Frank L. Lewis
Hongwei Zhang
Kristian Hengster-Movric
Abhijit Das

Cooperative Control of Multi-Agent Systems

Optimal and Adaptive Design Approaches

Springer

1

Cooperative control of multi-agent systems: optimal and adaptive design approaches

Frank L. Lewis, Hongwei Zhang, Kristian Hengster-Movric, and Abhijit Das

F.L. Lewis, H. Zhang, K. Hengster-Movric, A. Das, *Cooperative Control of Multi-Agent Systems: Optimal Design and Adaptive Control*, Springer-Verlag, 2013, to appear.

Dedications

To Galina, for her beauty and her wonderful frame of mind
Frank Lewis

To Lin and my parents, for their unconditional love
Hongwei Zhang

To my family, for all the support, love and kindness
Kristian Hengster-Movric

To my mother, because of whom I evaded the dark side of me
Abhijit Das

Preface

This book studies cooperative control of multi-agent dynamical systems interconnected by a communication network topology. In cooperative control, each system is endowed with its own state variable and dynamics. A fundamental problem in multi-agent dynamical systems on networks is the design of distributed protocols that guarantee consensus or synchronization in the sense that the states of all the systems reach the same value. The states could represent vehicle headings or positions, estimates of sensor readings in a sensor network, oscillation frequencies, trust opinions of each agent, and so on. In multi-agent systems, all systems should agree on the values of these quantities to achieve synchronized behavior.

Of fundamental concern for networked cooperative dynamical systems is the study of their interactions and collective behaviors under the influence of the information flow allowed in the communication network. This communication network can be modeled as a graph with directed edges or links corresponding to the allowed flow of information between the systems. The systems are modeled as the nodes in the graph and are sometimes called agents. Information in communication networks only travels directly between immediate neighbors in a graph. Nevertheless, if a graph is connected, then this locally transmitted information travels ultimately to every agent in the graph.

In cooperative control systems on graphs, there are intriguing interactions between the individual agent dynamics and the topology of the communication graph. The graph topology may severely limit the possible performance of any control laws used by the agents. Specifically, in cooperative control on graphs, all the control protocols must be *distributed* in the sense that the control law of each agent is only allowed to depend on information from its immediate neighbors in the graph topology. If enough care is not taken while designing the local agent control laws, the individual agent dynamics may be stable, but the networked systems on the graph may exhibit undesirable behaviors. Since the communication restrictions imposed by graph topologies can severely complicate the design of synchronization controllers, complex and intriguing behaviors are observed in multi-agent systems on graphs that do not occur in single-agent, centralized, or decentralized feedback control systems.

The study of networks of coupled dynamical systems arises in many fields of research. Charles Darwin showed that the interactions between coupled biological species over long time scales are responsible for natural selection. Adam Smith showed that the dynamical relationships between geopolitical entities are responsible for the balances in international finance and the wealth of nations. Distributed networks of coupled dynamical systems have received much attention over the years because they occur in many different fields including biological and social systems, physics and chemistry, and computer science. Various terms are used in literature for phenomena related to collective behavior on networks of systems, such as flocking, consensus, synchronization, frequency matching, formation, rendezvous, and so on. Collective synchronization phenomena occur in biology, sociology, physics, chemistry, and human engineered systems. The nature of synchronization in different groups depends on the manner in which information is allowed to flow between the individuals of the group.

The collective motions of animal social groups are among the most beautiful sights in nature. Each individual has its own inclinations and motions, yet the aggregate motion makes the group appear to be a single entity with its own laws of motion, psychology, and responses to external events. Flocks of birds, herds of animals, and schools of fish are aggregate entities that take on an existence of their own due to the collective motion instincts of their individual members. Collective motions allow the group to achieve what the individual cannot. Collective synchronized motion is a product not of planned scripts, but of instantaneous decisions and responses by individual members.

Analysis of groups based on social behaviors is complex, yet the individuals in collectives appear to follow simple rules. In many biological and sociological groups such as schools of fish, bird flocks, mammal herds on the move, and human panic behavior in emergency building evacuation, evidence supports the idea that the decisions made by all the individuals follow simple local protocols based on their nearest neighbors. The collective motion of large groups can be captured by using a few simple rules governing the behavior of the individuals. These rules depend on the awareness of each individual of its neighbors.

Mechanisms of information transfer in groups involve questions such as how information about required motion directions, originally held by only a few informed individuals, can propagate through an entire group by simple mechanisms that are the same for every individual. The information flow between members of a social group is instrumental in determining the characteristics of the combined motion of the overall group.

The engineering study of multi-agent cooperative control systems uses principles observed in sociology, chemistry, and physics to obtain synchronized behavior of all systems by using simple local distributed control protocols that are the same for each agent and only depend on that agent's neighbors in the group. Applications have been to oscillator synchronization, aircraft formations, mobile sensor area coverage, spacecraft attitude alignment, vehicle routing in traffic systems, containment control of moving bodies, and biological cell sorting.

Optimal feedback control design has been responsible for much of the successful performance of engineered systems in aerospace, manufacturing, industrial processes, vehicles, ships, robotics, and elsewhere since the 1960s. Optimal control designs generally require complete information of the system dynamics and rely on off-line solutions of matrix design equations. Adaptive control is a powerful method for the design of dynamic controllers that are tuned online in real time to learn stabilizing feedback controllers for systems with unknown dynamics. Many successful applications have been made in manufacturing and aerospace systems, and elsewhere.

In this book we use distributed cooperative control principles to design optimal control systems and adaptive control systems for multi-agent dynamics on graphs. These designs are complicated by the fact that all control protocols and parameter tuning protocols must be distributed in the sense that they depend only on immediate neighbors in the graph. Optimal control for cooperative multi-agent systems is considered in Part I of the book. Cooperative adaptive control is considered in Part II.

Chapter 1 of this book presents an overview of synchronization behavior in nature and social systems. It is seen that distributed decisions made by each agent in a group based only on the information locally available to it can result in collective synchronized motion of the overall group. The idea of a communication graph that models the information flows in a multi-agent group is introduced. Synchronization and collective behavior phenomena are discussed in biological systems, physics and chemistry, and engineered systems. Various different graph topologies are presented including random graphs, small world networks, scale-free networks, and distance formation graphs. The early work in cooperative control systems on graphs is outlined.

Chapter 2 introduces cooperative synchronization control of multi-agent dynamical systems interconnected by a fixed communication graph topology. Each agent or node is mathematically modeled by a dynamical linear time-invariant system. A review is given of graph basics and algebraic graph theory, which studies certain matrices associated with the graph. Dynamical systems on graphs are introduced. The idea of distributed control and the consensus problem are introduced. We begin our study with first-order integrator dynamics for continuous-time systems and discrete-time systems. Then, results are given for second order position-velocity systems which include motion control in formations. We present some key matrix analysis methods for systems on graphs that are important for the analysis and design of cooperative controllers.

In Part I of the book, which contains chapters 3 through 6, we study local and global optimal control for cooperative multi-agent systems linked to each other by a communication graph. In cooperative control systems on graphs it turns out that local optimality for each agent and global optimality for all the agents are not the same. The relations between stability and optimality are well understood for single-agent systems. However, there are more intriguing relations between stability and optimality in cooperative control than appear in the single-agent case, since local stability and global team stability are not the same, and local agent optimality and global team optimality are not the same. New phenomena appear that are not present for single-agent systems. Moreover, throughout everything the synchronization of the states of all agents must be guaranteed.

In Chapter 3 we study optimal control for continuous-time systems, and we shall see that local optimal design at each agent guarantees global synchronization of all agents to the same state values on any suitably connected digraph. Chapter 4 considers discrete-time systems and shows that an extra condition relating the local agent dynamics and the graph topology must be satisfied to guarantee global synchronization using local optimal design. Global optimization of collective group motions is more difficult than local optimization of the motion of each agent. A common problem in optimal decentralized control is that global optimization problems generally require global information from all the agents, which is not available to distributed controllers which can only use information from nearest neighbors. In Chapter 5 we shall see that globally optimal controls of distributed form may not exist on a given graph. To obtain globally optimal performance using distributed protocols that only depend on local agent information in the graph, the global performance index must be selected to depend on the graph properties in a certain way, specifically, through the graph Laplacian matrix. In Chapter 6 we define a different sort of global optimality for which distributed control solutions always exist on suitably connected graphs. There, we study multi-agent graphical games and show that if each agent optimizes its own local performance index, a Nash equilibrium is obtained.

In Part II of the book, which contains chapters 7 through 10, we show how to design cooperative adaptive controllers for multi-agent systems on graphs. These controllers allow synchronization of nonlinear systems where the agents have different dynamics. The dynamics do not need to be known and may have unknown disturbances. In adaptive controllers that are admissible for a prescribed communication graph topology, only distributed control protocols and distributed adaptive tuning laws are permitted. It is not straightforward to develop distributed adaptive tuning laws for cooperative agents on graphs that only require information from each agent and its neighbors. We show that the key to this is selecting special Lyapunov functions for adaptive control design that depend in specific ways on the graph topology. Such Lyapunov functions can be constructed using the concept of graph Laplacian potential, which depends on the communication graph topology.

In Chapter 7 we show that for networked multi-agent systems, there is an energy-like function, called the graph Laplacian potential, that depends on the communication graph topology. The Laplacian potential captures the notion of a virtual potential energy stored in the graph. The Laplacian potential is further used to construct Lyapunov functions that are suitable for the stability analysis of cooperative control systems on graphs. These Lyapunov functions depend on the graph topology, and based on them a Lyapunov analysis technique is introduced for cooperative multi-agent systems on graphs. Control protocols coming from such Lyapunov functions are distributed in form, depending only on information about the agent and its neighbors.

Chapter 8 covers cooperative adaptive control for systems with first-order nonlinear dynamics. The dynamics of the agents can be different, and they may be affected by disturbances. The dynamics may be unknown. A special Lyapunov function that depends on the graph topology is used to construct cooperative adaptive controllers wherein both the control protocols and the parameter tuning laws are distributed in the sense that they depend only on information available locally from the neighbors of each agent. Chapter 9 shows how to design adaptive controllers for nonlinear second-order multi-agent systems with position-velocity dynamics. Chapter 10 designs cooperative adaptive controllers for higher-order nonlinear systems with unknown dynamics and disturbances. The challenge of ensuring that both the control protocols and parameter tuning laws are distributed on the graph is confronted by using a Lyapunov function that involves two terms, one depending on the system dynamics and one depending on the communication graph topology.

Acknowledgements

Frank Lewis, Arlington, Texas
Hongwei Zhang, Chengdu, China
Kristian Hengster-Movric, Arlington, Texas
Abhijit Das, Ames, Iowa

Table of Contents

1

Introduction to Synchronization in Nature and Physics and Cooperative Control for Multi-agent Systems on Graphs

This chapter presents an overview of synchronization behavior in nature and social systems. It is seen that distributed decisions made by each agent in a group based only on the information locally available to it can result in collective synchronized motion of an overall group. The idea of a communication graph that models the information flows in a multi-agent group is introduced. Mechanisms are given by which decisions can be made locally by each agent and informed leaders can guide collective behaviors by interacting directly with only a few agents. Synchronization and collective behavior phenomena are discussed in biological systems, physics and chemistry, and engineered systems. The dependence of collective behaviors of a group on the type of information flow allowed between its agents is emphasized. Various different graph topologies are presented including random graphs, small world networks, scale-free networks, and distance formation graphs. ==The early work in cooperative control systems on graphs is outlined.==

   This chapter presents the fundamental ideas used in the rest of the book to develop analysis and design methods for cooperative control of multi-agent dynamical systems on graphs. It is seen that the basic ideas and philosophy of cooperative control systems rest upon phenomena that occur every day in natural and biological systems, physics, and chemistry.

1.1 Synchronization in Nature and Social Systems

The collective motions of animal social groups are among the most beautiful sights in nature. Each individual has its own inclinations and motions, yet the aggregate motion makes the group appear to be a single entity with its own laws of motion, psychology, and responses to external events. Flocks of birds, schools of fish, and herds of animals are aggregate entities that take on an existence of their own due to the collective motion instincts of their individual members (see Figure 1.1). The aggregate turns and wheels to achieve its objectives such as seeking food or migrating, and to avoid predators and obstacles. Such synchronized and

responsive motion makes one think of choreographed motions in a dance, yet they are a product not of planned scripts, but of instantaneous decisions and responses by individual members. In this section we study the mechanisms for such synchronized choreographic behavior of groups in terms of instantaneous decisions made by their individual members.

Collective motions allow the group to achieve what the individual cannot. Benefits of aggregate motion include defense from predators, social and mating advantages, and group foraging for food. In migrating bird flocks, the energy required by individual birds in flying is reduced by remaining in the wingtip vortex upwash of those ahead. Choreographed motions of groups of lions, wolves, and jackals allow more efficient pursuit of prey. In collective motion situations, the important entity becomes the group, not the individual. Such behavior has been observed in human crowd panic and mob situations, where there is pressure to follow the group's collective lead, not to think on one's own.

a. A flock of geese on migration. Courtesy of Bill Boaden

b. A flock of birds appears to be a single composite entity in its motion and behaviors. Courtesy of Eric Tofsrud Photography

c. A school of fish in synchronized motion. Courtesy of Russell Taylor

d. A school of fish responding as a single organism. Courtesy of Burak Dedeler

e. Wildebeest herd on migration. Courtesy of John Harrison

f. Wildebeest stampede to avoid danger. Courtesy of Jacqueline C. Baz

Fig. 1.1 The collective motion of animal social groups depends on individual member responses and is different for different species.

1.1.1 Animal Motion in Collective Groups

To reproduce the collective motion of an animal group in computer animation has been a challenge. It would be impossible to script the motion of each individual using planned motions or trajectories. Analysis of groups based on social behaviors is complex, yet the individuals in collectives appear to follow simple rules that make their motion efficient, responsive, and practically instantaneous. The cumulative motion of animal groups can be programmed in computer animation by endowing each individual with the same few rules that allow it to respond to the situations it encounters. The responses of the individuals accumulate to produce the combined motion of the group. The material in this section is from Reynolds [6].

Dependence of Group Motion on the Information Flow in Social Groups. In large social groups, each individual is aware only of the motions of its immediate neighbors. The field of perceptual awareness of the individual changes for different types of animal groups and in different motion scenarios. This results in different types of motion for different species. In flocking motion, birds are aware only of a few neighbors ahead of them or beside them. This results in the elegant, slow, wheeling motions of geese and ducks in migration. In schools of fish, shock waves transmitted by the water allow individuals to be aware instantaneously of motions of neighbors that they may not even see. This results in the quick, darting motions of fish schools that respond as a single entity. In animal herds, vibrations of the earth allow individuals to be aware of the general motion tendencies of others that may not be close by.

It is thus seen that the type of motion of the overall group depends on the manner in which information is allowed to flow between the individuals of the group. In well-connected social groups where each individual is aware of the motion of a greater number of neighbors, response speeds are generally faster.

Reynolds' Rules. The collective motion of large groups can be captured by using a few simple rules governing the behavior of the individuals. Individual motions in a group are the result of the balance of two opposing behaviors: a desire to stay close to the group, and a desire to avoid collisions with other individuals. Reynolds has captured the tendencies governing the motions of individuals through his three rules. These rules depend on the motions of the neighbors of each individual in the group.

Reynolds' Rules [6]:

1. Collision avoidance: avoid collisions with neighbors
2. Velocity matching: match speed and direction of motion with neighbors
3. Flock centering: stay close to neighbors

1.1.2 Communication Graphs and Implementing Reynolds' Rules

Reynolds' rules capture very well the collective motion of animal groups and can also be used as control schemes for human systems such as vehicle formations. These rules depend on the awareness of each individual of his neighbors. We have seen that the information flow between members of a social group is instrumental in determining the motion of the overall group. In this book we are concerned with the behaviors and interactions of dynamical systems that are interconnected by the links of a communication network. This communication network is modeled as a graph with directed edges corresponding to the allowed flow of information between the systems. The systems are modeled as the nodes in the graph and are sometimes called agents.
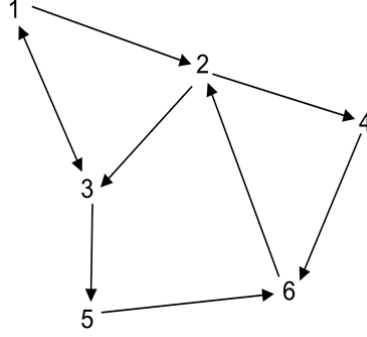
Fig. 1.2 A directed graph models the information flow between individuals of a social group.

Communication Graph. Figure 1.2 shows a representative graph that models the information flow interactions between six agents in a group. A *graph* is a pair $G = (V, E)$ with $V = \{v_1, \cdots, v_N\}$ a set of N nodes or vertices and E a set of edges or arcs. Elements of E are denoted as (v_i, v_j) which is termed an edge or arc from v_i to v_j, and represented as an arrow with tail at v_i and head at v_j. The in-degree of v_i is the number of edges having v_i as a head. The out-degree of a node v_i is the number of edges having v_i as a tail. The set of (in-) neighbors of a node v_i is $N_i = \{v_j : (v_j, v_i) \in E\}$, i.e. the set of nodes with edges incoming to v_i. The neighbor set N_i represents the nodes from which agent i receives information and in response to which it determines its own motion. The number of neighbors $|N_i|$ of node v_i is equal to its in-degree.

We have already intimated that the number of neighbors of each individual determines the speed of response and type of motion of the collective group. In subsequent chapters we shall study relationships between the graph topology and the resulting motion of each agent.

Associate with each edge $(v_j, v_i) \in E$ a weight $a_{ij} > 0$. Note the order of the indices in this definition. Then the neighbor set of node i can be written as $N_i = \{v_j : a_{ij} > 0\}$. The weights a_{ij} are selected to model the strength of the interaction between nodes. For instance, if agent j has higher social standing, then a_{ij} might be selected larger so that agent i is more responsive to the behaviors of agent j.

A graph is bi-directional if $a_{ij} \neq 0 \Rightarrow a_{ji} \neq 0$ so that communication between agents occurs bi-directionally. A graph is said to be undirected if $a_{ij} = a_{ji}, \forall i, j$, that is, if it is bi-directional and the weights of edges (v_i, v_j) and (v_j, v_i) are the same.

A directed path is a sequence of nodes $v_0, v_1, \cdots, v_r$ such that $(v_i, v_{i+1}) \in E, i \in \{0, 1, \cdots, r-1\}$. Node v_i is said to be connected to node v_j if there is a directed path from v_i to v_j. The distance from v_i to v_j is the length of the shortest path from v_i to v_j. Graph G is said to be strongly connected if v_i, v_j are connected for all distinct nodes $v_i, v_j \in V$. For bidirectional and undirected graphs, if there is a directed path from v_i to v_j, then there is a directed path from v_j to v_i, and the qualifier 'strongly' is omitted. If a graph is not connected, it is disconnected. A component of an undirected graph is a connected subgraph that is not connected to the remainder of the graph.

Information in social networks only travels directly between immediate neighbors in a graph. Nevertheless, if a graph is connected, then this locally transmitted information travels finally to every agent in the graph.

The diameter of a graph is the longest distance between two nodes. If the graph is disconnected, the diameter is infinity. The diameter is a global property of the graph.

Implementing Reynolds' Rules in Dynamical Motion Systems

There are many mechanisms for implementing Reynolds' rules in dynamical systems control. Of importance is the definition of an individual's 'neighborhood'. Consider motion in the R^2 and define $(p_i(t), q_i(t))$ as the position of node i in the (x, y)-plane. Define the state of agent i as $x_i = [p_i \quad q_i]^T \in R^2$. In this dis-

cussion, all interaction graphs are assumed undirected. That is, $a_{ij} = a_{ji}$ so that communication between agents occurs bidirectionally.

Agents seek to avoid collisions. Therefore, define a circle of radius ρ_c which each agent seeks to keep clear of other agents. Define the collision neighborhood for agent i as $N_i^c = \{ j : r_{ij} \le \rho_c \}$ where the distance between nodes i and j is

$$r_{ij} = \left| x_j - x_i \right| = \sqrt{(p_j - p_i)^2 + (q_j - q_i)^2} \tag{1.1}$$

Define also an interaction radius $\rho > \rho_c$ and the interaction neighborhood by $N_i = \{ j : r_{ij} \le \rho \}$. It is noted that radii ρ_c, ρ are different for different animal groups and different vehicles. Moreover, for some groups, such as flocks of birds in migration, the collision and interaction neighborhoods are not circular.

The dynamics used to simulate the individual group members can be very simple, yet realistic results are obtained. Consider agent motion in 2-D according to the dynamics

$$\dot{x}_i = u_i \tag{1.2}$$

with states $x_i = [p_i \quad q_i]^T \in R^2$. This is a simple point-mass dynamics with velocity control inputs $u_i = [u_{pi} \quad u_{qi}]^T \in R^2$.

A suitable law for collision avoidance is given by

$$u_i = - \sum_{j \in N_i^c} c_{ij} (x_j - x_i) \tag{1.3}$$

which causes agent i to turn away from other agents inside the collision neighborhood N_i^c.

