitions

Graph theory plays a crucial role in describing the interconnection topology of distributed control systems. In this section we only present basic definitions about graph theory. For systematic study of graph theory we refer the reader to, for example, Biggs (1994); Bollobás (1998); Diestel (1997); Godsil and Royle (2001).

Digraph

A directed graph (in short, digraph) of order $n$ is a pair $G = (V, E)$, where $V$ is a set with $n$ elements called vertices (or nodes) and $E$ is a set of ordered pairs of vertices called edges. In other words, $E \subseteq V \times V$. We denote by $V(G)$ and $E(G)$ the vertex set and edge set of graph $G$, respectively, and denote by $\mathbb{1}_n := \{1, \ldots, n\}$ a finite set for vertex index. For two vertices $v_i, v_j \in V$, i.e., $i, j \in \mathbb{1}_n, i \neq j$, the ordered pair $(v_i, v_j)$ represents an edge from $v_i$ to $v_j$ and is also simply denoted by $e_{ij}$.

A digraph $G(V', E')$ is said to be a subgraph of a digraph $(V, E)$ if $V' \subset V$ and $E' \subset E$. In particular, a digraph $(V', E')$ is said to be a spanning subgraph of a digraph $(V, E)$ if it is a subgraph and $V' = V$. The digraph $(V', E')$ is the subgraph of $(V, E)$ induced by $V' \subset V$ if $E'$ contains all edges in $E$ between two vertices in $V'$.

Path and connectivity

A path in a digraph is an ordered sequence of vertices such that any ordered pair of vertices appearing consecutively in the sequence is an edge of the digraph. A path is simple if no vertices appear more than once in it, except possibly for the initial and final vertices. The length of a path is defined as the number of consecutive edges in the path. For a simple path, the path length is less than the number of vertices contained in the path by unity.

A vertex $v_i$ in digraph $G$ is said to be reachable from another vertex $v_j$ if there is a path in $G$ from $v_i$ to $v_j$. A vertex in the digraph is said to be globally reachable if it is reachable from every other vertex in the digraph. A digraph is strongly connected if every vertex is globally reachable. In Figure 1.3, $v_1, v_2, v_4, v_5$ are globally reachable vertices. But the digraph is not strongly connected because $v_3$ is unreachable from the other vertices.
Cycle and tree

A cycle is a simple path that starts and ends at the same vertex. A cycle containing only one vertex is called a self-cycle (or self-loop). The length of a cycle is defined as the number of edges contained in the cycle. A cycle is odd (even) if its length is odd (even). If a vertex in a cycle is globally reachable, then any other vertex in the cycle is also globally reachable. In Figure 1.3, the path \((v_1, v_2, v_5, v_1)\) is a cycle. The path \(\{v_2, v_4, v_5, v_2\}\) and the path \(\{v_1, v_2, v_4, v_5, v_1\}\) are also cycles. This digraph has no self-cycle. A digraph with self-cycle is shown in Figure 1.4.

Figure 1.3 A digraph.

Figure 1.4 A digraph with a self-cycle.
A digraph is acyclic if it contains no cycles. An acyclic digraph is called a directed tree if it satisfies the following property: there exists a vertex, called the root, such that any other vertex of the digraph can be reached by one and only one path starting at the root. A directed spanning tree of a digraph is a spanning subgraph that is a directed tree.

The digraph shown in Figure 1.5 is a directed tree. Obviously, it is a directed spanning tree of both the digraph in Figure 1.3 and the digraph in Figure 1.4.

**Neighbor and degree**

If \((v_i, v_j)\) is an edge of digraph \(G\), then \(v_j\) is an out-neighbor of \(v_i\), and \(v_i\) is an in-neighbor of \(v_j\). The set of out-neighbors (in-neighbors) of vertex \(v_i\) in digraph \(G(V, E)\) is denoted by \(N_i^{\text{out}}(G)\) (\(N_i^{\text{in}}(G)\)).

The out-degree (in-degree) of \(v_i\) is the cardinality of \(N_i^{\text{out}}\) (\(N_i^{\text{in}}\)). A digraph is topologically balanced if each vertex has the same in- and out-degrees. Note that neither \(N_i^{\text{out}}\) nor \(N_i^{\text{in}}\) contains the vertex \(i\) itself even if there is a self-loop at vertex \(i\).

Let us consider the example of a digraph shown in Figure 1.3. For vertex 1, its out-neighbor set is \(N_1^{\text{out}} = \{v_2\}\), and its in-neighbor set is \(N_1^{\text{in}} = \{v_3, v_5\}\). The out-degree and the in-degree of \(v_1\) are 1 and 2, respectively.