A suitable law for flock centering is given by

$$u_i = \sum_{j \in N_i \setminus N_i^c} a_{ij} (x_j - x_i) \tag{1.4}$$

which causes agent i to turn towards other agents inside the interaction neighborhood N_i and outside the collision neighborhood N_i^c. The flock centering protocol can be written in terms of the components of velocity as

$$\dot{p}_i = u_{pi} = \sum_{j \in N_i \setminus N_i^c} a_{ij}(p_j - p_i) \qquad (1.5)$$

$$\dot{q}_i = u_{qi} = \sum_{j \in N_i \setminus N_i^c} a_{ij}(q_j - q_i) \qquad (1.6)$$

The net control velocity control u_i applied to the system is the sum of (1.3) and (1.4). The relation between the collision avoidance gains c_{ij} and the flock centering gains a_{ij} determines different response behaviors insofar as agents prefer to flock or prefer to avoid neighbors. The relation between the radii ρ, ρ_c is also instrumental in designing different behaviors.

Distributed Control Protocols on Communication Graphs. The control protocols (1.4)-(1.6) are known as *local voting protocols* because each agent seeks to make the difference between his state and those of his neighbors equal to zero. That is, each agent seeks to achieve consensus with its neighbors. These protocols are *distributed* in the sense that they depend only on the local neighbor information as allowed by the communication graph topology.

Alternative protocols for collision avoidance and flock centering are given respectively by

$$u = -\sum_{j \in N_i^c} c_{ij} \frac{(x_j - x_i)}{r_{ij}} = -\sum_{j \in N_i^c} c_{ij} \frac{(x_j - x_i)}{|x_j - x_i|} \qquad (1.7)$$

$$u = \sum_{j \in N_i \setminus N_i^c} a_{ij} \frac{(x_j - x_i)}{r_{ij}} = \sum_{j \in N_i \setminus N_i^c} a_{ij} \frac{(x_j - x_i)}{|x_j - x_i|} \qquad (1.8)$$

which are normalized by dividing by the distance between agents. Note that the sum is over components of a unit vector. Therefore, these laws prescribe a desired direction of motion and result in motion of uniform velocity. By contrast, the laws (1.3), (1.4) give velocities that are larger if one is further from one's neighbors.

These protocols guarantee consensus behavior in terms of (x, y) position, yet do not include velocity matching control. Simple motions of a group of N agents in the (x, y)-plane can alternatively be described by the node dynamics

$$\dot{p}_i = V_i \cos \theta_i$$
$$\dot{q}_i = V_i \sin \theta_i$$

(1.9)

where V_i is the speed and θ_i the heading of agent i. Velocity matching can be achieved by using the local voting protocol to reach heading consensus according to

$$\dot{\theta}_i = \sum_{j \in N_i} a_{ij} (\theta_j - \theta_i)$$

(1.10)

and a second local voting protocol to match speeds according to

$$\dot{V}_i = \sum_{j \in N_i} a_{ij} (V_j - V_i)$$

(1.11)

As a third alternative, one can use the Newton's law agent dynamics

$$\dot{x}_i = v_i$$
$$\dot{v}_i = u_i$$

(1.12)

with vector position $x_i \in R^n$, speed $v_i \in R^n$, and acceleration input $u_i \in R^n$. This is more realistic than (1.2) for motion of physical bodies in n dimensions. For motion in the 2-D plane, one would take $x_i = [p_i \quad q_i]^T$ where $(p_i(t), q_i(t))$ is the position of node i in the (x, y)-plane.

Consider the distributed position/velocity feedback at each node given by the second-order local neighborhood protocols

$$u_i = c \sum_{j \in N_i} a_{ij} (x_j - x_i) + c\gamma \sum_{j \in N_i} a_{ij} (v_j - v_i)$$
$$= \sum_{j \in N_i} c a_{ij} \left((x_j - x_i) + \gamma (v_j - v_i) \right)$$

(1.13)

where $c > 0$ is a stiffness gain and $c\gamma > 0$ is a damping gain. This is based on local voting protocols in both position and velocity so that each node seeks to match all its neighbors' positions as well as their velocities. Thus, this protocol realizes Reynolds' rules for flock centering and velocity matching. A protocol such as (1.3) could be added for collision avoidance.

Alternatively, offsets could be added for desired relative separation as in

$$u_i = c \sum_{j \in N_i} a_{ij} \left(x_j - x_i - \Delta_{ji} \right) + c\gamma \sum_{j \in N_i} a_{ij} (v_j - v_i) \qquad (1.14)$$

where $\Delta_{ji} \in R^n$ are the desired offsets of node i from node j. This law contains both flock centering and collision avoidance aspects.

Alternative methods to local cooperative protocols for implementing Reynolds' rules include potential field approaches [35],[36],[37]. Potential field methods include approaches for obstacle avoidance and goal seeking, and are intimately related to the topics in this section.

1.1.3 Leadership in Animal Groups on the Move

We have seen that information can be transferred locally between individual neighboring members of animal groups, yet result in collective synchronized motions of the whole group. Local motion control protocols are based on a few simple rules that are followed by all individuals. However, in many situations, the whole group must move towards some goal, such as along migratory routes or towards food sources. In these cases, only a few informed individuals may have pertinent information about the required directions of motion. In this section we will study how a small percentage of informed individuals can have a global impact that results in control of the whole group towards desired goals. The role of different leaders was discussed in [17].

The material in this section is from Couzin et al. [16]. Some species have evolved specialized mechanisms for conveying information about location and direction of goals. One example is the waggle dance of the honeybee that recruits hive members to visit food sources. However, in many groups such as schools of fish, bird flocks, and mammal herds on the move evidence supports the idea that the decisions made by all individuals follow simple local protocols based on nearest neighbors similar to Reynolds' Rules. Mechanisms of information transfer in groups involve questions such as how information about required motion directions, originally held by only a few informed individuals, can propagate through an entire group by simple mechanisms that are the same for every individual. It is not clear how individuals recognize leaders who are informed, or even if this is required, or how conflict is resolved when several informed individuals differ in their motion preferences.

Protocols for Leadership by a Small Percentage of Informed Individuals

The setup used in Couzin et al. [16] is as follows. Consider a set of N agents $\{i \in N\}$ and let the communication between the agents be modeled by a general directed graph. The edges of the graph correspond to agents that are close together in some sense depending on the specific animal group considered, as discussed

in the previous section. Each agent has a position vector, a direction vector, and a speed of motion.

One dynamics that captures this for motion in the (x, y)-plane is the discrete-time dynamics

$$
\begin{aligned}
p_i(k+1) &= p_i(k) + V_i(k)T\cos\theta_i(k) \\
q_i(k+1) &= q_i(k) + V_i(k)T\sin\theta_i(k)
\end{aligned}
\tag{1.15}
$$

where V_i is the speed of agent i, θ_i its heading, and $x_i = [p_i \quad q_i]^T$ its position. The time index is k and the sampling period is T. This is the discrete-time version of (1.9). The velocity vector of agent i is

$$
v_i(k) = \begin{bmatrix} V_i(k)\cos\theta_i(k) & V_i(k)\sin\theta_i(k) \end{bmatrix}^T
\tag{1.16}
$$

Each agent turns away from others in a collision avoidance region N_i^c according to

$$
\overline{\theta}_i(k+1) = -\sum_{j\in N_i^c} c_{ij} \frac{(x_j(k)-x_i(k))}{\left|x_j(k)-x_i(k)\right|}
\tag{1.17}
$$

which causes agent i to turn away from other agents inside the collision neighborhood. If agents are not detected within the collision avoidance region, then each agent will turn towards and align with its neighbors in an interaction region N_i according to

$$
\overline{\theta}_i(k+1) = \sum_{j\in N_i \setminus N_i^c} a_{ij} \frac{(x_j(k)-x_i(k))}{\left|x_j(k)-x_i(k)\right|} + \sum_{j\in N_i \setminus N_i^c} a_{ij} \frac{v_j(k)}{\left|v_j(k)\right|}
\tag{1.18}
$$

This is converted to a unit vector by

$$
\hat{\theta}_i(k+1) = \frac{\theta_i(k+1)}{\left|\theta_i(k+1)\right|}
\tag{1.19}
$$

A small percentage p of the group consists of informed individuals i, each of whom have a preferred desired direction of motion given by a unit reference direc-

tion vector r_i. The final heading control protocols for (1.15) are therefore taken as

$$\theta_i(k+1) = \frac{\hat{\theta}_i(k+1) + g_i r_i}{\left|\hat{\theta}_i(k+1) + g_i r_i\right|} \tag{1.20}$$

with $g_i \geq 0$ a leader weighting gain. If $g_i = 0$, agent i is not informed, exhibits no preferred direction of travel, and merely aligns with his neighbors. If $g_i > 0$, agent i is an informed individual and, as g_i increases, agent i exhibits a stronger desire to move in its reference direction r_i.

1.2 Networks of Coupled Dynamical Systems in Nature and Science

The study of networks of coupled dynamical systems arises in many fields of research. Charles Darwin [1] showed that the interactions between coupled biological species over long time scales are responsible for natural selection. Adam Smith [2] showed that the dynamical relationships between geopolitical entities are responsible for the balances in international finance and the wealth of nations.

In this section we look at some collective synchronization phenomena in biology, sociology, physics, chemistry, and human engineered systems. The nature of synchronization in different groups depends on the manner in which information is allowed to flow between the individuals of the group. Therefore, we also study several different types of graph topologies which support different types of information flows and have different basic properties. These different topologies result in different sorts of synchronization behaviors. Finally, we give a literature survey of cooperative control of multi-agent dynamical systems on communication graphs, which is the topic of interest in this book.

1.2.1 Collective Motion in Biological and Social Systems, Physics and Chemistry, and Engineered Systems

Distributed networks of coupled dynamical systems have received much attention over the years because they occur in many different fields including biological and social systems, physics and chemistry, and computer science. Various terms are used in literature for phenomena related to collective behavior on networks of systems, such as flocking, consensus, synchronization, formation, rendezvous, and so

on.

In many biological and sociological groups such as schools of fish, bird flocks, mammal herds on the move, and human panic behavior in emergency building evacuation, evidence supports the idea that the decisions made by all individuals follow simple local protocols based on their nearest neighbors. Mechanisms of information transfer in groups involve questions such as how information about required motion directions, originally held by only a few informed individuals, can propagate through an entire group by simple mechanisms that are the same for every individual.

Reynolds developed a distributed behavioral model for animal flocks, herds, and Schools [6]. He presented three rules by which decisions are made by agents using information only from their nearest neighbors. Reynolds Rules are: Collision avoidance, avoid collisions with neighbors; Velocity matching, match speed and direction of motion with neighbors; and Flock centering: stay close to neighbors. Couzin et al. [16] studied leadership and decision-making in animal groups on the move. They showed that by using protocols based only on local neighbor information, a small percentage of informed individuals can have a global impact that results in control of the whole group towards desired goals. The role of different types of leaders was studied in [17].

Axelrod [5] studied the evolution of cooperation in social groups. Strogatz and Stewart [9] studied coupled oscillator effects in biological synchronization. Wasserman and Faust [10] provided analysis methods for social networks. Helbing et al. [15] showed that local neighbor responses can accurately model the panic behavior of crowds during emergency building evacuation. The spread of infectious diseases in structured populations was studied by Sattenspiel and Simon [7]. Metabolic stability in random genetic nets was studied by Kauffman [3].

Kuramoto [4] analyzed local coupling in populations of chemical oscillators to study waves and turbulence. He showed that there is a critical information coupling coefficient above which coupled oscillators synchronize. The dynamics of a ring of pulse-coupled oscillators were studied by Bresslof et al. [14]. Results on frequency-locking dynamics of disordered Josephson arrays were given by Wiesenfeld [13]. In 1995, Vicsek et al. [12] reported interesting results of collective behaviors of self-driven particles in phase transition. They showed that by using nearest neighbor interaction rules, all agents eventually move in the same direction.

The ability to coordinate agents is important in many real-world decision tasks where it is necessary for agents to exchange information with each other. Distributed local decision and control algorithms are often desirable due to their computational simplicity, flexibility, and robustness to the loss of single agents. Parallel problem solving was studied based on phenomena in nature by Das et al. [11].

In the past few decades, an increasing number of industrial, military, and consumer applications have called for the coordination of multiple interconnected agents. Research on behaviors of networked cooperative systems, or multi-agent systems, and distributed decision algorithms has received extensive attention due

to their widespread applications in spacecraft, unmanned air vehicles (UAVs) [20], multi-vehicle systems [22], mobile robots, multipoint surveillance [19], sensor networks [21], networked autonomous teams, and so on. See [18], [19] for surveys of engineering applications of cooperative multi-agent control systems.

1.2.2 Graph Topologies and Structured Information Flow in Collective Groups

In large social groups, each individual is aware only of the motions of its immediate neighbors. The field of perceptual awareness of the individual changes for different types of animal groups and in different motion scenarios. This results in different types of motion for different species. In flocking motion, birds are aware only of a few neighbors ahead of them or beside them. This results in the elegant, slow, wheeling motions of geese and ducks in migration. In schools of fish, shock waves transmitted by the water allow individuals to be aware instantaneously of motions of neighbors that they may not even see. This results in the quick, darting motions of fish schools that respond as a single entity.

It is thus seen that the type of collective motion of the overall group depends on the manner in which information is allowed to flow between the individuals of the group. In well-connected social groups where each individual is aware of the motion of a greater number of neighbors, response speeds are generally faster.

In this section we study several different types of graph topologies which support different types of information flows and have different basic properties. These different topologies result in different sorts of synchronization behaviors. We shall discuss random graphs, small world networks, scale free networks, and distance graphs. A review of graph topologies is given by Strogatz [23].

Random Graphs

The growth, evolution, and properties of random graphs were studied by Erdos and Renyi [24] and Bollobas [25]. Random graphs are a class of undirected graphs.

In random graphs, there are N agents that are initially disconnected. It is desired to form m edge links between these agents in a random fashion. This is accomplished by selecting two agents at random, and with probability p forming an edge between them. This is repeated until all of the m edges have been disbursed.

Phase Transitions. The key factor in the growth of random graphs is the relationship between the number of agents N and the allowed number of edges m. It is found that there exists a threshold value m^* such that for $m > m^*$ a large connected component of the graph emerges. The value of this threshold is shown to be $m^* = N/2$. This giant cluster contains most of the N nodes in the graph. It

is connected in the sense that one can move from any agent in the cluster to any other agent in the cluster by following the graph edges.

A sample random graph topology is shown in Figure 1.3. The graph agents have been reorganized to show connected the components. A giant connected cluster is clearly seen in this figure.

Phase transitions are of extreme importance in graph theory. They capture transitions between phases, for instance, in chemical and physical systems, as well as the transition between normal behavior and panic behavior in human crowds.

Note that a random graph is connected if $m \geq \ln(N)$ [25].

Fig. 1.3. Random graph showing a giant connected component. Courtesy of[23].
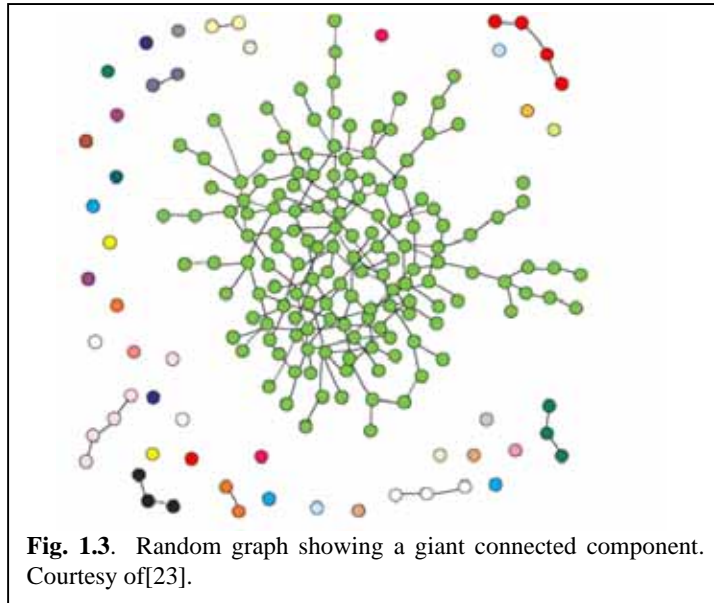
Small World Networks

Random graphs represent one extreme of graph organization, namely complete disorganization. Regular lattices represent another extreme where the graph is completely organized. An undirected graph is regular if all nodes have the same number of neighbors. Small world networks provide a method for generating graph topologies that interpolate between these two extremes. The material here is from [26].

In human social groups we can define a graph by friendship links. That is, an edge exists between two individuals if they are friends. In today's global society, it is observed that, remarkably, any two individuals selected at random worldwide

are connected by a path made up of friendship links. Moreover, the length of this friendship path is generally equal to six edges. That is to say, two people who meet at random can find a path of length six that connects them together through pairs of mutual friends. Now, with the advent of Facebook and Twitter, this friendship shortest path is no doubt decreasing.

Regular graphs and random graphs cannot capture the small world connectivity phenomenon of social networks. To construct networks with the small world property, Watts and Strogatz proceeded as follows [26]. Start with a regular ring lattice. Cycle through all the edges once, with probability p rewiring one end of each edge to a node selected uniformly at random from all edges in the graph. Repeated edges are not allowed.

Figure 1.4 shows the graphs resulting from this procedure for various values of the rewiring probability p. It is seen that the small world graphs form a continuum between the regular ring lattices and the random graphs that is a function of p.
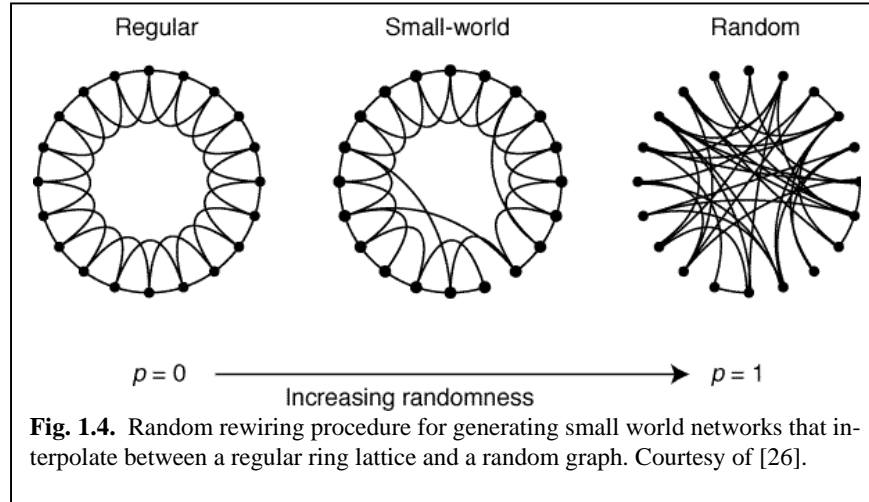
Fig. 1.4. Random rewiring procedure for generating small world networks that interpolate between a regular ring lattice and a random graph. Courtesy of [26].

To study the properties of small world networks two concepts are defined. Define the characteristic path length $L(p)$ as the number of edges in the shortest path connecting any two vertices, averaged over all pairs of vertices. The characteristic path length is a global quantity related to the graph diameter. The clustering coefficient captures the local connectivity properties of the graph. Suppose a node has k neighbors. Then these k neighbors have at most $k(k-1)/2$ edges between them, which occurs when all the neighbor pairs have edges connecting them. Let ℓ be the number of edges that actually exist between the neighbors. The fraction of edges that actually exist out of all possible edges is $\ell/(k(k-1)/2)$. Then

the clustering coefficient is defined as the average of this ratio over all nodes in the graph.

In terms of human social groups, $L(p)$ is the average number of friendships in the shortest chain connecting two people. Clustering coefficient $C(p)$ measures how many friends of an individual are also friends of each other, and hence captures the cliquishness of the group.

Figure 1.5 shows the change in characteristic path length and clustering coefficient as a function of rewiring probability p. It is seen that with even a few rewired edges, the path length decreases dramatically, whereas the clustering coefficient remains approximately the same. This means that a few rewired edges result in a few long hops that decrease the friendship distance between nodes in the graph, while the local friendship structures in the graph are preserved.

The fast change of characteristic path length with p is a *phase transition* phenomenon of the sort discussed above in connection with random graphs.

Fig. 1.5. Characteristic path length $L(p)$ and clustering coefficient $C(p)$ for small world networks as a function of rewiring probability p. Courtesy of [26].
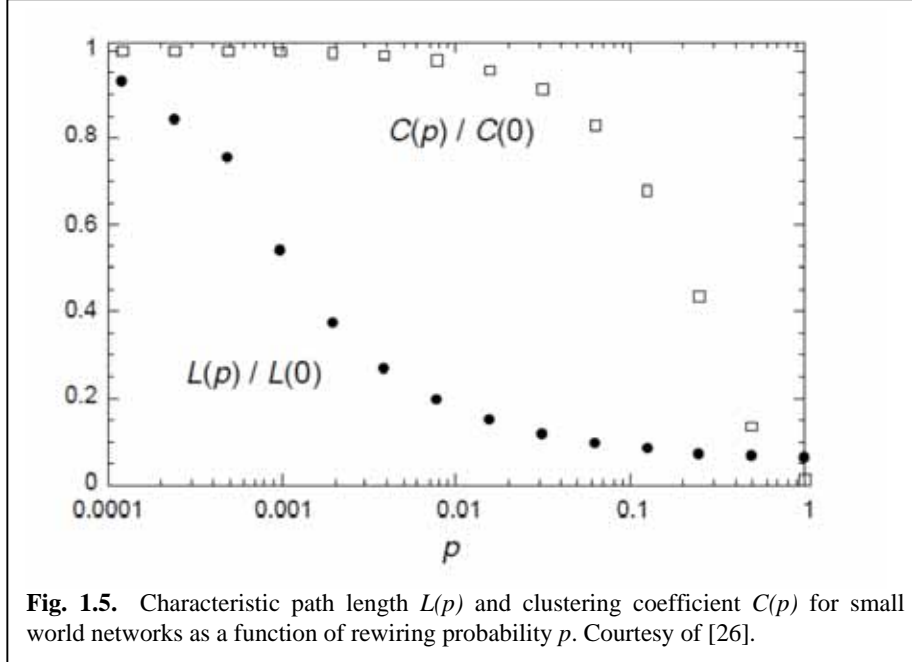
It is shown in [26] that both the collaboration graph of movie actors and the western United States electric power grid are Small World networks. Moreover, small world networks explain the spread patterns of infectious diseases in humans.

Homogeneity Property of Random Graphs and Small World Networks. In both the random graphs and the small world networks, most nodes have the same numbers of neighbors. This is known as the homogeneity property and expresses

the fact that the nodes are anonymous in the sense that they have the same neighborhood structures. The number of neighbors of a node in an undirected graph is known as its degree. The degree distribution $P(k)$ is the probability that a node has degree k in a given graph. For random graphs, the degree distribution is Poisson. Figure 1.6 shows Poisson distributions for some representative values of the defining parameter λ. It is seen that the probability of having k neighbors decreases quickly either above or below the mean value.

Fig 1.6. Poisson degree distribution for random graphs, showing $P(k)$, the probability that a node has degree k, vs. k.
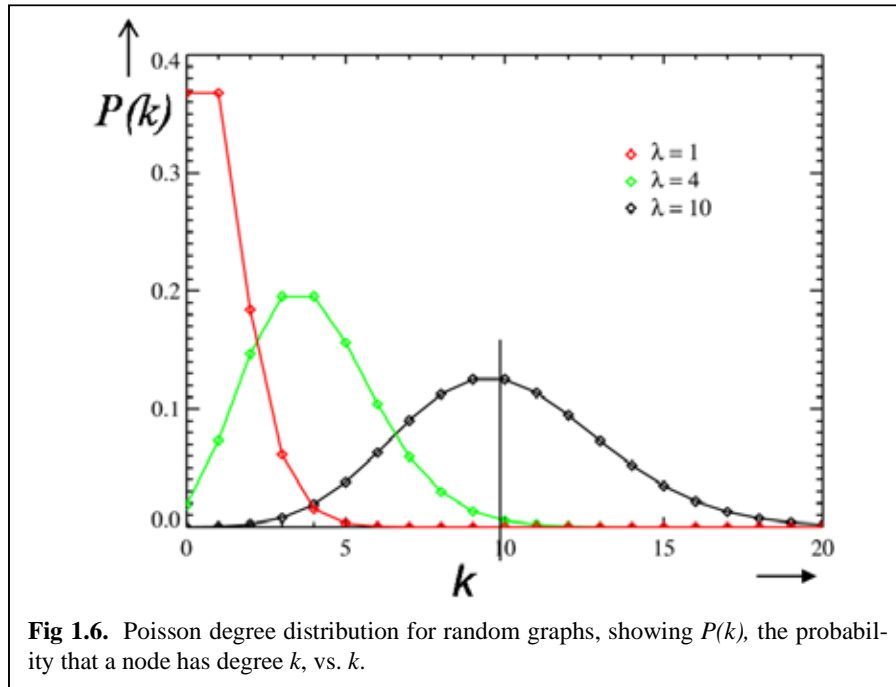
Scale-Free Networks

In random graphs and small world networks, the nodes are homogeneous in the sense that most nodes have the same number of neighbors. That is, the nodes are anonymous in terms of their degrees. Figure 1.7 shows the domestic route map of China Southern Airlines in China. This network has an immediately visible feature that is not explained by random graphs or small world networks. Namely, there are a few large hubs that are directly connected to each other, while most nodes are connected only to these large hubs.

In human social networks, it is usually the case that there are a few very influential individuals with many friends, with most individuals having fewer friends.

The degree, or number or friends, can be interpreted as the social standing of an individual. In these networks the nodes are not anonymous, but a few are more influential.

Fig. 1.7. Domestic route map for China Southern Airlines, showing a few well-connected large hubs. Courtesy of GCW Consulting.
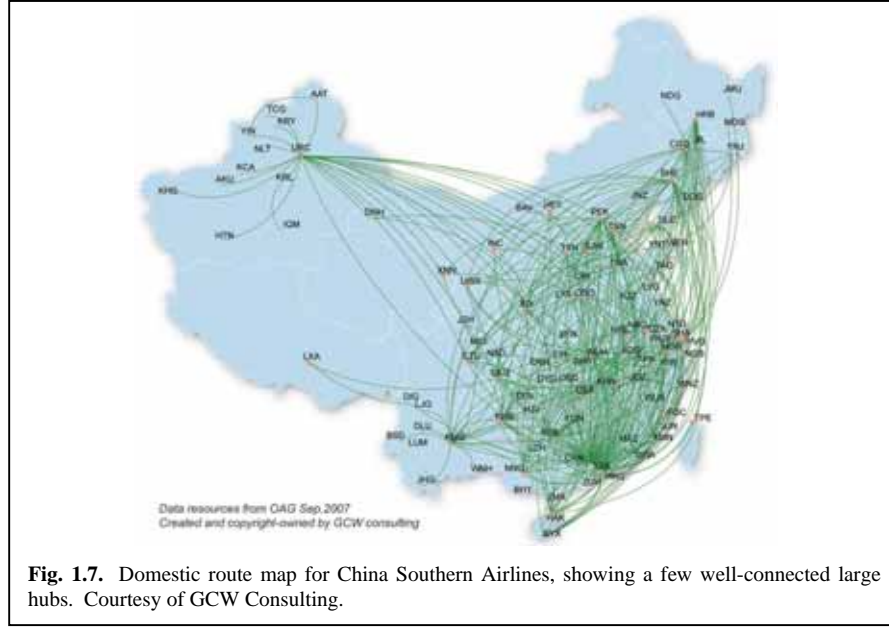
The properties of graphs with a few well-connected nodes were studied as the emergence of scaling in random networks by Barabasi and Albert [27]. They considered two new features that are not present in random graphs or small world networks. First, actual networks in real life applications exhibit growth. Think of the world-wide web and the collaboration network of scientific authors, both of which accrue new nodes as time passes. Second new nodes attach themselves preferentially to existing nodes with large degree, or social influence.

To construct graphs that exhibit growth and preferential attachment, they proceeded as follows. Start with a small number m_0 of nodes. At every time step add a new node that links to $m \leq m_0$ existing nodes in the network. These links are made preferentially to existing nodes with large degrees as follows. Define the degree of node i as k_i. The volume of the graph is defined as the sum of the degrees $Vol = \sum_i k_i$. Then, the m new links are connected to node i with probability $\Pi(i) = k_i / \sum_j k_j$.

A sample network grown by this algorithm is shown in Figure 1.8. It has a few nodes with large degree and many with small degree. A node is said to be large if it has many neighbors. This connection probability growth model results in new nodes connecting preferentially to larger existing nodes, and results in large nodes becoming larger. It explains the airline network and social network just discussed.
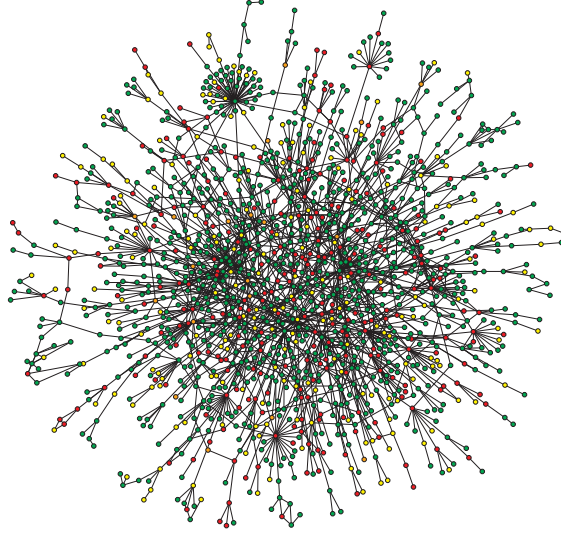
Fig. 1.8. Scale-free network showing a few large well-connected nodes and many small nodes. Courtesy of Prof. Hawoong Jeong and [28]

As the number of time steps becomes large, the network evolves into a graph that has a power law degree distribution $P(k) = k^{-\gamma}$, with γ approximately equal to 3. This is a scale-free state that does not depend on the number of nodes in the graph. A power law distribution is shown in Figure 1.9.

In [27] it is shown that many well-known networks are scale-free with power law distributions $P(k) = k^{-\gamma}$. The collaboration network of movie actors (212,250 nodes) has actors as nodes and an edge if two actors were in the same movie, and is scale-free with $\gamma \approx 2.3$. The world-wide web (800 million nodes) has web pages as nodes and an edge if one page points to another, and has $\gamma \approx 2.1$. The electric power grid in the western United States (4941 nodes) has edges corresponding to transmission lines and has $\gamma \approx 4$.

Fig. 1.9. Power law degree distribution showing *P(k),* the probability that a node has degree *k*, vs. *k*.
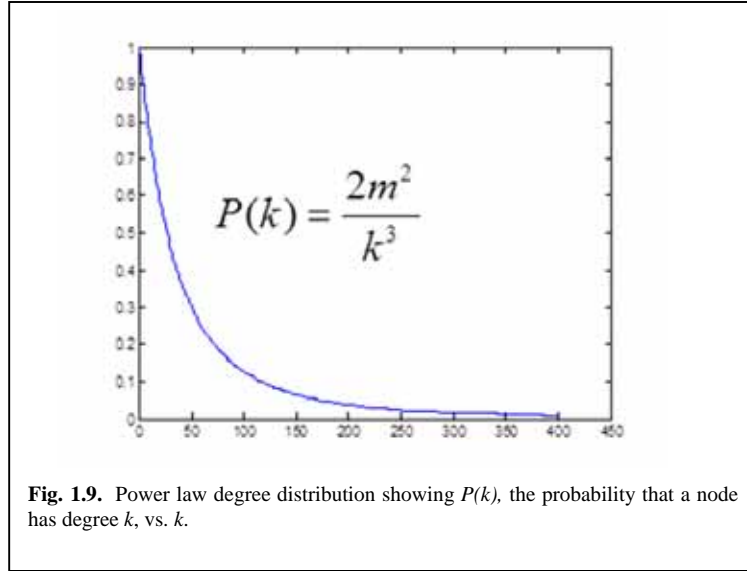
Distance Graphs

In vehicle formations, mobile sensor networks, and the animal synchronization examples in Section 1.1 the connectivity of the graph depends on the maximum communication or sensing range of each individual agent. Given any node, its neighbors are those nodes that fall within its communication range. As the agents move and separation distances change, new edges may form and existing edges may break. The result is a time-varying dynamically changing graph topology. This distance or proximity graph idea also describes the connectivity in flocks of birds and animal herds on the move. A distance graph is shown in Figure 1.10.

Research has shown that a social group on a time-varying distance graph still has cohesiveness and responds as a single collective as long as some conditions on the graph topology are satisfied. Though the graph may not be connected at one particular time, the union of the time-varying graphs should be connected over intervals of time. Many papers have been written about defining potential fields or distributed motion protocols that guarantee the group does not lose connectivity. See for instance [32],[33],[38],[39],[40],[41],[42]. Quite interesting in this context are the potential field methods [35],[36],[37].

Fig. 1.10. Distance or Proximity graph with interaction radius of *d*.
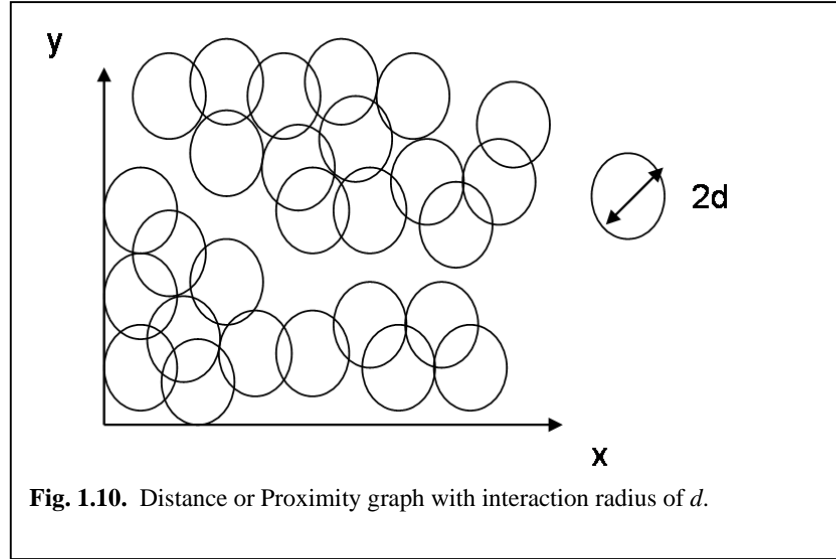
1.3 Cooperative Control of Multi-Agent Systems on Communication Graphs

The main interest of this book is cooperative control of multi-agent dynamical systems interconnected by a communication graph topology. The links of the graph represent the allowed information flow between the systems. In general, directed graphs are used to represent the direction of the allowed flow of information. In cooperative control systems on graphs, there are intriguing interactions between the individual agent dynamics and the topology of the communication graph. The graph topology may severely limit the possible performance of any control laws used by the agents. Moreover, in cooperative control on graphs, all the control protocols must be distributed in the sense that the control law of each agent is only allowed to depend on information from its immediate neighbors in the graph topology. If enough care is not shown in the design of the local agent control laws, the individual agent dynamics may be stable, but the networked systems on the graph may exhibit undesirable behaviors.

In cooperative control, each agent is endowed with its own state variable and dynamics. A fundamental problem in multi-agent dynamical systems on graph is the design of distributed protocols that guarantee consensus or synchronization in the sense that the states of all the agents reach the same value. The states could represent vehicle headings or positions, estimates of sensor readings in a sensor network, oscillation frequencies, trust opinions of each agent, and so on.

The principles of networked dynamical systems that we have discussed in this chapter have been applied to the cooperative control problem. Distributed decision and parallel computation algorithms were studied and developed in [29],[30],[31]. The field of cooperative control was started in earnest with the two seminal papers in 2003 [32] and 2004 [33]. Early work was furthered in [34],[38],[39],[40]. In the development of cooperative control theory, work was generally done initially for simple systems including first-order integrator dynamics in continuous time and discrete time. Then, results were established for second-order systems and higher-order dynamics. Applications were made to vehicle formation control and graphs with time-varying topologies and communication delays. See [41],[42][43][44][45][46].

Early work focused on reaching consensus or synchronization of networks of anonymous dynamical systems, where all agents have the same role. This is known as the leaderless consensus problem or the cooperative regulator problem. Later work focused on synchronization to the dynamics of a leader or root node which generates a command target trajectory. This is the controlled consensus problem or the cooperative tracker. We shall have more to say about the background references and the design techniques for these problems in subsequent chapters.

References

To the Springer Editorial Team.

The references in this chapter need to stay in the order given here. They are grouped into topics and arranged historically by date. Also, please leave the category headings below.

The references in all other chapters should be listed alphabetically.

Thank you

Author F. Lewis

Networks of coupled dynamical systems

Collective Motion in Biological and Social Systems, Physics and Chemistry, and Engineered Systems

[1] C.R. Darwin, *The Origin of Species,* John Murray Publisher, London, 1859

[2] Adam Smith, An Inquiry into the Nature and Causes of the Wealth of Nations, Strahan and Cadell, London, 1776.

[3] Kauffman, S. A., "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.* **22,** 437–467 (1969).

[4] Y. Kuramoto, *Chemical Oscillations, Waves, and Turbulence,* Springer, Berlin, 1984.

[5] Axelrod, R. *The Evolution of Cooperation,* Basic Books, New York, 1984.

[6] C.W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model," Proc. SIGGRAPH 87, Computer Graphics 21(4), pages 25-34. edited by Maureen C. Stone, 1987.

[7] Sattenspiel, L. & Simon, C. P. The spread and persistence of infectious diseases in structured populations. *Math. Biosci.* **90,** 341–366 (1988).

[8] Gray, C. M., Ko¨nig, P., Engel, A. K. & Singer,W. Oscillatory responses in cat visual cortex exhibit intercolumnar synchronization which reflects global stimulus properties. *Nature* **338,** 334–337 (1989).

[9] Strogatz, S. H. & Stewart, I. "Coupled oscillators and biological synchronization,"*Sci. Am.* **269**(6), 102–109 (1993).

[10] Wasserman, S. & Faust, K. *Social Network Analysis: Methods and Applications* (Cambridge Univ. Press, 1994).

[11] Das, R., Mitchell, M. & Crutchfield, J. P. in *Parallel Problem Solving from Nature* (eds Davido, Y., Schwefel, H.-P. & Ma¨nner, R.) 344–353 (Lecture Notes in Computer Science 866, Springer, Berlin, 1994).

[12] T. Vicsek, A. Czir´ok, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, no. 6, pp. 1226–1229, 1995.

[13] Wiesenfeld, K. New results on frequency-locking dynamics of disordered Josephson arrays. *Physica B,* **222,** 315–319 (1996).

[14] Bressloff, P. C., Coombes, S. & De Souza, B. Dynamics of a ring of pulse-coupled oscillators: a group theoretic approach. *Phys. Rev. Lett.* **79,** 2791–2794 (1997).

[15] D. Helbing, I. Farkas, and T. Vicsek, Simulating Dynamical Features of Escape Panic, Nature, Vol. 407, pp. 487– 490, 2000.

[16] I.D. Couzin, J. Krause, N.R. Franks, S.A. Levin, "Effective leadership and decision-making in animal groups on the move," Nature, vol. 533, pp. 513-516, Feb. 2005.

[17] W. Wang and J. Slotine, "A theoretical study of different leader roles in networks*," IEEE Trans. Autom. Control*, vol. 51, no. 7, pp. 1156–1161, Jul. 2006.

[18] M. Defoort, T. Floquet, A. Kokosy, and W. Perruquetti, "Sliding-mode formation control for cooperative autonomous mobile robots," *IEEE Trans. Ind. Electron.*, vol. 55, no. 11, pp. 3944–3953, Nov. 2008.

[19] D. Cruz, J. McClintock, B. Perteet, O. Orqueda, Y. Cao, and R. Fierro, "Decentralized cooperative control - a multivehicle platform for research in networked embedded systems," *IEEE Control Syst. Mag.*, vol. 27, no. 3, pp. 58–78, Jun. 2007.

[20] X. Wang, V. Yadav, and S. Balakrishnan, "Cooperative UAV formation flying with obstacle/collision avoidance," *IEEE Trans. Contr. Syst. Technol.*, vol. 15, no. 4, pp. 672–679, Jul. 2007.

[21] J. Chen, X. Cao, P. Cheng, Y. Xiao, and Y. Sun, "Distributed collaborative control for industrial automation with wireless sensor and actuator networks," *IEEE Trans. Ind. Electron.*, vol. 57, no. 12, pp. 4219–4230, Dec. 2010.

[22] R. Murray, "Recent research in cooperative control of multivehicle systems*," J. Dyn. Syst., Meas., Control*, vol. 129, no. 5, pp. 571–583, Sep. 2007.

Types of Graph Topologies and Structured Information Flow in Collective Groups

[23] S.H. Strogatz, "Exploring complex networks," Nature, vol. 410, pp. 268-276, March 2001.

[24] P.Erdos and A. Renyi, "On the evolution of random graphs," Publ. Math. Inst. Hung. Acad. Sci. vol. **5**, pp. 17-61, 1960.

[25] B. Bollobas, Random Graphs (Academic Press, London, 1985).

[26] D.J. Watts and S.H. Strogatz, "Collective dynamics of small world networks," Nature, vol. 393, pp. 440-442, June 1998.

[27] A.L. Barabasi and R. Albert, "Emergence of scaling in random networks," Science, vol. 286, pp. 509-512, Oct. 1999.

[28] H. Jeong, S.P. Mason, A.-L. Barabasi, and Z.N. Oltvai, "Lethality and centrality in protein networks", Nature, vol. 411, pp.41-42, 2001.

Distributed Computation and Cooperative Control of Multi-Agent Systems on Graphs

[29] J. Tsitsiklis, "Problems in Decentralized Decision Making and Computation," Ph.D. dissertation, Dept. Elect. Eng. and Comput. Sci., MIT, Cambridge, MA, 1984.

[30] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control,* vol. 31, no. 9, pp. 803–812, Sep. 1986.

[31] D. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Belmont, MA, 1997.