In this book, if the superscript is dropped in formulae or the prefix is omitted in texts, it will be referred to as the out-neighbor, i.e., \(N_i(G) = \{v_j \in V : (v_i, v_j) \in E\}\).

Sometimes we may use the term multi-level neighbor. If there is an \(m\)-length path in digraph \(G\) from vertex \(i\) to vertex \(j\), then vertex \(j\) is said to be an \(m\)-level neighbor of vertex \(i\). Of course, a neighbor in conventional sense is a 1-level neighbor. The set of \(m\)-level neighbors of agent \(i\) in digraph \(G\) is denoted by \(N_i^{m}(G)\). For example, in the digraph shown in Figure 1.3, \(N_1^{2}(G) = \{v_4, v_5\}\).
Digraph and information flow

In a distributed control system, each agent can be considered as a vertex in a digraph, and the information flow between two agents can be regarded as a directed path between the vertices in the digraph. Thus, the interconnection topology of a distributed control system can be described by a digraph. However, differing from the classic signal-flow graph (Chen 1984), in this book and in many other references on the distributed control system, the direction of an edge in the digraph does not mean the direction of an information flow. Let us consider the digraph shown in Figure 1.3 for an instance. Denote by $x_i \in \mathbb{R}, i \in \{1, 5\}$, the state of agent $i$ associated with vertex $i$. The existence of edge $e_{ij}$ implies that agent $i$ gets the state information $x_j$ from agent $j$. For example, agent 1 gets information from agent 2.

1.2.2 Graph Operations

We shall construct new graphs from old ones by graph operations.

For two digraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the intersection and union of $G_1$ and $G_2$ are defined by

$$G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2),$$

$$G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2).$$

For a digraph $G = (V, E)$, the reverse digraph of $G$ is a pair $\text{rev}(G) = (V, \text{rev}(E))$, where $\text{rev}(E)$ consists of all edges in $E$ with reversed directions.

If $W \subset V(G)$, then $G - W = G[V \setminus W]$ is the subgraph of $G$ obtained by deleting the vertices in $W$ and all edges incident with them. Obviously, $G - W$ is the subgraph of $G$ induced by $V \setminus W$. Similarly, if $E' \subset E(G)$, then $G - E' = (V(G), E(G) \setminus E')$. If $W$ (or $E'$) contains a single vertex $w$ (or a single edge $xy$, respectively), the notion is simplified to $G - w$ (or $G - xy$, respectively). Similarly, if $x$ and $y$ are non-adjacent vertices of $G$, then $G + xy$ is obtained from $G$ by joining $x$ to $y$.

Undirected graph

An undirected graph (in short, graph) $G$ consists of a vertex set $V$ and a set $E$ of unordered pairs of vertices. If each edge of the graph $G$ is given a particular orientation, then we get an oriented graph of $G$, denoted by $G^\rightarrow$, which is a digraph. Denote by $G^\leftarrow$ the reverse of $G^\rightarrow$. Then, $G = G^\rightarrow \cup G^\leftarrow$.

For an undirected graph $G$, the in-neighbor set of any vertex is always equal to the out-neighbor set of the same vertex. Therefore, in the undirected case we simply use the terminations neighbor, neighbor set and degree.

For an undirected graph, if it contains a globally reachable node, then any other vertex is also globally reachable. In that case we simply say that the undirected graph is connected.

For an undirected graph, it is said to be a tree if it is connected and acyclic.

**Theorem 1.1** $G(V, E)$ is a tree if and only if $G$ is connected and $|E| = |V| - 1$. Alternatively, $G(V, E)$ is a tree if and only if $G$ is acyclic and $|E| = |V| - 1$.

**Theorem 1.2** A graph is connected if and only if it contains a spanning tree.
Bipartite graph

A graph $G$ is a bipartite graph with vertex classes $V_1$ and $V_2$ if $V(G)$ is a direct sum of $V_1$ and $V_2$, i.e., $V = V_1 \oplus V_2$, which implies that $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$, and every edge joins a vertex of $V_1$ to a vertex of $V_2$. It is also said that $G$ has bipartition $(V_1, V_2)$. Figure 1.6 shows an example of a bipartite graph. For this graph, the vertex set $V$ is a direct sum of $V_1 = \{v_1, v_2, v_3\}$ and $V_2 = \{v_I, v_{II}, v_{III}, v_{IV}\}$. Each vertex in $V_1$ has neighbors only in $V_2$, and vice versa.