[32] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules*," IEEE Trans. Autom. Control,* vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[33] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[34] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations*," IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.

[35] N.E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials, and coordinated control of groups," Proc. IEEE Conf. Decision and Control, pp. 2968-2973, Dec. 2001.

[36] Gazi V., Passino K.M., "Stability analysis of swarms," IEEE Trans. Automatic Control, **48**(4), 692-697 (2003).

[37] Gazi V., Passino K.M., "Stability Analysis of Social Foraging Swarms," IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 34, No. 1. Feb. pp. 539-557, 2004.

[38] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[39] L. Moreau, "Stability of multiagent systems with time-dependent communication links*," IEEE Trans. Autom. Control*, vol. 50, no. 2, pp. 169– 182, Feb. 2005.

[40] W. Ren, R. Beard, and E. Atkins,, "A survey of consensus problems in multi-agent coordination," in *Proc. Amer. Control Conf.*, Portland, OR, pp. 1859–1864. , 2005.

[41] J. Han, M. Li, and L. Guo, "Soft control on collective behavior of a group of autonomous agents by a shill agent*," J. System Science and Complexity,* vol. 19, pp.54-62, 2006.

[42] W. Ren, R. Beard, and E. Atkins, "Information consensus in multivehicle co-operative control," *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.

[43] W. Ren, K. Moore, and Y. Chen, "High-order and model reference consensus algorithms in cooperative control of multivehicle systems," *J. Dyn. Syst., Meas., Control*, vol. 129, no. 5, pp. 678–688, 2007.

[44] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in net-worked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[45] W. Zhu and D. Cheng, "Leader-following consensus of second-order agents with multiple time-varying delays," *Automatica*, vol. 46, no. 12, pp. 1994–1999, 2010.

[46] W. Ren and C. Cao, *Distributed Coordination of Multi-agent Networks: Emergent Problems*, Models, and Issues . London: Springer-Verlag, 2011.

2
Algebraic Graph Theory and Cooperative Control Consensus

Cooperative control studies the dynamics of multi-agent dynamical systems linked to each other by a communication graph. The graph represents the allowed information flow between the agents. The objective of cooperative control is to devise control protocols for the individual agents that guarantee synchronized behavior of the states of all the agents in some prescribed sense. In cooperative systems, any control protocol must be distributed in the sense that it respects the prescribed graph topology. That is, the control protocol for each agent is allowed to depend only on information about that agent and its neighbors in the graph. The communication restrictions imposed by graph topologies can severely limit what can be accomplished by local distributed control protocols at each agent. In fact, the graph topological properties complicate the design of synchronization controllers and result in intriguing behaviors of multi-agent systems on graphs that do not occur in single-agent, centralized, or decentralized feedback control systems.

Of fundamental concern for cooperative systems on graphs is the study of their collective behaviors under the influence of the information flow allowed in the graph. This chapter introduces cooperative synchronization control of multi-agent dynamical systems interconnected by a fixed communication graph topology. Each agent or node is mathematically modeled by a dynamical linear time-invariant (LTI) system. First, in Section 2.1 we give a review of graph basics and algebraic graph theory, which studies certain matrices associated with the graph. The graph Laplacian matrix is introduced and its eigenstructure is studied, including the first eigenvalue, the first left eigenvector, and the second eigenvalue or Fiedler eigenvalue.

In Section 2.2 dynamical systems on graphs are introduced. The idea of distributed control and the consensus problem are introduced. Section 2.3 studies

consensus for first-order integrator dynamics for continuous-time systems. The basic results and key ideas of cooperative control emerge from this study. The importance of the graph eigenstructure in interconnected agent dynamical behavior is shown. Section 2.4 reveals the existence of certain motion invariants for consensus on graphs. Section 2.5 studies consensus for first-order discrete-time systems.

Section 2.6 introduces consensus for general linear dynamical systems on graphs. A fundamental result is given that relates the synchronization of multi-agent systems to the stability of a set of systems that depend on the graph topology properties. This reveals the relationships between local agent feedback design and the global synchronization properties based on the graph communication restrictions. In Section 2.7 is studied consensus for second order position-velocity systems which include motion control in formations.

In Section 2.8 we present some key concepts needed in this book for the design of optimal and adaptive distributed controllers on graphs. That section introduces several important matrix analysis methods for systems in graphs, including irreducible matrices, Frobenius form, stochastic matrices, and M-Matrices. Relationships with the graph topology and eigenstructure are given.

In Section 2.9 we introduce Lyapunov functions for the analysis of the stability properties of cooperative multi-agent control systems. It is seen that Lyapunov functions for studying the stability properties of cooperative control protocols depend on the graph topology. Therefore, a given cooperative control protocol may be stable on one graph topology but not on another. This reveals the close interactions between the performance of locally design control protocols and the manner in which the agents are allowed to communicate.

This chapter follows the development of cooperative control results in the early literature since the first papers [22],[7],[10],[9],[16]. To understand the relationships between the communication graph topology and the local design of distributed feedback control protocols, it was natural to study first-order systems and then second-order systems. Early applications were made to formation control [18],[14],[16],[13]. These studies brought an understanding of the limitations and caveats imposed by the graph communication restrictions and opened the way for many well-known results from systems theory to be extended to the case of multi-agent systems on graphs.

The relations discovered between the communication graph topology and the design of distributed control protocols have resulted in new design techniques for cooperative feedback control. New intriguing interactions have been discovered between graph topology and local control protocol design that reveal the richness of the study of cooperative multi-agent control on graphs, where new phenomena are seen that do not occur in control of single-agent systems. In the remainder of the book we explore these relationships for optimal design and adaptive control on graphs.

2.1. Algebraic Graph Theory

In this book we are concerned with the behaviors and interactions of dynamical systems that are interconnected by the links of a communication network. This communication network is modeled as a graph with directed edges corresponding to the allowed flow of information between the systems. The systems are modeled
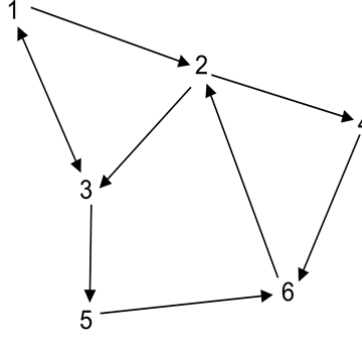
Fig. 2.1 A directed graph.

as the nodes in the graph and are sometimes called agents. We call this the study of multi-agent dynamical systems on graphs. The fundamental control issues concern how the graph topology interacts with the local feedback control protocols of the agents to produce overall behaviors of the interconnected nodes.

2.1.1 Graph Theory Basics

Here we present some basic graph theory concepts that are essential in the study of multi-agent dynamical systems. Good references are [1],[5].

Basic Definitions and Connectivity.

A *graph* is a pair $G = (V, E)$ with $V = \{v_1, \cdots, v_N\}$ a set of N nodes or vertices and E a set of edges or arcs. Elements of E are denoted as (v_i, v_j) which is termed an edge or arc from v_i to v_j, and represented as an arrow with tail at v_i and head at v_j. We assume the graph is simple, i.e. $(v_i, v_i) \notin E, \forall i$ no self loops, and no multiple edges between the same pairs of nodes. Edge (v_i, v_j) is said to be outgoing with respect to node v_i and incoming with respect to v_j; node v_i is termed the parent and v_j the child. The in-degree of v_i is the number of edges having v_i as a head. The out-degree of a node v_i is the number of edges having v_i as a tail. The

set of (in-) neighbors of a node v_i is $N_i = \{v_j : (v_j, v_i) \in E\}$, i.e. the set of nodes with edges incoming to v_i. The number of neighbors $|N_i|$ of node v_i is equal to its in-degree.

If the in-degree equals the out-degree for all nodes $v_i \in V$ the graph is said to be *balanced*. If $(v_i, v_j) \in E \Rightarrow (v_j, v_i) \in E, \forall i, j$ the graph is said to be *bi-directional*, otherwise it is termed a directed graph or digraph. Associate with each edge $(v_j, v_i) \in E$ a weight a_{ij}. Note the order of the indices in this definition. We assume in this chapter that the nonzero weights are strictly positive. A graph is said to be *undirected* if $a_{ij} = a_{ji}, \forall i, j$, that is, if it is bi-directional and the weights of edges (v_i, v_j) and (v_j, v_i) are the same.

A directed path is a sequence of nodes $v_0, v_1, \cdots, v_r$ such that $(v_i, v_{i+1}) \in E, i \in \{0, 1, \cdots, r-1\}$. Node v_i is said to be connected to node v_j if there is a directed path from v_i to v_j. The distance from v_i to v_j is the length of the shortest path from v_i to v_j. Graph G is said to be strongly connected if v_i, v_j are connected for all distinct nodes $v_i, v_j \in V$. For bidirectional and undirected graphs, if there is a directed path from v_i to v_j, then there is a directed path from v_j to v_i, and the qualifier 'strongly' is omitted.

A (directed) tree is a connected digraph where every node except one, called the root, has in-degree equal to one. A spanning tree of a digraph is a directed tree formed by graph edges that connects all the nodes of the graph. A graph is said to have a spanning tree if a subset of the edges forms a directed tree. This is equivalent to saying that all nodes in the graph are reachable from a single (root) node by following the edge arrows. A graph may have multiple spanning trees. Define the root set or leader set of a graph as the set of nodes that are the roots of all spanning trees. If a graph is strongly connected it contains at least one spanning tree. In fact, if a graph is strongly connected then all nodes are root nodes.
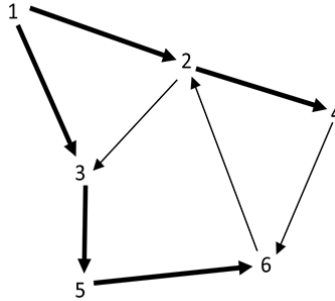
Fig. 2.2 A spanning tree for the graph in Fig. 2.1 with root node 1.

2.1.2 Graph Matrices

Graph structure and properties can be studied by examining the properties of certain matrices associated with the graph. This is known as algebraic graph theory [1],[5].

Given the edge weights a_{ij}, a graph can be represented by an adjacency or connectivity matrix $A = [a_{ij}]$ with weights $a_{ij} > 0$ *if* $(v_j, v_i) \in E$ and $a_{ij} = 0$ otherwise. Note that $a_{ii} = 0$. Define the weighted in-degree of node v_i as the *i*-th row sum of A

$$d_i = \sum_{j=1}^{N} a_{ij} \tag{2.1}$$

and the weighted out-degree of node v_i as the *i*-th column sum of A

$$d_i^o = \sum_{j=1}^{N} a_{ji} \tag{2.2}$$

The in-degree and out-degree are local properties of the graph. Two important global graph properties are the diameter $Diam\,G$, which is the greatest distance between two nodes in a graph, and the (in-)volume, which is the sum of the in-degrees

$$Vol\,G = \sum_i d_i \tag{2.3}$$

The adjacency matrix A of an undirected graph is symmetric, $A = A^T$. A graph is said to be *weight balanced* if the weighted in-degree equals the weighted out-degree for all *i*. If all the nonzero edge weights are all equal to 1, this is the same as the definition of balanced graph. An undirected graph is weight balanced, since if $A = A^T$ then the *i*-th row sum equals the *i*-th column sum. We may be loose at times and refer to node v_i simply as node *i*, and refer simply to in-degree, out-degree, and the balanced property, without the qualifier 'weight', even for graphs having non-unity weights on the edges.

Graph Laplacian Matrix. Define the diagonal in-degree matrix $D = diag\{d_i\}$ and the (weighted) graph Laplacian matrix $L = D - A$. Note that L has all row sums equal to zero. Many properties of a graph may be studied in terms of its

graph Laplacian. In fact, we shall see that the Laplacian matrix is of extreme importance in the study of dynamical multi-agent systems on graphs.

Example 2.1. Graph Matrices

Consider the digraph shown in Figure 2.1 with all edge weights equal to 1. The graph is strongly connected since there is a path between any two pairs of nodes. A spanning tree with root node 1 is shown in bold in Figure 2.2. There are other spanning trees in this graph. In fact, every node is a root node since the graph is strongly connected.

   The adjacency matrix is given by

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

and the diagonal in-degree matrix and Laplacian are

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 & -1 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Note that the row sums of L are all zero.

∎

2.1.3 Eigenstructure of Graph Laplacian Matrix

We shall see that the eigenstructure of the graph Laplacian matrix L plays a key role in the analysis of dynamical systems on graphs. Define the Jordan normal form [2] of the graph Laplacian matrix by

$$L = MJM^{-1} \tag{2.4}$$

with the Jordan for m matrix and transformation matrix given as

$$J = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix} \quad , \quad M = \begin{bmatrix} v_1 & v_2 & \cdots & v_N \end{bmatrix} \qquad (2.5)$$

where the eigenvalues λ_i and right eigenvectors v_i satisfy

$$(\lambda_i I - L)v_i = 0 \qquad (2.6)$$

with I the identity matrix.

In general, the λ_i in (2.5) are not scalars but are Jordan blocks of the form

$$\begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}$$

The number of such Jordan blocks associated to the same eigenvalue λ_i is known as the geometric multiplicity of eigenvalue λ_i. The sum of the sizes of all Jordan blocks associated to λ_i is called its algebraic multiplicity. For ease of notation and discussion we assume here that the Jordan form is simple, that is it is diagonal with all Jordan blocks of size 1. This is guaranteed if all eigenvalues of L are distinct. A symmetric matrix may not have distinct eigenvalues but has a simple Jordan form. All of our discussions generalize without difficulty to the case of nontrivial Jordan blocks.

The inverse of the transformation matrix M is given as

$$M^{-1} = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_N^T \end{bmatrix} \qquad (2.7)$$

where the left eigenvectors w_i satisfy

$$w_i^T (\lambda_i I - L) = 0 \tag{2.8}$$

and are normalized so that $w_i^T v_i = 1$.

We assume the eigenvalues are ordered so that $|\lambda_1| \leq |\lambda_2| \leq \cdots \leq |\lambda_N|$. Any undirected graph has $L = L^T$ so all its eigenvalues are real and one can order them as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$.

Since L has all row sums zero, one has

$$L\underline{1}c = 0 \tag{2.9}$$

with $\underline{1} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^T \in R^N$ the vector of 1's and c any constant. Therefore, $\lambda_1 = 0$ is an eigenvalue with a right eigenvector of $\underline{1}c$. That is, $\underline{1}c \in N(L)$ the nullspace of L. If the dimension of the nullspace of L is equal to one, i.e. the rank of L is *N-1*, then $\lambda_1 = 0$ is nonrepeated and $\underline{1}c$ is the only vector in *N(L)*. The next standard result states when this occurs.

Theorem 2.1. [16],[13]. *L has rank N-1, i.e.* $\lambda_1 = 0$ *is nonrepeated, if and only if graph G has a spanning tree.*

If the graph has a spanning tree then $|\lambda_2| > 0$. If the graph is strongly connected, then it has a spanning tree and L has rank *N-1*.

The Laplacian has at least one eigenvalue at $\lambda_1 = 0$. The remaining eigenvalues can be localized using the following result.

Geršgorin Circle Criterion. [6] *All eigenvalues of a matrix* $E = [e_{ij}] \in R^{N \times N}$ *are located within the union of N discs*

$$\bigcup_{i=1}^{N} \left\{ z \in C : |z - e_{ii}| \leq \sum_{j \neq i} |e_{ij}| \right\}$$

∎
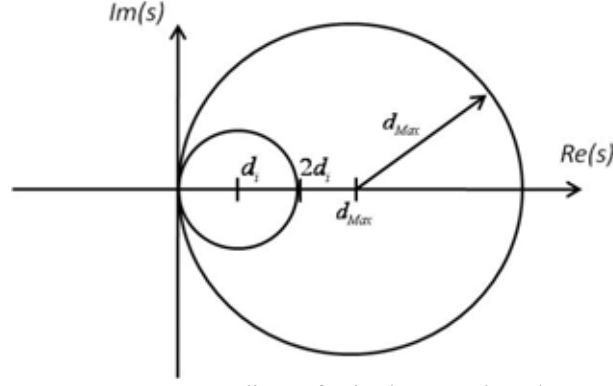
Fig. 2.3. Geršgorin discs of L in the complex plane.

The i-th disc in the Geršgorin Circle Criterion is drawn with a center at the diagonal element e_{ii} and with a radius equal to the i-th absolute row sum with the diagonal element deleted, $\sum_{j\neq i}\left|e_{ij}\right|$. Therefore the Geršgorin discs for the graph Laplacian matrix $L = D - A$ are centered at the in-degrees d_i and have radius equal to d_i. Let d_{Max} be the maximum in-degree of G. Then, the largest Geršgorin disc of the Laplacian matrix L is given by a circle centered at d_{Max} and having radius of d_{Max}. This circle contains all the eigenvalues of L. See Figure 2.3. The Geršgorin Circle Criterion ties the eigenvalues of L rather closely to the graph structural properties in terms of the in-degrees.

We have thus discovered that if the graph has a spanning tree, there is a nonrepeated eigenvalue at $\lambda_1 = 0$ and all other eigenvalues have positive real parts, i.e. in the open right half-plane, and are within a circle centered at d_{Max} and having radius of d_{Max}.

When comparing eigenvalues between two graphs, it is often more useful to use the normalized Laplacian matrix

$$\overline{L} = D^{-1}L = D^{-1}(D - A) = I - D^{-1}A \tag{2.10}$$

Since the normalized adjacency matrix $\overline{A} = D^{-1}A$ has row sums equal to one, $d_i = 1, \forall i$ and $\overline{L}$ has all Geršgorin discs centered at s=1 with radius of 1.

Example 2.2. Laplacian Matrix Eigenvalues for Various Graph Types

In this example we will compute the eigenvalues of the graph Laplacian matrix $L = D - A$ for various types of graphs. The intent is to give a feeling for the dependence of the Laplacian eigenvalues on the graph topology. This example is the work of David Maxwell in the class EE 5329 Distributed Decision & Control, Spring 2011, Dept. of Electrical Engineering, The University of Texas at Arlington. The class link is http://arri.uta.edu/acs

The graphs studied in this example include several commonly occurring topologies and are depicted in Figure 2.4. The graph notation is standard and is explained in [3],[5]. We include it for interest only and do not use this notation further in the book. All edge weights are taken equal to 1. Note that a complete (or fully connected) graph is one that has all possible edges between the nodes.

The eigenvalues of these graphs are listed in Table 2 and plotted in the complex s-plane in Figure 2.5. Several things are worthy of note:

1. The Laplacian matrix has row sums equal to zero so that all graphs have the first eigenvalue at $\lambda_1 = 0$.

2. All undirected graphs have a symmetric Laplacian matrix *L* and so their graph eigenvalues are real.

3. Complete (or fully connected) graphs with *N* nodes have all the nonzero eigenvalues at *s=N*.

4. Directed trees have all the nonzero eigenvalues at *s=1*.

5. A directed star is a directed tree, so it has all the nonzero eigenvalues at *s=1*.

6. Directed cycle graphs of length *N* have *N* eigenvalues uniformly spaced around a circle in the *s*-plane centered at *s=1* with radius of *s=1*, with the first eigenvalue at $\lambda_1 = 0$.
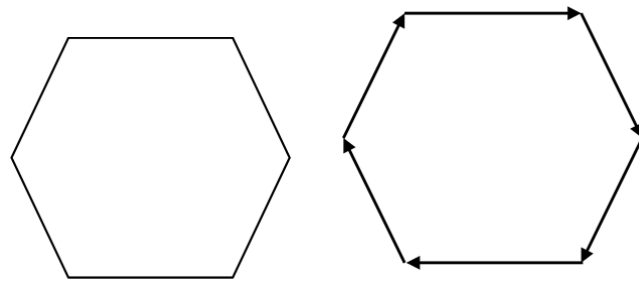
■

a. Directed graph.   b. Undirected graph

c. Complete graph K^6 d. Directed tree graph (formation graph)

e. Undirected star $K_{1,5}$ f. Directed star

Fig. 2.4 Different graph topologies for Example 2.2.

g. Undirected 6-cycle.
    (2-regular graph)

h. Directed 6-cycle
   (6-periodic graph)
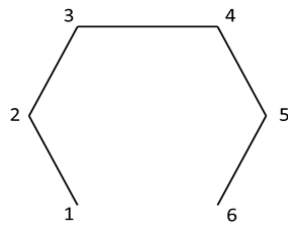
i. Undirected path P^5

Fig. 2.4 Different graph topologies for Example 2.2.

a. directed graph	b. undirected graph	c. complete graph	d. directed tree	e. undirected star	f. directed star	g. undirected 6-cycle	h. directed 6-cycle	i. undirected path
0	0	0	0	0	0	0	0.0000	0
0.7793	1.3820	6	1	1	1	1	0.5 +0.866i	0.2679
1.0000	1.6972	6	1	1	1	1	0.5 -0.866i	1
2.2481 +1.0340i	3.6180	6	1	1	1	3	1.5 +0.866i	2
2.2481 -1.0340i	4.0000	6	1	1	1	3	1.5 -0.866i	3
2.7245	5.3028	6	1	6	1	4	2.	3.7321

Table 2.1 Eigenvalues for different graph topologies in Example 2.2.
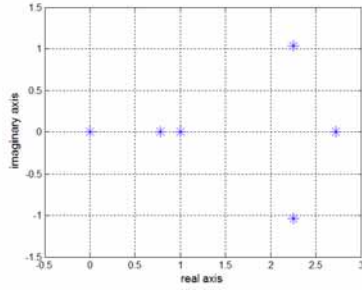
a. Directed graph.
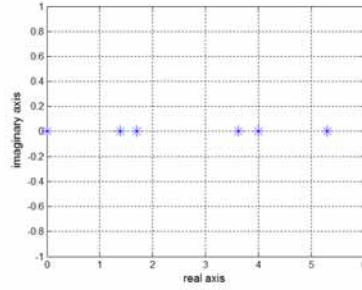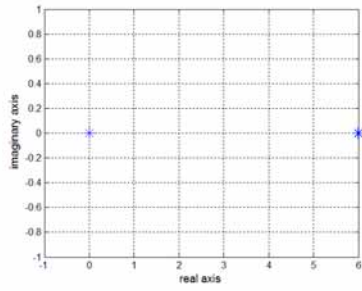
b. Undirected graph.

c. Complete graph K^6.
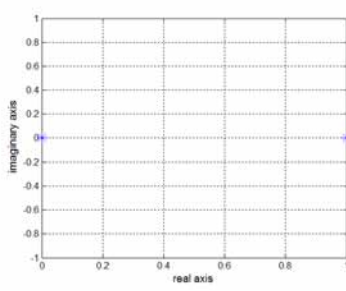
d. Directed tree (formation graph).
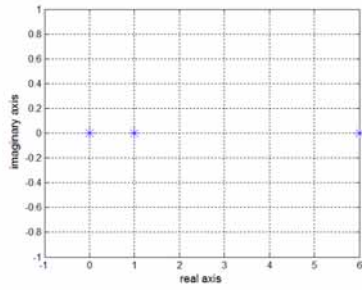
e. Undirected star $K_{1,5}$.
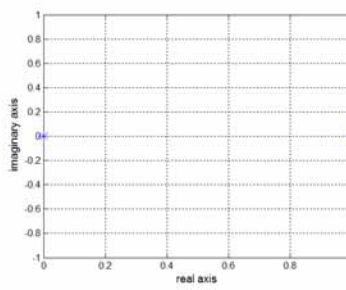
f. Directed star.