**Theorem 1.3** A graph is bipartite if and only if it does not contain an odd cycle.

For the bipartite graph $G$ with vertex classes $V_1$ and $V_2$, define

$$E'_{12} = \bigcup_{v_j \in V_2} \{(v_i, v_j) : v_i, v_j \in N_l\}, \quad (1.1)$$

$$E'_{21} = \bigcup_{v_i \in V_1} \{(v_i, v_j) : v_i, v_j \in N_r\}. \quad (1.2)$$

Note that in both (1.1) and (1.2), $v_i$ and $v_j$ can be the same vertex. In this case, $(v_i, v_i)$ is a self-loop. Let

$$G_1 = G - V_2 + E'_{12} \quad (1.3)$$

and

$$G_2 = G - V_1 + E'_{21}. \quad (1.4)$$

$G_1$ is a new graph obtained from $G$ by deleting all the vertices in $V_2$ and adding the new edges in $E'_{12}$. By definition (1.1), $E'_{12}$ is the set of new edges, each of which adds a self-loop to one vertex or joins two vertices in $V_1$ that are neighbors of a vertex in $V_2$. Similarly, $G_2$ is a new graph obtained from $G$ by deleting all the vertices in $V_1$ and adding the new edges in $E'_{21}$. 

![Figure 1.6](image_url)
Figure 1.7  Graph operation and equivalent graph for $V_1$.

$E'_{21}$ is the set of new edges, each of which adds a self-loop to one vertex or joins two vertices in $V_2$ that are neighbors of a vertex in $V_1$.

$G_1$ ($G_2$) is said to be the equivalent graph for $V_1$ ($V_2$) deduced from $G$. Obviously, neither $G_1$ nor $G_2$ is a subgraph of $G$. But the interconnection topology between any two vertices in one vertex class remains unchanged for $G$ and the equivalent graph $G_1$ (or $G_2$).

The graph operation defined by (1.3) for the bipartite graph shown in Figure 1.6 is illustrated by Figure 1.7 (a). Since only one edge is defined between any pair of vertices in graph, the equivalent graph for $V_1$ is given by Figure 1.7 (b). Similarly, the graph operation defined by (1.4) for the same bipartite graph is illustrated by Figure 1.8 (a). The equivalent graph for $V_2$ is given by Figure 1.8 (b).

Figure 1.8  Graph operation and equivalent graph for $V_2$. 
1.2.3 Algebraic Graph Theory

Algebraic graph theory studies matrices associated with digraphs.

Weighted digraph and adjacency matrix

A weighted digraph of digraph $G(V, E)$ is a triplet $G = (V, E, A)$, where $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ is an adjacency matrix satisfying

$$a_{ij} = \begin{cases} > 0, & \text{if } (v_i, v_j) \in E, \\ 0, & \text{otherwise}. \end{cases}$$

We denote by $A(G)$ the adjacency matrix of a weighted digraph $G$. Figure 1.9 shows an example of a weighted digraph. The adjacency matrix of the weighted digraph is

$$A = \begin{bmatrix} 0 & 3 & 1 & 4 & 0 \\ 0 & 0 & 0 & 3.5 & 0 \\ 1.5 & 2 & 0 & 6 & 0 \\ 1 & 0 & 0 & 0 & 6 \\ 3 & 0 & 0 & 0 & 1 \end{bmatrix}.$$  

Obviously, a weighted digraph is undirected if and only if $a_{ij} = a_{ji}$, that is, $A(G)$ is symmetric.  

If digraph $G$ contains no self-loop, then all the diagonal elements of its adjacency matrix $A$ are zero, i.e., $a_{ii} = 0, i \in \mathbb{I}_n$. However, sometimes we may encounter with graphs with self-loops. In this case, $a_{ii}$ may be positive numbers. To avoid confusion in terminology, by

---

1 If $A \in \mathbb{C}^{m \times n}$, under the symmetry, we mean the conjugate symmetry, i.e., $A^* = A$, where $A^*$ is the conjugate transpose of $A$. 

---

**Figure 1.9** A weighted digraph.
adjacency matrix we still mean the matrix $A$ with zero entries in the diagonal. Actually, $A(G)$ is associated with the digraph that is obtained by cutting off all the self-loops in $G$. We denote by $A(G)$ the matrix that characterizes the existence of all edges including self-loops in $G$, i.e.,

$$\tilde{A} = \text{diag}(a_{ii}, i \in \overline{1,n}) + A,$$

and refer to $\tilde{A}$ as the generalized adjacency matrix of $G$.

For a weighted digraph $G(V, E, \tilde{A})$ of order $n$, we use $\tilde{A}_{0,1} = (d_{ij})^{n \times n}$ to denote its un-weighted adjacency matrix, where

$$d_{ij} = \begin{cases} 1, & \text{if } a_{ij} > 0, \\ 0, & \text{if } a_{ij} = 0. \end{cases}$$

Let $G(V, E, \tilde{A})$ be a weighted digraph of $G(V, E)$ of order $n$. Given a matrix $\tilde{A}' = (a'_{ij})^{n \times n}$ with $a'_{ij} \in \mathbb{R}^+$, if

$$(\tilde{A}')_{0,1} = \tilde{A}_{0,1},$$

then, by the definition of adjacency matrix, $G(V, E, \tilde{A}')$ is also a weighted digraph of $G(V, E)$.

Naturally, a digraph $G = (V, E)$ can be considered as a weighted digraph with $\{0, 1\}$-weights, i.e.,

$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E, \\ 0, & \text{otherwise}. \end{cases}$$

The weighted out-degree matrix of digraph $G$, denoted by $D^{\text{out}}(G)$, is the diagonal matrix with the weighted out-degree of each node along its diagonal, i.e.,

$$D^{\text{out}}(G) = \text{diag}(d^{\text{out}}(v_i), i \in \overline{1,n}).$$

The weighted in-degree matrix of digraph $G$, denoted by $D^{\text{in}}(G)$, is the diagonal matrix with the weighted in-degree of each node along its diagonal, i.e.,

$$D^{\text{in}}(G) = \text{diag}(d^{\text{in}}(v_i), i \in \overline{1,n}).$$

While the adjacency matrix characterizes the location of edges among vertices in a digraph, the following result shows that powers of the adjacency matrix characterize the relationship between directed paths and vertices in the digraph.

**Lemma 1.4** Let $G(V, E, \tilde{A})$ be a weighted digraph of order $n$ possibly with self-loops. $\tilde{A}_{0,1}$ is its un-weighted adjacency matrix. Then, for all $i, j, k \in \overline{1,n},$

1. the $(i, j)$ entry of $\tilde{A}_{0,1}^k$ equals the number of directed paths of length $k$ (including paths with self-loops) from vertex $i$ to vertex $j$; and
2. the $(i, j)$ entry of $\tilde{A}^k$ is positive if and only if there exists a directed path of length $k$ (including paths with self-loops) from vertex $i$ to vertex $j$. 

Gain, measurement matrix and state-transfer matrix

Let $x_i(k) \in \mathbb{R}$, $i \in \{1, n\}$, be the state of agent $i$ associated with vertex $i$, where $k$ represents the time. By our stipulation of the physical meaning of the edge direction in a digraph, the existence of $e_{ij}$ implies that agent $i$ gets the state information $x_j$ from agent $j$. Then, the weight $a_{ij}$ associated with $e_{ij}$ can be regarded as the gain for the information flow. So, for a system with $G(V, E, A)$ as its interconnection topology digraph, the existence of $e_{ij}$ implies that the agent $i$ gets the amplified state information $a_{ij}x_j(k)$ from agent $j$. Let the measurement $y_i(k)$ of agent $i$ be equal to the sum of all the amplified states at the present time received by agent $i$, i.e.,

$$y_i(k) = a_{ii}x_i(k) + \sum_{j \in N_i} a_{ij}x_j(k). \quad (1.9)$$

Such a measurement will be also referred to as aggregated measurement. Sometimes, each agent can get only some relative measurement which can be expressed as

$$y_i(k) = a_{ii}x_i(k) - \sum_{j \in N_i} a_{ij}x_j(k). \quad (1.10)$$

Denote by $x(k) = [x_1(k), \cdots, x_n(k)]^T$ the state vector and $y(k) = [y_1(k), \cdots, y_n(k)]^T$ the measurement vector. Then, the matrix form of the aggregated measurement (1.9) is

$$y(k) = \bar{A}x(k). \quad (1.11)$$

Equation (1.11) provides an interpretation of the generalized adjacency matrix $\bar{A}$ from a viewpoint of system theory: it can be considered as a measurement matrix.

Suppose the state formation of each agent is updated by the following local law:

$$x(k+1) = Ky(k), \quad (1.12)$$

where $K = \text{diag}\{\kappa_i \in \mathbb{R}, i \in \{1, n\}\}$. Then, we have

$$x(k+1) = KA\bar{x}(k). \quad (1.13)$$

So, with the updating law (1.12), $K\bar{A}$ is a state-transfer matrix of a closed-loop system, as shown in Figure 1.10.