Fig. 2.5 Complex plane plots of graph eigenvalues for Example 2.2.
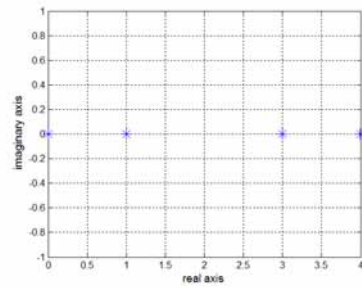
g. Undirected 6-cycle
(2-regular graph).
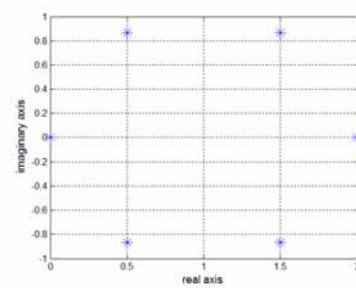
h. Directed 6-cycle
(6-periodic graph).
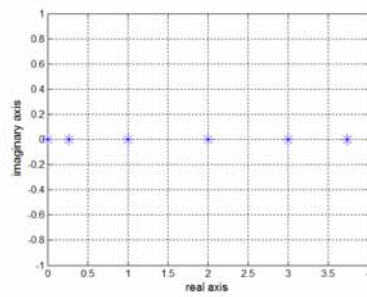
i. Undirected path P^5

Fig. 2.5 Complex plane plots of graph eigenvalues for Example 2.2.

2.1.4 The Fiedler Eigenvalue

We shall see in Theorem 2.2 that the second eigenvalue λ_2 of the graph Laplacian matrix L is important in determining the speed of interaction of dynamic systems on graphs. Graph topologies that have a large value of λ_2 are better for achieving convergence. The second eigenvalue of L is known as the *Fiedler* eigenvalue and it has important and subtle connections to the topology of the graph. It is also known as the graph algebraic connectivity. Fiedler eigenvalues for some different graph topologies are given in Example 2.2.

For undirected graphs there are many useful bounds on the Fiedler eigenvalue, including [23]

$$\lambda_2 \leq \frac{N}{N-1} d_{\min} \qquad (2.11)$$

where $d_{\min}$ is the minimum in-degree, and for connected undirected graphs

$$\lambda_2 \geq \frac{1}{Diam\, G \times Vol\, G} . \qquad (2.12)$$

The distance between two nodes in an undirected graph is the length of the shortest path connecting them, and if they are not connected the distance between them is infinity. The diameter of a graph $Diam\, G$ is the greatest distance between any two nodes in G and the volume $Vol\, G$ is given by (2.3). Bounds on the Fiedler eigenvalue for directed graphs are more complicated [23].

Interactive systems on graph converge to steady-state values faster when λ_2 is larger, as we shall see. Given that, according to (2.12) convergence is faster in graphs that are more fully connected, that is, have shorter distances between any pair of nodes, and for graphs whose nodes have fewer neighbors. Among the fastest graph topologies are the fully connected complete graph and the star graph [23], where one root node is connected to all other nodes. The star is a tree graph with depth equal to one.

2.2. Systems on Communication Graphs and Consensus

A network may be considered as a set of nodes or agents that collaborate to achieve what each cannot achieve alone. To capture the notion of dynamical

agents, endow each node i of a graph with a time-varying state vector $x_i(t)$. A graph with node dynamics [10] is (G, x) with G a graph having N nodes and $x = [x_1^T \cdots x_N^T]^T$ a global state vector, where the state of each node evolves according to some dynamics

$$\dot{x}_i = f_i(x_i, u_i) \tag{2.13}$$

with u_i a control input and $f_i(.)$ some flow function. Given a graph $G = (V, E)$, we interpret $(v_i, v_j) \in E$ to mean that node v_j can obtain information from node v_i for feedback control purposes. [10],[7],[9].

Definition 2.1. Distributed Control Protocols. *The control given by $u_i = k_i(x_{i_1}, x_{i_2}, \cdots, x_{i_{m_i}})$ for some function $k_i(.)$ is said to be distributed if $m_i < N, \forall i$, that is, the control input of each node depends on some proper subset of all the nodes. It is said to be a protocol with topology G if $u_i = k_i(x_i, \{x_j \mid j \in N_i\})$, that is, each node can obtain information about the state only of itself and its (in)-neighbors in N_i.*

Cooperative control, or control of distributed dynamical systems on graphs, refers to the situation where each node can obtain information for controls design only from itself and its neighbors. The graph might represent a communication network topology that restricts the allowed communications between the nodes. This has also been referred to as multi-agent control, but is not the same as the notion of multi-agent systems used by the Computer Science community [20].

Of fundamental concern for systems on graphs is the study of their collective behaviors under the influence of the information flow allowed in the graph. A basic control design objective is the following.

Definition 2.2. Consensus Problem. *Find a distributed control protocol that drives all states to the same constant steady-state values $x_i = x_j, \forall i, j$. This value is known as a consensus value.*

2.3. Consensus with Single-Integrator Dynamics

We start our study of dynamical systems on graphs, or cooperative control, by considering the case where all nodes of the graph G have scalar single-integrator dynamics

$$\dot{x}_i = u_i \tag{2.14}$$

with $x_i, u_i \in R$. This corresponds to endowing each node or agent with a memory.

2.3.1 Distributed Control Protocols for Consensus

Consider the local control protocols for each agent i

$$u_i = \sum_{j \in N_i} a_{ij}(x_j - x_i) \tag{2.15}$$

with a_{ij} the graph edge weights. This control is distributed in that it only depends on the immediate neighbors N_i of node i in the graph topology. This is known as a *local voting protocol* since the control input of each node depends on the difference between its state and all its neighbors. Note that if these states are all the same, then $\dot{x}_i = u_i = 0$. In fact, it will be seen that, under certain conditions, this protocol drives all states to the same value.

We wish to show that protocol (2.15) solves the consensus problem and to determine the consensus value reached. Write the closed-loop dynamics as

$$\dot{x}_i = \sum_{j \in N_i} a_{ij}(x_j - x_i) \tag{2.16}$$

$$\dot{x}_i = -x_i \sum_{j \in N_i} a_{ij} + \sum_{j \in N_i} a_{ij}x_j = -d_i x_i + \begin{bmatrix} a_{i1} & \cdots & a_{iN} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \tag{2.17}$$

with d_i the in-degree. Define the global state vector $x = [x_1 \cdots x_N]^T \in R^N$ and the diagonal matrix of in-degrees $D = diag\{d_i\}$. Then the global dynamics are given by

$$\dot{x} = -Dx + Ax = -(D - A)x \qquad (2.18)$$

$$\dot{x} = -Lx \qquad (2.19)$$

Note that the global control input vector $u = [u_1 \cdots u_N]^T \in R^N$ is given by

$$u = -Lx \qquad (2.20)$$

It is seen that, using the local voting protocol (2.15), the closed-loop dynamics (2.19) depends on the graph Laplacian matrix L. We shall now see how the evolution of first-order integrator dynamical systems on graphs depends on the graph properties through the Laplacian matrix. The eigenvalues of L are instrumental in this analysis.

The dynamics given by (2.19) has a system matrix of $-L$, and hence has eigenvalues in the left-half plane, specifically inside the circle in Figure 2.3 reflected into the left half plane. At steady-state, according to (2.19) one has

$$0 = -Lx_{ss} \qquad (2.21)$$

Therefore, the steady-state global state is in the nullspace of L. According to (2.9) one vector in $N(L)$ is $\underline{1}c$ for any constant c. If L has rank of N-1, then $\underline{1}c$ is the only vector in nullspace of L and one has $x_{ss} = \underline{1}c$ for some constant c. Then one has at steady-state

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = c\underline{1} = \begin{bmatrix} c \\ \vdots \\ c \end{bmatrix} \qquad (2.22)$$

and $x_i = x_j = c = const, \ \forall i, j$. Then, consensus is reached. The next result formalizes this discussion and delivers the consensus value c.

Theorem 2.2. Consensus for First-order Systems. *The local voting protocol (2.15) guarantees consensus of the single-integrator dynamics (2.14) if and only if the graph has a spanning tree. Then, all node states come to the same steady-state values $x_i = x_j = c, \ \forall i, j$. The consensus value is given by*

$$c = \sum_{i=1}^{N} p_i x_i(0) \qquad (2.23)$$

where $w_1 = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix}^T$ *is the normalized left eigenvector of the Laplacian L for* $\lambda_1 = 0$. *Finally, consensus is reached with a time constant given by*

$$\tau = 1/\lambda_2 \qquad\qquad (2.24)$$

with λ_2 *the second eigenvalue of L, that is, the Fiedler eigenvalue.*

Proof. The proof is given for case that *L* is simple, i.e. has all Jordan blocks of order one. The general case follows similarly. Using modal decomposition [2] write the solution of (2.19) in terms of the Jordan form of *L* as

$$x(t) = e^{-Lt}x(0) = Me^{-Jt}M^{-1}x(0) = \sum_{j=1}^{N} v_i e^{-\lambda_i t} w_i^T x(0) = \sum_{j=1}^{N} \left(w_i^T x(0) \right) e^{-\lambda_i t} v_i$$

$$(2.25)$$

Here, the left and right eigenvectors are normalized so that $w_i^T v_i = 1$. If the graph has a spanning tree, then Theorem 2.1 shows that $\lambda_1 = 0$ is simple and the Geršgorin Circle Criterion shows that all other eigenvalues of *L* are in the open right half of the complex plane. Then system (2.19) is marginally stable with one pole at the origin and the rest in the open left half plane. Therefore, in the limit as $t \to \infty$ one has

$$x(t) \to v_2 e^{-\lambda_2 t} w_2^T x(0) + v_1 e^{-\lambda_1 t} w_1^T x(0) \qquad\qquad (2.26)$$

with Fiedler eigenvalue λ_2 the smallest magnitude nonzero eigenvalue. One has $\lambda_1 = 0$, and take $v_1 = \underline{1}$. Define the left eigenvector $w_1 = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix}^T$, normalized so that $w_i^T v_i = 1$, that is, $\sum_i p_i = 1$. Then

$$x(t) \to v_2 e^{-\lambda_2 t} w_2^T x(0) + \underline{1} \sum_{i=1}^{N} p_i x_i(0) \qquad\qquad (2.27)$$

The last term in this equation is the steady-state value $x_{ss} = c\underline{1}$, with *c* given in (2.23). The first term on the right verifies that the consensus value is reached with a time constant given by $\tau = 1/\lambda_2$.

∎

If the graph is strongly connected, then it has a spanning tree and consensus is reached.

If the graph does not have a spanning tree, then the dimension of nullspace of L is greater than one and there is a ramp term in (2.25) that increases with time. Then, consensus is not reached.

If the graph has a spanning tree, there is a simple pole at zero and the system is of Type I. Then, it reaches a steady-state value. Otherwise, the system is of Type 2 or higher and a constant steady-state value is not reached. This is in line with the previous paragraph. Thus, we have the curious situation that the system dynamical behavior, notably the system type, depends on the graph topology, that is, the manner in which the nodes communicate.

Theorem 2.2 shows that the consensus value is in the convex hull of the initial node states, and is the normalized linear combination of the initial states weighted by the elements p_i of the left eigenvector w_1 for $\lambda_1 = 0$.

Due to their importance in the analysis of dynamics and consensus on graphs, we call $\underline{1}$ the first right eigenvector of L and w_1 its first left eigenvector.

It is seen from (2.24) that the speed of convergence to consensus depends on the Fiedler eigenvalue λ_2. Convergence is faster for graphs with larger λ_2. The Fiedler eigenvalue was displayed for various graph topologies in Example 2.2.

This development has made clear the importance for networked dynamical systems of the communication graph topology as given in terms of the eigenstructure of the Laplacian matrix L, including its first eigenvalue $\lambda_1 = 0$, its right eigenvector $v_1 = \underline{1}$ and left eigenvector w_1, and the second (Fiedler) eigenvalue λ_2. In some cases, especially in undirected graphs, these quantities can be more closely tied to topological properties of the graph.

Example 2.3. Single-Integrator Consensus for Different Graph Topologies

In this example we plot the system states in consensus dynamics (2.19) for the graph topologies in Example 2.2, that is in Figure 2.4. This example is the work of Ernest Strinden in the class EE 5329 Distributed Decision & Control, Spring 2011, Dept. of Electrical Engineering, The University of Texas at Arlington. The class link is http://arri.uta.edu/acs

Note first that the Fiedler eigenvalues λ_2 are shown in Table 2.1. According to Theorem 2.2, these give a first-order estimate of the time to reach consensus, since from (2.24), consensus is reached faster for graph topologies having larger λ_2.

The consensus plots in Figure 2.6 show the states x_i in the consensus dynamics $\dot{x} = -Lx$ of (2.19) for the graph topologies in Example 2.2. Several things are worthy of note:

1. Speed of convergence is generally faster for larger Fiedler eigenvalues in Table 2.1.
2. Undirected graphs have real eigenvalues so there is no oscillation on the way to consensus.
3. Complete graph has fastest consensus since information flows immediately between any pair of nodes.
4. Undirected graph *b.* has faster consensus than directed graph *a.*, since more information passes between the nodes.
5. Directed 6-cycle graph has slow convergence and oscillations since it has complex eigenvalues uniformly spaced around a circle in the *s*-plane.
6. Undirected path has very slow convergence since it has a large diameter.
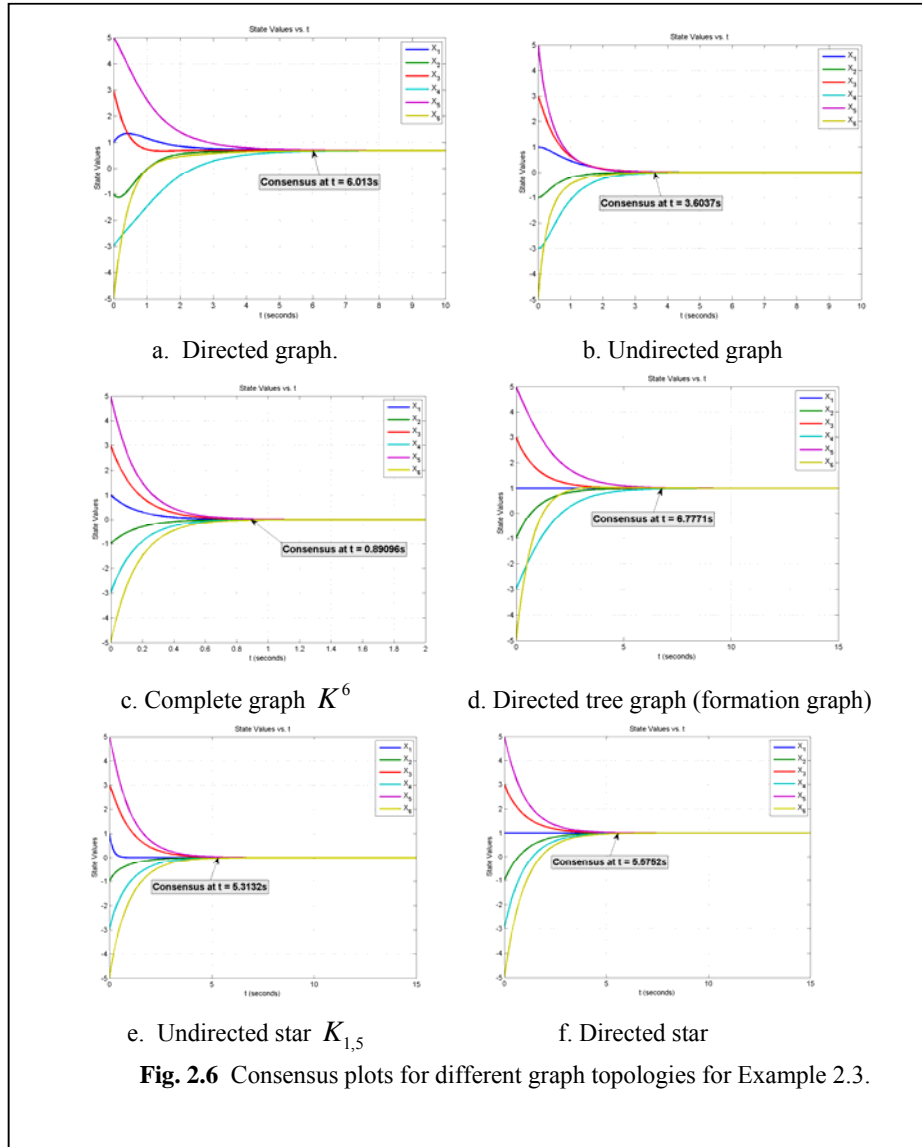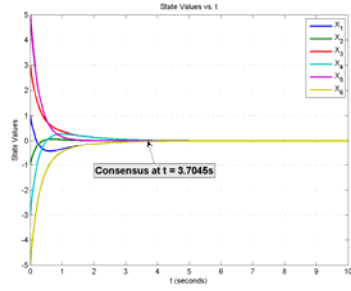
∎

a. Directed graph.

b. Undirected graph

c. Complete graph K^6

d. Directed tree graph (formation graph)

e. Undirected star $K_{1,5}$

f. Directed star

Fig. 2.6 Consensus plots for different graph topologies for Example 2.3.

g. Undirected 6-cycle.
(2-regular graph)
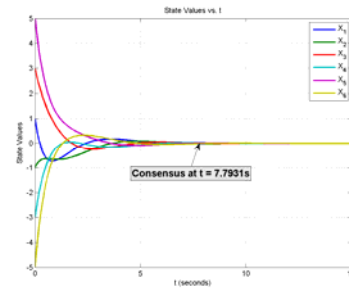
h. Directed 6-cycle
(6-periodic graph)
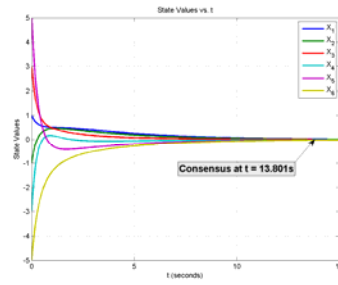
i. Undirected path P^5

Fig. 2.6 Consensus plots for different graph topologies for Example 2.3..