**Weighted bipartite graph**

Let $G(V, E, A)$ be a weighted bipartite graph with vertex classes $V_1 = \{v_1, \cdots, v_{n_1}\}$ and $V_2 = \{v_{n_1+1}, \cdots, v_n\}$. Then, by the definition of bipartite graph, the adjacency matrix

![Figure 1.10](image-url)
\( A(G) \) can be partitioned as
\[
A = \begin{bmatrix}
0 & A_{12} \\
A_{12}^T & 0
\end{bmatrix}, \tag{1.14}
\]
where \( A_{12} \in \mathbb{R}^{n_1 \times (n-n_1)} \). Hence,
\[
A^2 = \begin{bmatrix}
A_{12}A_{12}^T & 0 \\
0 & A_{12}^TA_{12}
\end{bmatrix}. \tag{1.15}
\]

By Lemma 1.4 we know that any vertex in \( V_1 \) has two-level neighbors in \( V_1 \), and any vertex in \( V_2 \) has two-level neighbors in \( V_2 \). Furthermore, the existence of paths of length 2 (including self-loops) in \( G \) between any pair of vertices in \( V_1 \) (or \( V_2 \)) is characterized by the sign of entries of \( A_{12}A_{12}^T \) (or \( A_{12}^TA_{12} \)). So, by the definition of equivalent graph, \( A_{12}A_{12}^T \) (or \( A_{12}^TA_{12} \)) can be defined as a weighted adjacency matrix associated with the equivalent graph of the bipartite graph \( G \) for \( V_1 \) (or \( V_2 \)). Summarizing the discussion we have the following proposition.

**Proposition 1.5** Let \( G_1(V_1, E_1) \) and \( G_2(V_2, E_2) \) be the equivalent graphs of the bipartite graph \( G(A) \) for \( V_1 \) and \( V_2 \), respectively, where \( A \) is given by (1.14). Then,
\[
\tilde{A}_1 = A_{12}A_{12}^T, \tag{1.16}
\]
\[
\tilde{A}_2 = A_{12}^TA_{12}, \tag{1.17}
\]
are weighted adjacency matrices associated with \( G_1 \) and \( G_2 \), respectively.

By the definition of un-weighted adjacency matrix,
\[
A_{0,1} = \begin{bmatrix}
0 & (A_{12})_{0,1} \\
(A_{12}^T)_{0,1} & 0
\end{bmatrix}. \tag{1.18}
\]
So, we have
\[
\tilde{A}_{0,1}^2 = \begin{bmatrix}
(A_{12})_{0,1}(A_{12}^T)_{0,1} & 0 \\
0 & (A_{12}^T)_{0,1}(A_{12})_{0,1}
\end{bmatrix}.
\]
So, by Lemma 1.4, the number of paths of length 2 (including self-loops) in \( G \) from vertex \( i \) to vertex \( j \) (\( i, j \in 1, n_1 \)) in \( V_1 \) equals the value of the \((i, j)\) entry of \((A_{12})_{0,1}(A_{12}^T)_{0,1}\); and the number of paths of length 2 (including self-loops) in \( G \) from vertex \( i \) to vertex \( j \) (\( i, j \in n-n_1, n \)) in \( V_2 \) equals the value of the \((i, j)\) entry of \((A_{12}^T)_{0,1}(A_{12})_{0,1}\). However, the signs of entries of \((A_{12})_{0,1}(A_{12}^T)_{0,1}\) (or \((A_{12}^T)_{0,1}(A_{12})_{0,1}\)) also characterize the existence of paths of length 2 (including self-loops) in \( G \) between any pair of vertices in \( V_1 \) (or \( V_2 \)) (see Exercise 1.7). So, by the definition of equivalent graph, the following proposition is also true.

**Proposition 1.6** Let \( G_1(V_1, E_1) \) and \( G_2(V_2, E_2) \) be the equivalent graphs of the bipartite graph \( G(A) \) for \( V_1 \) and \( V_2 \), respectively, where \( A \) is given by (1.14). Denote
\[
\tilde{A}_I = (A_{12})_{0,1}(A_{12}^T)_{0,1}, \tag{1.19}
\]
\[
\tilde{A}_{II} = (A_{12}^T)_{0,1}(A_{12})_{0,1}. \tag{1.20}
\]