Example 2.4. Consensus of Headings in Swarm Motion Control

The local voting protocol (2.15) can be used in many applications to yield consensus. An example is the consensus of headings in a formation or in animal groups. Simple motions of a group of N agents in the (x, y) plane as shown in Figure 2.7 can be described by the node dynamics

$$\dot{x}_i = V \cos \theta_i$$
$$\dot{y}_i = V \sin \theta_i$$

(2.28)

where V is the speed of the agents, assumed here to be the same, and θ_i is the heading of agent i.
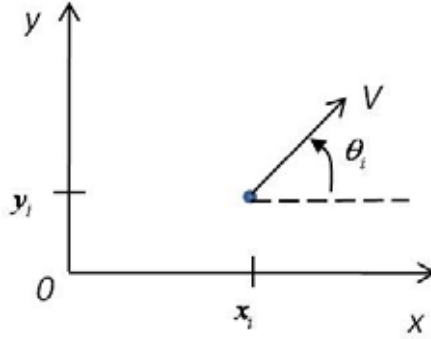
Fig. 2.7. Motion of nodes in (x, y)-plane

According to the observed behavior of animal social groups such as flocks, fish schools, etc, agents moving in groups tend to align their headings to a common value. This is formalized in Reynolds three rules of animal behavior in groups [19]. Therefore, use the local voting protocol (2.15) to reach heading consensus according to

$$\dot{\theta}_i = \sum_{j \in N_i} a_{ij} (\theta_j - \theta_i)$$

(2.29)
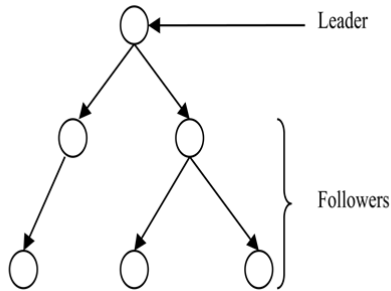
Fig. 2.8. Tree graph for motion control.

Consider the tree graph shown in Figure 2.8 with 6 agents. A simulation is run of the dynamics (2.29), (2.28). Figure 2.9 shows the headings of the agents, starting from random values. All headings reach the same consensus value. Figure 2.10 shows the motion of the agents in the (x, y) plane with different randomly selected initial positions. All agents converge to the same heading and move off in a formation together. Note that all nodes converge to the heading of the leader.

In this example, if the node speeds are not all the same, one could run two consensus algorithms at each node, one for the headings θ_i and one for the speeds V_i

$$\dot{V}_i = \sum_{j \in N_i} a_{ij}(V_j - V_i) \tag{2.30}$$
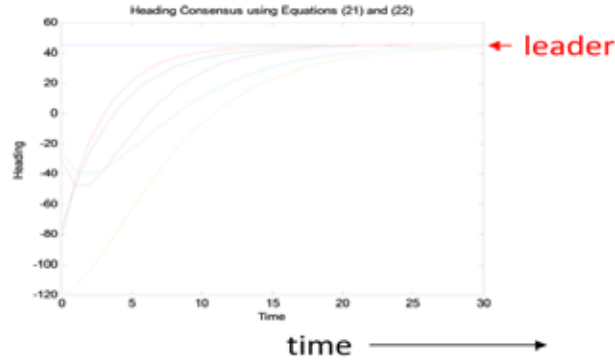
Fig. 2.9 Consensus of headings.
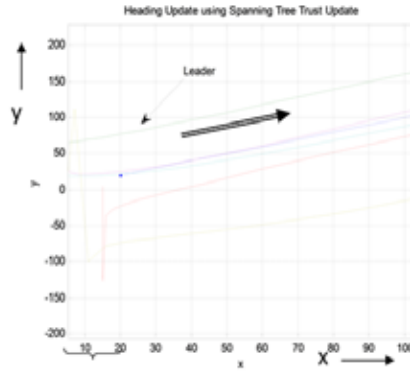
Fig. 2.10 Motion consensus in *(x,y)* plane

2.3.2 Consensus Value for Balanced Graphs- Average Consensus

The consensus value (2.23) is the weighted average of the initial conditions of the states of the nodes. It depends on the graph topology through the left eigenvector $w_1 = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix}^T$ of the zero eigenvalue of Laplacian matrix L. It is not always desirable for the consensus value to depend on the graph topology, i.e. on the manner in which the nodes communicate. An important problem in cooperative control is therefore the following [7],[10].

Average Consensus Problem. *Find a control protocol that drives all states to the same constant steady-state values* $x_i = x_j = c, \forall i, j$, *where c is the average of the initial states of the nodes*

$$c = \frac{1}{N} \sum_i x_i(0) \tag{2.31}$$

This value does not depend on the graph structure. The average consensus problem is important is many applications. For instance, in wireless sensor networks each node measures some quantity (e.g. temperature, salinity content, etc.) and it is desired to determine the best estimate of the measured quantity, which is the average if all sensors have identical noise characteristics.

A graph is said to be weight balanced if the weighted in-degree equals the weighted out-degree for all nodes $v \in V$. We shall omit the qualifier 'weight' for simplicity. The row sums of the Laplacian L are all zero. For balanced graphs, the *i*-th row sum and *i*-th column sum of adjacency matrix A, and hence L, are equal. Then, the column sums of the Laplacian matrix are also all equal to zero and

$$w_1^T L = c \underline{1}^T L = 0 \tag{2.32}$$

so that a left eigenvector for $\lambda_1 = 0$ is $w_1 = c\underline{1}$. This is normalized by dividing each element by N. Then the consensus value (2.23) is (2.31) and average consensus is reached.

Note that if a graph is undirected, it is balanced, so that average consensus is reached.

2.3.3 Consensus Leaders

A (directed) tree is a connected digraph where every node except one, called the root or leader, has in-degree equal to one. It was seen in Example 2.4 that all

nodes reached a consensus heading equal to the initial heading of the leader. The consensus value is given by (2.23) with p_i the i-th component of the left eigenvector w_1 for the zero eigenvalue. If the graph is strongly connected, one has $p_i > 0, \forall i$ since each node influences all other nodes along directed paths. If the graph has a spanning tree but is not strongly connected, then some nodes do not have paths to all other nodes. Thus, all other nodes cannot be aware of the initial states of such nodes.

Suppose a graph contains a spanning tree. Then it may contain more than one. Define the root set or leader set of a graph as the set of nodes that are the roots of all spanning trees.

Theorem 2.3. Consensus Leaders [23]. *Suppose a graph $G = (V, E)$ contains a spanning tree and define the leader set $W \subset V$. Define the left eigenvector of Laplacian matrix L for $\lambda_1 = 0$ as $w_1 = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix}^T$. Then $p_i > 0, i \in W$ and $p_i = 0, i \notin W$.*

This result shows that the consensus value (2.23) is actually the weighted average of the initial conditions of the root or leader nodes in a graph, that is, of the nodes that have a path to all other nodes in the graph. In Example 2.4 the graph is a tree. Therefore, all nodes reach a consensus heading equal to the initial heading of the leader node. In a strongly connected graph, all nodes are leader nodes so that $p_i > 0, \forall i$.

2.3.4 Non-Scalar Node States

It was assumed that the node states and controls in the dynamics (2.14) are scalars $x_i, u_i \in R$. If the states and controls are vectors so that $x_i, u_i \in R^n$, then the global state and control are $x = [x_1^T \cdots x_N^T]^T \in R^{nN}$, $u = [u_1^T \cdots u_N^T]^T \in R^{nN}$ and the elements a_{ij} and d_i in (2.17) are multiplied by the $n \times n$ identity matrix I_n. Then, the global dynamics are written as

$$u = -(L \otimes I_n)x \tag{2.33}$$

$$\dot{x} = -(L \otimes I_n)x \tag{2.34}$$

with $\otimes$ the Kronecker product [1]. All of our developments still hold if the Kronecker product is added where appropriate.

Given two matrices $A = [a_{ij}], B$ the Kronecker product is defined as $A \otimes B = [a_{ij}B]$, the matrix with block elements $a_{ij}B$.

Simulation of Cooperative Systems with Vector States

If the states in (2.14) are vectors so that $x_i, u_i \in R^n$, then $x = [x_1^T \cdots x_N^T]^T \in R^{nN}$, $u = [u_1^T \cdots u_N^T]^T \in R^{nN}$ and the global dynamics are written as (2.34). This can cause numerical integration problems in simulations since the dimension nN of the global state becomes large quickly as the node state dimension n and the number of nodes N increase. The matrix $(L \otimes I_n) \in R^{nN} \times R^{nN}$ is a large sparse matrix with many zero entries. For simulation purposes, it is better to define the $N \times n$ matrix of states

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \\ x_N^T \end{bmatrix} \tag{2.35}$$

Then

$$\dot{X} = -LX \tag{2.36}$$

which does not involve the Kronecker product and has better numerical properties for simulation.

2.4. Motion Invariants for First-Order Consensus

The first-order local protocol (2.15) has a number of important properties. We have seen that it guarantees consensus if the graph has a spanning tree. Here we show that it induces a decomposition based on motion invariants that depend on the graph topology [8].

2.4.1 Graph Motion Invariant and 'Internal Forces'

Let $w_1 = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix}^T$ be a (not necessarily normalized) left eigenvector of L for $\lambda_1 = 0$, then according to (2.19)

$$\frac{d}{dt}(w_1^T x) = w_1^T \dot{x} = -w_1^T L x = 0 \tag{2.37}$$

so that the quantity

$$\bar{x} \equiv w_1^T x = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \sum_i p_i x_i \tag{2.38}$$

is invariant. That is, regardless of the values of the node states $x_i(t)$, the quantity $\bar{x} = \sum_i p_i x_i(t)$ is a constant of the motion. This means that the global velocity vector $\dot{x}(t) \in R^N$ is always orthogonal to the vector $w_1 \in R^N$.

Accordingly, $\sum_i p_i x_i(0) = \sum_i p_i x_i(t)$, $\forall t$. Therefore, if the graph has a spanning tree, at steady-state one has consensus so that $x_i = x_j = c, \forall i, j$ where the consensus value is given by

$$c = \frac{\sum_i p_i x_i(0)}{\sum_i p_i} \tag{2.39}$$

This provides another proof for (2.23) (note that in (2.23), the left eigenvector is normalized).

According to (2.20) the global control input is $u = -Lx$ so that

$$w_1^T u = \sum_i p_i u_i = -w_1^T L x = 0 \tag{2.40}$$

with $u_i(t)$ the node control inputs. This states that the linear combination of control inputs weighted by the elements of the left eigenvector for $\lambda_1 = 0$ is equal to zero. This can be interpreted as a statement that the internal forces in the graph do no work.

2.4.2 Center of Gravity Dynamics and Shape Dynamics

The quantity $\bar{x}$ in (2.38) is a weighted centroid, or center of gravity, of the group of agents, which remains stationary under the local voting protocol. Define the state-space coordinate transformation [8] $z = Mx$ with

$$
M = \begin{bmatrix}
p_1 & p_2 & p_3 & \cdots & p_N \\
1 & -1 & & & \\
& 1 & -1 & & \\
& & & \ddots & \\
& & & 1 & -1
\end{bmatrix}
\tag{2.41}
$$

with $w_1 = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix}^T$ the normalized first left eigenvector of L, that is, $w_1^T L = 0$, $\sum_i p_i = 1$. Then

$$
z = \begin{bmatrix}
\bar{x} \\
x_1 - x_2 \\
\vdots \\
x_{N-1} - x_N
\end{bmatrix} \equiv \begin{bmatrix} \bar{x} \\ \tilde{x} \end{bmatrix}
\tag{2.42}
$$

with $\tilde{x}(t) \in R^{N-1}$ an error vector that shows how far the node states are from consensus. It can be seen that

$$M^{-1} = \begin{bmatrix} 1 & P_2 & P_3 & \cdots & P_N \\ 1 & P_2 - 1 & P_3 & & P_N \\ 1 & P_2 - 1 & P_3 - 1 & & \vdots \\ \vdots & \vdots & \vdots & \ddots & P_N \\ 1 & P_2 - 1 & P_3 - 1 & & P_N - 1 \end{bmatrix} \tag{2.43}$$

where $P_k = \sum_{i=k}^{N} p_i$.

Now, one has

$$\dot{z} = -MLM^{-1}z \tag{2.44}$$

where

$$MLM^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & \bar{L} \end{bmatrix} \tag{2.45}$$

(Note that the first row of M is the first left eigenvector of L and the first column of M^{-1} the first right eigenvector of L.) Therefore,

$$\begin{aligned} \dot{\bar{x}} &= 0 \\ \dot{\tilde{x}} &= -\bar{L}\tilde{x} \end{aligned} \tag{2.46}$$

Since a state-space transformation does not change the eigenvalues, the eigenvalues of $\bar{L}$ are the eigenvalues $\lambda_2, \cdots, \lambda_N$ of L. If the graph has a spanning tree, therefore, $-\bar{L}$ has all eigenvalues in the left-half plane and so is asymptotically stable. This means that the weighted centroid remains stationary, while, if there is a spanning tree, the error vector $\tilde{x}(t)$ converges to zero. Vector $\tilde{x}(t)$ is called the shape vector and it indicates the spread of the node error vectors about the centroid.

2.5. Consensus with First-Order Discrete-Time Dynamics

Now suppose each node or agent has scalar discrete-time (DT) dynamics given by

$$x_i(k+1) = x_i(k) + u_i(k) \tag{2.47}$$

with $x_i, u_i \in R$. This corresponds to endowing each node with a memory in the form of a shift register of order one. We discuss two ways to select the distributed control input protocols $u_i(k)$ [7],[10],[11],[16].

2.5.1 Perron Discrete-time Systems

Consider the distributed local control protocols

$$u_i(k) = \varepsilon \sum_{j \in N_i} a_{ij}(x_j(k) - x_i(k)) \tag{2.48}$$

with a_{ij} the graph edge weights and $\varepsilon > 0$. The closed-loop system becomes

$$\begin{aligned} x_i(k+1) &= x_i(k) + \varepsilon \sum_{j \in N_i} a_{ij}(x_j(k) - x_i(k)) \\ &= (1 - \varepsilon d_i)x_i(k) + \varepsilon \sum_{j \in N_i} a_{ij}x_j(k) \end{aligned} \tag{2.49}$$

The corresponding global input is

$$u(k) = -\varepsilon L x(k) \tag{2.50}$$

and the global dynamics is

$$x(k+1) = (I - \varepsilon L)x(k) \equiv P x(k) \tag{2.51}$$

with $u = [u_1 \cdots u_N]^T \in R^N$, $x = [x_1 \cdots x_N]^T \in R^N$.
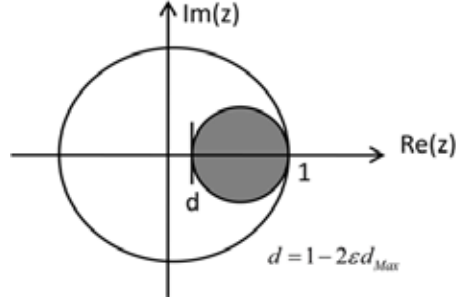
Fig. 2.11 Region of eigenvalues of Perron matrix P.

The matrix $P = I - \varepsilon L$ is known as the Perron matrix. Laplacian L has one eigenvalue at $s=0$ and the remaining eigenvalues in the right half of the s-plane, as seen in Figure 2.3. Therefore, $-L$ has all eigenvalues in the open left half-plane, and P has all eigenvalues in the shaded region of the z-plane shown in Figure 2.11. If ε is small enough, P has all eigenvalues inside the unit circle. Then, if the graph has a spanning tree, it has a simple eigenvalue $\lambda_1 = 1$ at $z = 1$, and the rest strictly inside the unit circle. Then, (2.51) is marginally stable and of Type 1, and a steady-state value is reached. A sufficient condition for P to have all eigenvalues in the unit circle is

$$\varepsilon < 1/d_{Max} \tag{2.52}$$

Note that the conditions for stability of each agent's dynamics (2.49) are $\varepsilon < 1/d_i$.

Since Laplacian matrix L has row sums of zero, P has row sums of 1, and so is a row stochastic matrix. That is,

$$P\underline{1} = \underline{1} \tag{2.53}$$

$$(I - P)\underline{1} = 0 \tag{2.54}$$

and $\underline{1}$ is the right eigenvector of $\lambda_1 = 1$. Let w_1 be a left eigenvector of L for $\lambda_1 = 0$. Then $w_1^T P = w_1^T (I - \varepsilon L) = w_1^T$ so that w_1 is a left eigenvector of P for $\lambda_1 = 1$.

If the system (2.51) reaches steady state, then

$$x_{ss} = Px_{ss} \tag{2.55}$$

If the graph has a spanning tree, then the only solution of this is $x_{ss} = c\underline{1}$ for some $c > 0$. Then, consensus is reached so that $x_i = x_j = c, \forall i, j$.

Let $w_1 = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix}^T$ be a left eigenvector of P for $\lambda_1 = 1$, not necessarily normalized. Then

$$w_1^T x(k+1) = w_1^T Px(k) = w_1^T x(k) \tag{2.56}$$

so that the quantity

$$\bar{x} \equiv w_1^T x = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \sum_i p_i x_i \tag{2.57}$$

is invariant. That is, the quantity $\bar{x} = \sum_i p_i x_i(k)$ is a constant of the motion. Thus, $\sum_i p_i x_i(0) = \sum_i p_i x_i(k)$, $\forall k$. Therefore, if the graph has a spanning tree, at steady-state one has consensus so that $x_i = x_j = c, \forall i, j$ where the consensus value is given by (2.39), the normalized linear combination of the initial states weighted by the elements of the left eigenvector of P for $\lambda_1 = 1$. This depends on the graph topology and hence on how the nodes communicate.

If the graph is balanced, then row sums of L are equal to column sums, and this property carries over to P. Then $w_1 = c\underline{1}$ is a left eigenvector of P for $\lambda_1 = 1$ and the consensus value is the average of the initial conditions (2.31), which is independent of the graph topology.

2.5.2 Normalized Protocol Discrete-time Systems

Now consider the local control protocols

$$u_i(k) = \frac{1}{1+d_i} \sum_{j \in N_i} a_{ij}(x_j(k) - x_i(k)) \tag{2.58}$$

which are normalized by dividing by $1+d_i$, with d_i the in-degree of node i. Thus, nodes with larger in-degrees must use less relative control effort than in the protocol (2.48). The closed-loop system becomes

$$x_i(k+1) = x_i(k) + \frac{1}{1+d_i} \sum_{j\in N_i} a_{ij}(x_j(k)-x_i(k)) = \frac{1}{1+d_i}\left(x_i(k) + \sum_{j\in N_i} a_{ij}x_j(k) \right)$$

$$(2.59)$$

which is a weighted average of the previous states of node i and its neighbors. The corresponding global input is

$$u(k) = -(I+D)^{-1}Lx(k) \qquad (2.60)$$

with $D = diag\{d_i\}$. The global dynamics is

$$x(k+1) = x(k) - (I+D)^{-1}Lx(k) = (I+D)^{-1}(I+A)x(k) \equiv Fx(k) \quad (2.61)$$

where the normalized DT consensus matrix is

$$F = I - (I+D)^{-1}L = (I+D)^{-1}(I+A) \qquad (2.62)$$

According to the Geršgorin Circle Criterion the normalized Laplacian matrix $(I+D)^{-1}L$ has all eigenvalues in the right half s-plane within a disc centered at $d_{Max}/(1+d_{Max})$ with radius equal to $d_{Max}/(1+d_{Max})$. Therefore, the F matrix has eigenvalues inside the shaded region in Figure 2.12. Since $d_{Max}/(1+d_{Max})<1$, *this region is always inside the unit circle.* Therefore, if the graph has a spanning tree, F has a simple eigenvalue at $\lambda_1 = 1$ and all other eigenvalues strictly inside the unit circle. Then, the system (2.61) is marginally stable and of Type I, and the state reaches a steady-state value.
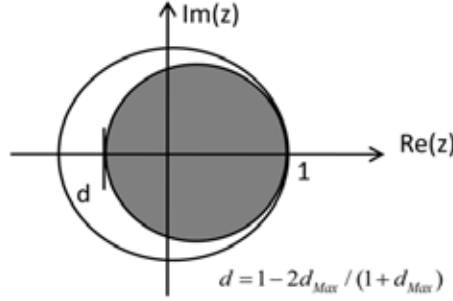
Fig. 2.12 Region of eigenvalues of discrete-time graph matrix F.

Since Laplacian matrix L has row sums of zero, F has row sums of 1, and so is a row stochastic matrix since

$$F\underline{1} = \left(I - (I+D)^{-1}L\right)\underline{1} = \underline{1}$$

(2.63)

Then, $\underline{1}$ is the right eigenvector of F for $\lambda_1 = 1$.

Let $w_1 = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix}^T$ be a left eigenvector of F for $\lambda_1 = 1$, not necessarily normalized. Then

$$w_1^T x(k+1) = w_1^T F x(k) = w_1^T x(k)$$

(2.64)

so that the quantity (2.57) is invariant, but now p_i are defined with respect to matrix F. Then, the quantity $\overline{x} = \sum_i p_i x_i(k)$ is a constant of the motion. If the graph has a spanning tree, the consensus value $x_i = x_j = c, \forall i, j$ is given by (2.39) in terms of the elements of the left eigenvector for $\lambda_1 = 1$ of F.

If the graph is balanced, however, then row sums of L are equal to column sums and $c\underline{1}$ is a left eigenvector of L for $\lambda_1 = 0$. However, in $(I+D)^{-1}L$ the i-th row is divided by $1+d_i$. Therefore, $(I+D)^{-1}L$ is not balanced. Hence, F is *not balanced* and so even for balanced graphs the average consensus (2.31) is not reached. The consensus value of (2.61) depends on graph properties even for balanced graphs. This is true also for general undirected graphs. However, if all nodes have the same in-degree, then L balanced implies that

$(I + D)^{-1} L$ is balanced. Hence F is balanced and average consensus is reached. An undirected graph where all nodes have the same degree d is called d-regular.

Let w_1 be a left eigenvector of L for $\lambda_1 = 0$. Then w_1 is not a left eigenvector of F for $\lambda_1 = 1$ unless the graph is regular. Then, it can be checked that

$$w_i^T F = w_i^T \left(I - (I + D)^{-1} L \right) = w_i^T .$$

2.5.3 Average Consensus Using Two Parallel Protocols at Each Node

The average consensus problem yields the average of the initial states (2.31), independently of the left eigenvector of $\lambda_1 = 1$ and hence of the graph topology. This is important in a variety of applications. It has been seen that if the graph is balanced, the continuous-time protocol (2.16) and the Perron protocol (2.49) both yield average consensus (2.31) independently of the detailed topology of the graph. The normalized DT protocol (2.59) does not yield average consensus unless the graph is regular (all nodes have the same in-degree). There are several ways to obtain average consensus for this protocol [12].

Suppose each node i knows its component p_i of the normalized left eigenvector w_1 of F for $\lambda_1 = 1$, and the number of nodes N in the graph. Suppose it is desired to reach average consensus on a prescribed vector $x(0) = \begin{bmatrix} x_1(0) & x_2(0) & \cdots & x_N(0) \end{bmatrix}^T$. Then, let each node run the normalized protocol (2.59) with initial state selected as $x_i(0) / N p_i$. If the graph has a spanning tree, the weighted average consensus (2.39) is achieved. Substituting the selected initial conditions into (2.39) yields average consensus (2.31).

The requirement for each node to know its own p_i as well as the number of nodes N in the graph assumes that there is a central authority which computes the graph structure and disseminates this global information, which is contrary to the idea of distributed cooperative control. It is possible in some graph topologies for *each node to estimate global graph structural information* by using multiple protocols at each node, as follows.

Let each node be endowed with two states $y_i(k), z_i(k)$, each running local protocols in the form (2.59) so that

$$y_i(k+1) = \frac{1}{1+d_i} \left(y_i(k) + \sum_{j \in N_i} a_{ij} y_j(k) \right) \qquad (2.65)$$

$$z_i(k+1) = \frac{1}{1+d_i}\left(z_i(k) + \sum_{j\in N_i} a_{ij} z_j(k)\right) \tag{2.66}$$

with prescribed initial states $y_i(0), z_i(0)$. The global dynamics of $y = [y_1 \cdots y_N]^T \in R^N$, are then

$$y(k+1) = (I+D)^{-1}(I+A)y(k) \equiv Fy(k) \tag{2.67}$$

$$z(k+1) = (I+D)^{-1}(I+A)z(k) \equiv Fz(k) \tag{2.68}$$

Now suppose the graph has a special topology known as the bidirectional equal-neighbor weight topology [12]. In this topology, one has $a_{ij} \neq 0 \Rightarrow a_{ji} \neq 0$ and $a_{ij} \neq 0 \Rightarrow a_{ij} = 1$. Thus, each node equally weights its own current state and its neighbors' current states by $1/(1+d_i)$ in computing the next state. For this special topology, it is easy to give explicit formulae for the components p_i of left eigenvector w_1. Define the (modified) volume of the graph as $Vol' = \sum_{i=1}^{N}(1+d_i)$. Then, for the bidirectional equal-neighbor weight topologies one has

$$p_i = \frac{1+d_i}{Vol'} \tag{2.69}$$

This is easily verified by verifying that $w_1^T F = w_1^T (I+D)^{-1}(I+A) = w_1^T$ for the given graph topology and left eigenvector components. Note that (2.69) gives a normalized w_1 since $\sum_i p_i = 1$.

Now suppose it is desired to reach average consensus on a prescribed vector $x(0) = [x_1(0) \quad x_2(0) \quad \cdots \quad x_N(0)]^T$. Then, select initial states $y_i(0) = 1/(1+d_i)$, $z_i(0) = x_i(0)/(1+d_i)$ This is local information available to each node i. If the graph has a spanning tree, then according to (2.39) the two protocols (2.65), (2.66) reach respective consensus values of

$$y_i(k) \to \frac{N}{Vol'}, \quad z_i(k) \to \frac{1}{Vol'}\sum_i x_i(0) \tag{2.70}$$

Then, the estimate of the average consensus value is given by

$$x_i(k) = \frac{z_i(k)}{y_i(k)} \to \frac{1}{N}\sum_i x_i(0) \tag{2.71}$$

This procedure results in average consensus independent of the graph topology. This has been achieved by running two local voting protocols at each node, one of which, $y_i(k)$, estimates the global graph property N/Vol'. This is used to divide out the volume in the second protocol for $z_i(k)$.

2.6. Higher-Order Consensus: Linear Systems on Graphs

We have just discussed cooperative control on graphs for dynamical systems that have first-order dynamics, that is, a single integrator or discrete-time shift register at each node. Now we discuss cooperative control for networked systems with higher-order dynamics. In this case, conditions for consensus become more complex and interesting. First we consider linear systems, then systems having second-order position/velocity type dynamics in the form of Newton's third law.

In this section we use A to denote the node dynamics system matrix, and a_{ij} to denote the graph edge weights. These should not be confused.

Consider the N systems distributed on communication graph G with identical linear time-invariant node dynamics

$$\dot{x}_i = Ax_i + Bu_i \tag{2.72}$$

where $x_i(t) \in R^n$ is the state of node i and $u_i(t) \in R^m$ its control input. Select the node controls as the distributed local state variable feedback

$$u_i = cK\sum_{j\in N_i} a_{ij}(x_j - x_i) \tag{2.73}$$

with scalar coupling gain $c > 0$ and feedback control matrix $K \in R^{m\times n}$. Here, a_{ij} are the elements of the graph adjacency matrix, not the node system matrix.

That is, they are the edge weights of graph G. These controls are based on the local voting protocol structure where the control input of each node depends on the difference between its state and those of all its neighbors. The coupling gains c and feedback gains K of all nodes are the same.

The node closed-loop dynamics are

$$\dot{x}_i = Ax_i + Bu_i = Ax_i + cBK \sum_{j \in N_i} a_{ij}(x_j - x_i) \qquad (2.74)$$

The overall closed-loop graph dynamics is

$$\dot{x} = \left[(I_N \otimes A) - cL \otimes BK\right]x \qquad (2.75)$$

where the overall (global) state vector is $x = \begin{bmatrix} x_1^T & x_2^T & \cdots & x_N^T \end{bmatrix}^T \in R^{nN}$, L is the graph Laplacian matrix, and $\otimes$ is the Kronecker product [1]. We use the Kronecker product heavily in this section, mainly the property $(M \otimes N)(P \otimes Q) = MP \otimes NQ$ for commensurate matrices M,N,P,Q.

Define the *cooperative feedback closed-loop system matrix* on communication graph G as

$$A_c = \left[(I_N \otimes A) - cL \otimes BK\right] \qquad (2.76)$$

This matrix reflects the local closed-loop system matrix $(A - BK)$ as modified by distributed control on the graph structure L. Its stability depends on both the stability properties of $(A - BK)$ and the graph topology properties.

The next key result from [4] provides a method for analysis of the dynamics (2.74)/(2.75) and relates their stability properties to the graph eigenvalues.

Lemma 2.1. Stability Condition for Cooperative Control. *Let $\lambda_i, i = 1, \cdots, N$ be the eigenvalues of the graph Laplacian matrix L. Then the stability properties of dynamics (2.74)/(2.75) are equivalent to the stability properties of the N systems*

$$A - c\lambda_i BK, \quad i = 1, \cdots, N \qquad (2.77)$$

in that they have the same eigenvalues.

Proof. Define a transformation M such that $J = M^{-1}LM$ is upper triangular with the eigenvalues $\lambda_i, i = 1, \cdots, N$ of L on the diagonal. Apply the state-space transformation

$$(M^{-1} \otimes I)\left[(I_N \otimes A) - cL \otimes BK\right](M \otimes I)$$
$$= \left[(I_N \otimes A) - cJ \otimes BK\right]$$
(2.78)

and define new state $\xi = (M^{-1} \otimes I)\delta$. Then the transformed system is in block triangular form with diagonal blocks of the form

$$\dot{\xi}_i = (A - c\lambda_i BK)\xi_i$$
(2.79)

Hence, stability of (2.74)/(2.75) is equivalent to stability of all of these systems, since a state space transformation does not change the eigenvalues. ∎

This proof shows that $A_c = \left[(I_N \otimes A) - cL \otimes BK\right]$ is equivalent to the system

$$diag\{(A - c\lambda_1 BK), (A - c\lambda_2 BK), \cdots (A - c\lambda_N BK)\}$$
(2.80)

with $\lambda_i, i = 1, \cdots, N$ the eigenvalues of the graph Laplacian matrix L, in the sense that they have the same eigenvalues.

Condition (2.77) is complicated and means that consensus for a given feedback matrix K may occur on some graphs but not on others. To guarantee consensus on a given graph with given $\lambda_i, i = 1, \cdots, N$, matrix K must be selected to stabilize all the systems (2.77). This is a complicated robust control problem. In subsequent chapters we show how to select K to obtain guaranteed consensus properties. Now, we apply Lemma 2.1 to second-order consensus.

2.7. Second-Order Consensus

One application of consensus in cooperative control is the control of formations of vehicles. Therefore, we now wish to study consensus for coupled systems that satisfy Newton's law $\ddot{x}_i = u_i$, which yields the second-order node dynamics

$$\dot{x}_i = v_i$$
$$\dot{v}_i = u_i \qquad\qquad (2.81)$$

with position $x_i \in R$, speed $v_i \in R$, and acceleration input $u_i \in R$. Consider the distributed position/velocity feedback at each node given by the second-order local neighborhood protocols

$$u_i = c \sum_{j \in N_i} a_{ij}(x_j - x_i) + c\gamma \sum_{j \in N_i} a_{ij}(v_j - v_i) = \sum_{j \in N_i} ca_{ij}\left((x_j - x_i) + \gamma(v_j - v_i)\right)$$

$$(2.82)$$

where $c > 0$ is a stiffness gain and $c\gamma > 0$ is a damping gain. This is based on local voting protocols in both position and speed, so that each node seeks to match all its neighbors' positions as well as their velocities. This is a variant of proportional-plus-derivative control.

We would like to determine when this protocol delivers consensus, and to find the consensus values of position and speed. We analyze this protocol by two methods.

2.7.1 Analysis of Second-Order Consensus Using Position/Velocity Local Node States

The first method for analyzing second-order consensus follows [24]. Define the position/velocity local node states as $z_i = \begin{bmatrix} x_i & v_i \end{bmatrix}$ and write the position/velocity node dynamics at each node as

$$\dot{z}_i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z_i + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_i = Az_i + Bu_i \qquad\qquad (2.83)$$

Define the local distributed control protocols

$$u_i = c\begin{bmatrix} 1 & \gamma \end{bmatrix} \sum_{j \in N_i} a_{ij} \begin{bmatrix} x_j - x_i \\ v_j - v_i \end{bmatrix} = cK \sum_{j \in N_i} a_{ij}(z_j - z_i) \qquad\qquad (2.84)$$

where the feedback gain matrix is $K = \begin{bmatrix} 1 & \gamma \end{bmatrix}$. This is state variable feedback at each node.

Now identify this with (2.74), define $z = \begin{bmatrix} z_1^T & z_2^T & \cdots & z_N^T \end{bmatrix}^T \in R^{2N}$, and write the global closed-loop dynamics (2.75) as

$$\dot{z} = \left[(I_N \otimes A) - cL \otimes BK \right] z = A_c z \qquad (2.85)$$

for the specific A,B,K just given.

Assume the graph has a spanning tree. Then, L has a simple eigenvalue at $\lambda_1 = 0$, and hence has rank N-1, and the rest of its eigenvalues are strictly in the right-half of the s-plane. Under these conditions we want to study consensus properties of the protocol (2.82). First, we need to look at the stability properties of (2.85), and second to find the position and speed consensus values.

To examine stability of the protocol, according to the proof of Lemma 2.1, A_c in (2.85) is equivalent to

$$diag\{A, (A - c\lambda_2 BK), \cdots (A - c\lambda_N BK)\} \qquad (2.86)$$

with $\mathrm{Re}\{\lambda_i\} > 0, i = 2, \cdots, N$. Matrix A has two eigenvalues at $\mu_1 = 0$ with geometric multiplicity equal to one and a Jordan block of order two. The characteristic polynomials of the other blocks are

$$\left| sI - (A - c\lambda_i BK) \right| = s^2 + c\gamma\lambda_i s + c\lambda_i \qquad (2.87)$$

Then the characteristic polynomial of A_c is

$$\left| sI - A_c \right| = \prod_{i=1}^{N} (s^2 + c\gamma\lambda_i s + c\lambda_i) \qquad (2.88)$$

Therefore, the eigenvalues of A_c are given by

$$s = -\frac{1}{2}\left(c\gamma\lambda_i \pm \sqrt{(c\gamma\lambda_i)^2 - 4c\lambda_i} \right) \qquad (2.89)$$

There are two eigenvalues of A_c for each eigenvalue λ_i of L.

The stability of the eigenvalues of A_c can be studied using the Routh test. Suppose graph eigenvalue λ_i is real. Then the Routh test shows that (2.87) is asymptotically stable (AS) for all $c\gamma > 0$. If λ_i is complex, then

$$
\begin{aligned}
&(s^2 + c\gamma\lambda_i s + c\lambda_i)(s^2 + c\gamma\lambda_i^* s + c\lambda_i^*) \\
&= s^4 + 2c\gamma\alpha s^3 + (2c\alpha + c^2\gamma^2\mu^2)s^2 + 2c^2\gamma\mu^2 s + c^2\mu^2
\end{aligned}
\tag{2.90}
$$

where * denotes complex conjugate, $\alpha = \mathrm{Re}\{\lambda_i\}$, $\mu = |\lambda_i|$. In the case $c=1$, one can obtain explicit expression for stability from the Routh test. When $c=1$, performing the Routh test shows that (2.90) is AS if and only if

$$
\gamma^2 > (\beta^2 - \alpha^2)/\alpha\mu^2
\tag{2.91}
$$

where $\beta = \mathrm{Im}\{\lambda_i\}$. Therefore, all the systems in (2.86) for $i = 2, \cdots, N$ are AS if and only if

$$
c\gamma^2 > \max_i \frac{\mathrm{Im}^2\{\lambda_i\} - \mathrm{Re}^2\{\lambda_i\}}{\mathrm{Re}\{\lambda_i\}|\lambda_i|^2}, \quad i = 2, \cdots, N
\tag{2.92}
$$

The next result tells when the nodes reach consensus under the second-order protocol (2.82), and specifies the consensus values reached.

Theorem 2.4. Consensus of Second-Order Dynamics. *Consider the Newton's law systems at each node given by (2.81) with the local neighborhood protocols (2.82) with c=1. The nodes reach consensus in both position and speed if and only if the graph has a spanning tree and the gain γ is selected to satisfy condition (2.92), with λ_i the eigenvalues of the graph Laplacian L. Then, the position and speed consensus values are given respectively by*

$$
\bar{x} = \tfrac{1}{N}\sum_{i=1}^{N} p_i x_i(0) + \tfrac{1}{N}t\sum_{i=1}^{N} p_i v_i(0)
\tag{2.93}
$$

$$
\bar{v} = \tfrac{1}{N}\sum_{i=1}^{N} p_i v_i(0)
\tag{2.94}
$$

Proof. It has been seen that, when c=1, (2.85) is AS if and only if (2.92) holds. Now, it is necessary to examine the eigenstructure of A_c in (2.85). First we need to find the left and right eigenvectors of A_c, which is equivalent to (2.86) in the sense that they have the same eigenvalues. Matrix A in (2.83) has an eigenvalue at $\mu_1 = 0$ of geometric multiplicity one and algebraic multiplicity 2. The rank-1 right eigenvector is $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$ since $A\begin{bmatrix} 1 & 0 \end{bmatrix}^T = 0$. The rank-2 right eigenvector is $\begin{bmatrix} 0 & 1 \end{bmatrix}^T$ since $A\begin{bmatrix} 0 & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$. The rank-1 left eigenvector is $\begin{bmatrix} 0 & 1 \end{bmatrix}^T$ since $\begin{bmatrix} 0 & 1 \end{bmatrix}A = 0$. The rank-2 left eigenvector is $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$ since $\begin{bmatrix} 1 & 0 \end{bmatrix}A = \begin{bmatrix} 0 & 1 \end{bmatrix}$.

The right eigenvector of L for $\lambda_1 = 0$ is $\underline{1} \in R^N$. Let the corresponding left eigenvector be $w_1 = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix}^T$, normalized so that $w_1^T \underline{1} = 1$. Then we claim a rank-1 right eigenvector of A_c in (2.85) for $\lambda_1 = 0$ is $\overline{y}_1^1 = \underline{1} \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}^T$. This is easily verified by checking that $A_c \overline{y}_1^1 = 0$. It is easy to verify that a rank-2 right eigenvector of A_c is $\overline{y}_1^2 = \underline{1} \otimes \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ by checking that $A_c \overline{y}_1^2 = \overline{y}_1^1$. Similarly, it is found that a rank-1 left eigenvector of A_c for $\lambda_1 = 0$ is given by $\overline{w}_1^1 = w_1 \otimes \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ and a rank-2 left eigenvector by $\overline{w}_1^2 = w_1 \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}^T$.

Now we will study the Jordan normal form of A_c and perform a modal decomposition on (2.85) to find the steady-state consensus values for position and speed. Bring A_c to Jordan normal form

$$A_c = MJM^{-1} = \begin{bmatrix} \overline{y}_1^1 & \overline{y}_1^2 & \cdots \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & stable \end{bmatrix} \begin{bmatrix} (\overline{w}_1^2)^T \\ (\overline{w}_1^1)^T \\ \vdots \end{bmatrix} \qquad (2.95)$$

where M is the matrix having right eigenvectors as columns, and M^{-1} is the matrix having left eigenvectors as rows. Then

$$z(t) = e^{A_c t} z(0) = M e^{Jt} M^{-1} z(0) = \begin{bmatrix} \overline{y}_1^1 & \overline{y}_1^2 & \cdots \end{bmatrix} \begin{bmatrix} 1 & t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & stable \end{bmatrix} \begin{bmatrix} (\overline{w}_1^2)^T \\ (\overline{w}_1^1)^T \\ \vdots \end{bmatrix} z(0)$$

$$(2.96)$$

yields the modal decomposition

$$z(t) = \begin{bmatrix} \overline{y}_1^1 & \overline{y}_1^2 & \cdots \end{bmatrix} \begin{bmatrix} (\overline{w}_1^2)^T z(0) + t(\overline{w}_1^1)^T z(0) \\ (\overline{w}_1^1)^T z(0) \\ \vdots \end{bmatrix}$$

$$(2.97)$$

$$= \left((\overline{w}_1^2)^T z(0) + t(\overline{w}_1^1)^T z(0) \right) \overline{y}_1^1 + (\overline{w}_1^1)^T z(0) \overline{y}_1^2 + stable\ terms$$

Note now that

$$z(t) = x(t) \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + v(t) \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad (2.98)$$

with global position and speed vectors $x = \begin{bmatrix} x_1^T & x_2^T & \cdots & x_N^T \end{bmatrix}^T \in R^N$, $v = \begin{bmatrix} v_1^T & v_2^T & \cdots & v_N^T \end{bmatrix}^T \in R^N$. According to the definitions of the left and right eigenvectors of A_c for $\lambda_1 = 0$, one sees that the position and speed consensus values induced by the protocol (2.82) are (2.93), (2.94).

∎

It is seen that condition (2.92) depends on the graph topology through the eigenvalues of the graph Laplacian matrix L. Therefore, for a given feedback gain matrix K, one may have consensus on one graph but not on a different graph.

If there is no spanning tree, then $\lambda_2 = 0$ in (2.86). Then, there are unstable terms in (2.96) and consensus is not reached.

If the graph is strongly connected then it has a spanning tree and Theorem 2.4 holds. Moreover, if the initial velocities are all equal to zero, then the consensus speed is equal to zero and all nodes end up at rest at the weighted center of gravity of the initial positions, that is, the first term in (2.93). The next result has also been established.

Corollary 2.1. *Let the graph have a spanning tree and all eigenvalues of the Laplacian L be real. Then the consensus values (2.93), (2.94).are reached for any positive control gains c, γ in (2.82).*

Proof. The Routh test reveals that (2.87) is asymptotically stable under the hypotheses. ∎

If the graph is undirected, then the Laplacian eigenvalues are real and this corollary holds.

2.7.2 Analysis of Second-Order Consensus Using Position/Velocity Global State

The second method for analyzing second-order consensus follows [18],[16]. It hinges on another way to write the global dynamics than (2.85).

Take Newton's law node dynamics (2.81) and the second-order protocols (2.82). Define the global position and speed vectors as $x = \begin{bmatrix} x_1^T & x_2^T & \cdots & x_N^T \end{bmatrix}^T \in R^N$, $v = \begin{bmatrix} v_1^T & v_2^T & \cdots & v_N^T \end{bmatrix}^T \in R^N$. Define the global state vector in position/velocity form as $\overline{z} = \begin{bmatrix} x^T & v^T \end{bmatrix}^T \in R^{2N}$. Then the global dynamics can be written as

$$\dot{\overline{z}} = \begin{bmatrix} 0 & I \\ -cL & -c\gamma L \end{bmatrix} \overline{z} \equiv \overline{A}_c z \tag{2.99}$$

This is in global position/velocity form, whereas (2.85) has the global state defined in terms of the interleaved node positions and velocities according to the local position/velocity state definitions (2.83).

The characteristic polynomial of this system is

$$\left| sI - \overline{A}_c \right| = \begin{vmatrix} sI & -I \\ cL & sI + c\gamma L \end{vmatrix} = \left| s^2 I + c\gamma Ls + cL \right| \tag{2.100}$$

As in the proof of Lemma 2.1, define a transformation M such that $J = M^{-1}LM$ is upper triangular with the eigenvalues $\lambda_i, i = 1, \cdots, N$ of L on the diagonal. Write

$$\left|s^2 I + c\gamma Ls + cL\right| = \left|M\right| \cdot \left|M^{-1}\right| \cdot \left|s^2 I + c\gamma Ls + cL\right|$$

$$= \left|M^{-1}(s^2 I + c\gamma Ls + cL)M\right| = \left|s^2 I + c\gamma Js + cJ\right| \qquad (2.101)$$

Matrix $(s^2 I + c\gamma Js + cJ)$ is in block triangular form with the eigenvalues λ_i of L on the diagonal. Consequently

$$\left|s^2 I + c\gamma Ls + cL\right| = \prod_{i=1}^{N}(s^2 + c\gamma\lambda_i s + c\lambda_i) \qquad (2.102)$$

Therefore, the eigenvalues of $\overline{A}_c$ are given by

$$s = -\tfrac{1}{2}\left(c\gamma\lambda_i \pm \sqrt{(c\gamma\lambda_i)^2 - 4c\lambda_i}\right) \qquad (2.103)$$

and there are two eigenvalues of $\overline{A}_c$ for each eigenvalue λ_i of L.

These results are the same as (2.88). Therefore all the subsequent development there follows here, eventually providing another path to Theorem 2.4.

2.7.3 Formation Control Second-Order Protocol

In the control of formations it is desired for all vehicles to reach the same consensus speed, but also for the steady-state positions to be in some prescribed formation relative to each other. The protocol just given causes all nodes to move to the same final position. Moreover, in formation control the positions are in 2-D or 3-D.

Therefore, let the formation move in an n-dimensional space and consider the node dynamics

$$\begin{aligned}
\dot{x}_i &= v_i \\
\dot{v}_i &= u_i
\end{aligned} \qquad (2.104)$$

with vector position $x_i \in R^n$, speed $v_i \in R^n$, and acceleration input $u_i \in R^n$. For motion in the 2-D plane, for instance, one would take $x_i = \begin{bmatrix} p_i & q_i \end{bmatrix}^T$ where $(p_i(t), q_i(t))$ is the position of node i in the (x,y)-plane. The node states are $z_i = \begin{bmatrix} x_i^T & v_i^T \end{bmatrix}^T \in R^{2n}$ and the local node dynamics are given by

$$\dot{z}_i = \begin{bmatrix} 0 & I_n \\ 0 & 0 \end{bmatrix} z_i + \begin{bmatrix} 0 \\ I_n \end{bmatrix} u_i = A z_i + B u_i \qquad (2.105)$$

Define the constant desired position of node i relative to the moving formation center x_0 as $\Delta_i \in R^n$. Suppose it is desired for the nodes to follow a leader or control node with position and speed given by $\dot{x}_0 = v_0$. Consider the distributed second-order leader-following formation protocols at each node given by [16]

$$
\begin{aligned}
u_i &= \dot{v}_0 + \gamma k_v (v_0 - v_i) + k_p (x_0 + \Delta_i - x_i) \\
&\quad + c \sum_{j \in N_i} a_{ij} \big((x_j - \Delta_j) - (x_i - \Delta_i) \big) + c\gamma \sum_{j \in N_i} a_{ij} (v_j - v_i) \\
&= \dot{v}_0 + \gamma k_v (v_0 - v_i) + k_p (x_0 + \Delta_i - x_i) \\
&\quad + \sum_{j \in N_i} c a_{ij} \big((x_j - \Delta_j) - (x_i - \Delta_i) + \gamma (v_j - v_i) \big)
\end{aligned}
\qquad (2.106)
$$

where $c > 0$ is a stiffness gain and $\gamma > 0$ is a damping gain. This protocol includes leader's velocity feedback with gain of $\gamma k_v > 0$, leader's position feedback with gain of $k_p > 0$, and leader's acceleration feedforward. The aim of this protocol is to ensure that all nodes reach consensus in position and speed and that $x_i(t) \to x_0(t) + \Delta_i$, $v_i(t) \to v_0(t)$, $\forall i$ asymptotically.

Define the motion errors $\tilde{x}_i = x_i - \Delta_i - x_0$, $\tilde{v}_i = v_i - v_0$ and the node error $\tilde{z}_i = \begin{bmatrix} \tilde{x}_i^T & \tilde{v}_i^T \end{bmatrix}^T \in R^{2n}$ Then the closed-loop node error dynamics are

$$\dot{\tilde{z}}_i = \begin{bmatrix} 0 & I_n \\ -k_p I_n & -\gamma k_v I_n \end{bmatrix} \tilde{z}_i + \begin{bmatrix} 0 \\ I_n \end{bmatrix} \sum_{j \in N_i} c a_{ij} \big((x_j - \Delta_j) - (x_i - \Delta_i) + \gamma (v_j - v_i) \big)$$

$$(2.107)$$

which yields the global error dynamics

$$\dot{\tilde{z}} = \big[(I_N \otimes \overline{A}) - cL \otimes BK \big] \tilde{z} \equiv \overline{A}_c \tilde{z} \qquad (2.108)$$

with $\tilde{z} = \begin{bmatrix} \tilde{z}_1^T & \tilde{z}_2^T & \cdots & \tilde{z}_N^T \end{bmatrix}^T \in R^{2nN}$, $K = \begin{bmatrix} I_n & \gamma I_n \end{bmatrix}$, and

$$\bar{A} = \begin{bmatrix} 0 & I_n \\ -k_p I_n & -\gamma k_v I_n \end{bmatrix} \tag{2.109}$$

We want to examine stability of the protocol and determine when $x_i(t) \to x_0(t) + \Delta_i$, $v_i(t) \to v_0(t)$, $\forall i$. Assume the graph has a spanning tree. To examine stability, according to the proof of Lemma 2.1, $\bar{A}_c$ in (2.108) is equivalent to

$$diag\{\bar{A}, (\bar{A} - c\lambda_2 BK), \cdots (\bar{A} - c\lambda_N BK)\} \tag{2.110}$$

with λ_i the eigenvalues of Laplacian matrix L and $\text{Re}\{\lambda_i\} > 0$, $i = 2, \cdots, N$.

Without loss of generality, assume motion in 1-D and set $n = 1$. Motion in other coordinate directions will be the same as the motion along a line as analyzed here. Matrix $\bar{A}$ has the characteristic polynomial $s^2 + k_v s + k_p$, which is stable for all $k_p, k_v > 0$. The characteristic polynomials of the other blocks are

$$\left| sI - (A - c\lambda_i BK) \right| = s^2 + \gamma(k_v + c\lambda_i)s + (k_p + c\lambda_i) \tag{2.111}$$

Then the characteristic polynomial of A_c is

$$\left| sI - A_c \right| = \prod_{i=1}^{N} (s^2 + \gamma(k_v + c\lambda_i)s + (k_v + c\lambda_i)) \tag{2.112}$$

Therefore, the eigenvalues of A_c are given by

$$s = -\tfrac{1}{2}\left((k + c\gamma\lambda_i) \pm \sqrt{\gamma^2(k_v + c\gamma\lambda_i)^2 - 4(k_p + c\lambda_i)} \right) \tag{2.113}$$

There are two eigenvalues of A_c (in each dimension of motion) for each eigenvalue λ_i of L. Since the graph has a spanning tree we have $\lambda_1 = 0$ and $\lambda_i > 0$, $i = 2, \cdots, N$. Note that for $\lambda_1 = 0$, (2.111) is AS for $k_p > 0, k_v > 0$.

For real λ_i the Routh test shows that (2.111) is asymptotically stable (AS) for all $c\gamma > 0$, $\alpha > 0$, $k_p > 0, k_v > 0$. If λ_i is complex it is difficult to get a con-

dition for stabilizing gains. A complicated expression in the vein of (2.92) can be developed for stability. Simpler expressions can be developed for special choices of gains.

Theorem 2.5. Consensus of Formation Protocol. *Consider the Newton's law motion dynamics at each node given by (2.104). The leader-following protocols (2.106) guarantee that $x_i(t) \to x_0(t) + \Delta_i$, $v_i(t) \to v_0(t)$, $\forall i$ asymptotically if the graph has a spanning tree and the gains c, γ, k_p, k_v are selected to make (2.112) asymptotically stable.*

Proof. Under the hypotheses, the polynomials in (2.112) are asymptotically stable so that the error dynamics (2.108) are AS. Therefore, $\tilde{x}_i = x_i - \Delta_i - x_0$ and $\tilde{v}_i = v_i - v_0$ go to zero and the theorem follows.

∎

The proof of this result is far simpler than the proof of Theorem 2.4 because $\bar{A}$ in (2.109) has stable eigenvalues, whereas matrix A in (2.83) has an eigenvalue at $\mu_1 = 0$ of geometric multiplicity one and algebraic multiplicity two. This requires an analysis of the structure of the eigenspace for $\mu_1 = 0$ in the proof of Theorem 2.4.

The following result has also been established.

Corollary 2.2. Let the graph have a spanning tree and all eigenvalues of the Laplacian L be real. Then $x_i(t) \to x_0(t) + \Delta_i$, $v_i(t) \to v_0(t)$, $\forall i$ asymptotically for any gains $c, \gamma > 0$ $k_p, k_v > 0$.

Proof: The Routh test reveals that (2.111) is asymptotically stable under the hypotheses.

∎

If the graph is undirected, then the Laplacian eigenvalues are real and this corollary holds.

A disadvantage of protocol (2.106) is that all nodes must know the leader node's position $x_0(t)$ and speed $v_0(t)$. In realistic formations, each node only knows position and speed information about a few neighbors. Only a few immediate followers of the leader, his wingmen, can sense his motion, and other nodes follow these wingmen, and so on. We shall show how to correct this deficiency in Chapters 3 and the following chapters, when we discuss the cooperative tracking problem, also known as pinning control.

2.8. Matrix Analysis of Graphs

We have studied basic concepts of consensus and cooperative control for dynamic agents connected by a communication graph topology. It was seen that the graph properties and the properties of the individual node dynamics interact in intriguing ways that are not at all obvious to a casual inspection. The graph topology can have beneficial effects on individual node and team behaviors, such as inducing consensus under certain conditions, or detrimental effects, such as destabilizing the overall team dynamics though the individual node dynamics are stable.

Graph properties can be studied through the theory of certain matrix properties. In this section we introduce some concepts that are important for analysis of graphs and cooperative control performance using matrix techniques. The main references in this section are [6],[13],[23].

2.8.1 Irreducible Matrices and Frobenius Form

An arbitrary square matrix E is reducible if it can be brought by a row/column permutation matrix T to lower block triangular form

$$F = TET^T = \begin{bmatrix} * & 0 \\ * & * \end{bmatrix} \tag{2.114}$$

Otherwise the matrix is irreducible. Two matrices that are similar using permutation matrices are said to be cogredient. Note that, if T is a permutation matrix then $T^{-1} = T^T$.

The next result ties irreducible matrices to graph theory.

Theorem 2.6 [13]. *A graph G is strongly connected if and only if its adjacency matrix A is irreducible.*

Lemma 2.2. *A is irreducible if and only if D-A is irreducible for any diagonal matrix D.*

Proof: $TAT^T = \begin{bmatrix} * & 0 \\ * & * \end{bmatrix}$ if and only if $T(D-A)T^T = \begin{bmatrix} * & 0 \\ * & * \end{bmatrix}$ since TDT^T

is diagonal for all permutation matrices *T*. ∎

According to these results, both the adjacency matrix *A* and Laplacian matrix $L = D - A$ of a graph *G* are irreducible if and only if the graph is strongly connected.

Example 2.5. Irreducible Sets of Equations [21].

Consider the linear equations

$$Ex = b \qquad (2.115)$$

with E a matrix, not necessarily square, and x and b vectors. If E is nonsingular, there is a unique solution x. Yet, E may be singular and the system of equations can still enjoy some important properties. Let T be a permutation matrix and write

$$Ex = T^T FTx = b, \quad FTx = Tb, \quad F\bar{x} = \bar{b} \qquad (2.116)$$

where $\bar{x} = Tx, \bar{b} = Tb$. Suppose now that E is reducible and write

$$\begin{bmatrix} F_{11} & 0 \\ F_{21} & F_{22} \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} \bar{b}_1 \\ \bar{b}_2 \end{bmatrix} \qquad (2.117)$$

or

$$F_{11}\bar{x}_1 = \bar{b}_1, \quad F_{22}\bar{x}_2 = \bar{b}_2 - F_{21}\bar{x}_1 \qquad (2.118)$$

This means the equations can be solved by first solving a reduced set of equations for $\bar{x}_1$ and then solving for $\bar{x}_2$. On the other hand, if E is irreducible, the equations must be solved simultaneously for all variables. ∎

Frobenius Form and Hierarchical Structure of Graphs

A reducible matrix E can be brought by a permutation matrix T to the lower block triangular Frobenius canonical form [13],[23]

$$F = TET^T = \begin{bmatrix} F_{11} & 0 & \cdots & 0 \\ F_{21} & F_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ F_{p1} & F_{p2} & \cdots & F_{pp} \end{bmatrix} \qquad (2.119)$$

where F_{ii} is square and irreducible. Note that if F_{ii} is a scalar and irreducible it is equal to 0. The degree of reducibility is defined as $p-1$. The graph has p strongly connected subgraphs corresponding to the graphs of the diagonal blocks.

The conversion to lower block triangular form can be done in linear time using search algorithms.

The Frobenius form (2.119) is said to be lower triangularly complete if in every block row i there exists at least one $j < i$ such that $F_{ij} \neq 0$. That is, every block row has at least one nonzero entry.

Theorem 2.7. [13] *A graph G has a spanning tree if and only if its adjacency matrix A has a Frobenius form that is lower triangularly complete.*

Diagonal matrix F_{11} corresponds to a strongly connected leader group of root nodes with no incoming edges from the remainder of the group. If the graph has a spanning tree, then the nodes having a directed path to all other nodes are roots of spanning trees, and members of the leader group. These correspond to the graph of leader block F_{11} in Frobenius form.

The Frobenius form of a graph adjacency matrix exposes the hierarchical structure of a graph. Note that diagonal block F_{11} corresponds to a strongly connected leader group with no incoming edges from the remainder of the graph. It may be a set of root nodes for certain blocks F_{ii} as determined by the nonzero entries in blocks F_{ij}, $j < i$.

Any diagonal block F_{kk} that has no nonzero entries in its row blocks F_{kj}, $j < k$ has no incoming edges in the graph and is also a strongly connected leader group. It may be a set of root nodes for a subgraph determined by the elements of the off diagonal blocks.

Cyclic or Periodic Matrices

An important class of graphs are the cyclic or periodic graphs. A graph is a cycle if all of its vertices are connected in a single closed path. A more general class of periodic graphs are those which have groups of vertices, with the groups being connected by a closed path. These graphs can be studied using Frobenius form.

The spectral radius of a square matrix $E \in R^{n \times n}$ is $\rho(E) = \max_i |\lambda_i|$, the maximum magnitude of the eigenvalues of E.

Theorem 2.8. Perron-Frobenius [13],[23],[6]. *Let a square matrix $E \in R^{n \times n}$ have nonnegative elements . Then if E is irreducible and if $p \geq 1$ eigenvalues of E are of magnitude $\rho(E)$, they are distinct roots of polynomial equation $\lambda^p - [\rho(E)]^p = 0$ and there exists a permutation matrix T such that E can be expressed in the cyclic form*

$$F = TET^T = \begin{bmatrix} 0 & F_{12} & 0 & \cdots & 0 \\ 0 & 0 & F_{23} & \cdots & 0 \\ 0 & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & F_{(p-1)p} \\ F_{p1} & 0 & 0 & 0 & 0 \end{bmatrix}$$

where the diagonal zero blocks are square. Then, matrix E is called p-cyclic or p-periodic. The index of cyclicity or periodicity is p. ∎

The cyclicity period *p* of *E* is the greatest common divisor of the lengths of all the loops in the graph of *E*. Matrix *E* is acyclic or aperiodic if the cyclicity period is 1. i.e. if *E* is 1-cyclic, *p=1*.

If graph *A* is a cycle of length *p*, then $A^p = I$ the identity matrix.

2.8.2 Stochastic Matrices

Important in the study of graphs are the stochastic matrices. We say a matrix *E* is nonnegative, $E \succeq 0$, if all its elements are nonnegative. Matrix *E* is positive, $E \succ 0$, if all its elements are positive.

A matrix $E \succeq 0$ is row stochastic if all its row sums equal to 1. A matrix $E \succeq 0$ is doubly stochastic if all its row sums and column sums equal to 1. The product of two row stochastic matrices *E,F* is row stochastic because $EF\underline{1} = E\underline{1} = \underline{1}$.

The maximum eigenvalue of a stochastic matrix is 1. A matrix $E \succeq 0$ is row stochastic if and only if $\underline{1}$ is an eigenvector for the eigenvalue 1.

Let square $n \times n$ matrix $E \succeq 0$ have all row sums equal to a constant $c > 0$. Then [16]:

1. $\rho(E) = c$, and is an eigenvalue of *E* with eigenvector $\underline{1}$.

2. If $e_{ii} > 0$ for all *i*, then $|\lambda| < c$ for all eigenvalues $\lambda \neq c$.

Let E be the adjacency matrix of a graph G. Then:

3. $\lambda_1 = \rho(E) = c$ is simple if and only if E has a spanning tree. Then $rank(E) = n - 1$.

4. If E has a spanning tree and if $e_{ii} > 0$ for all i, then $\lambda_1 = \rho(E) = c$ is the unique eigenvalue of maximum magnitude.

2.8.3 M-Matrices

M-matrices are important in the analysis of graphs due to the fact that the graph Laplacian matrix $L = D - A$ is an M-matrix. This section reproduces some results from [13].

Given an $m \times n$ matrix A, a $k \times k$ minor of A is the determinant of a $k \times k$ matrix obtained by deleting m-k rows and n-k columns of A. Define a subset I of $\{1,...,m\}$ with k elements and a subset J of $\{1,...n\}$ with k elements. Write $[A]_{I,J}$ for the $k \times k$ minor of A corresponding to the rows with index in I and the columns with index in J. If $I=J$ then $[A]_{I,J}$ is called a principal minor. Those principal minors corresponding to matrices that are square upper left submatrices of A are called leading principal minors.

A square matrix $n \times n$ is an M-matrix if all its off diagonal elements are nonpositive and all its principal minors are nonnegative. It is a nonsingular M-matrix if all its principal minors are positive. Pictorially,

$$M = \begin{bmatrix} + & \leq 0 \\ \leq 0 & + \end{bmatrix}$$

Any M-matrix can be written as $E = sI - A$ for some $s > 0, A \succeq 0$.

The following properties of M-matrices from [13] are important for the analysis of graphs. Define a Z-matrix as one having nonpositive off-diagonal elements. A vector v is positive, $v > 0$, if all its elements are positive.

Theorem 2.9. Properties of nonsingular M-matrices. *Let $E \in Z$ be an $n \times n$ matrix. Then the following conditions are equivalent:*

1. *E is a nonsingular M-matrix*
2. *the leading principal minors of E are all positive.*
3. *The eigenvalues of E have positive real parts.*
4. *E^{-1} exists and has nonnegative elements.*

5. *There exist vectors* $w, v > 0$ *such that* $Ev, E^T w$ *are both positive.*

6. *There exists a positive diagonal matrix S such that* $ES + SE^T$ *is positive definite.*

Theorem 2.10. Properties of singular M-matrices. *Let* $E \in Z$ *be an* $n \times n$ *matrix. Then the following conditions are equivalent:*

1. *E is a singular M-matrix*
2. *the leading principal minors of E are all nonnegative.*
3. *the eigenvalues of E have nonnegative real parts.*
4. $(B + E)^{-1}$ *exists and is nonnegative for all diagonal matrices* B *having all diagonal elements positive.*
5. *There exist vectors* $w, v > 0$ *such that* $Ev, E^T w$ *are both nonnegative.*
6. *There exists a nonnegative diagonal matrix S such that* $ES + SE^T$ *is positive semidefinite.*

Theorem 2.11. Properties of irreducible M-matrices. *Let* $E = sI - A$ *be an* $n \times n$ *irreducible M-matrix, that is,* $A \succeq 0$ *and is irreducible. Then:*

1. *E has rank* $n - 1$
2. *there exists a vector* $v > 0$ *such that* $Ev = 0$.
3. $Ex \geq 0$ *implies* $Ex = 0$.
4. *Each principal submatrix of order less that n is a nonsingular M-matrix.*
5. $(B + E)^{-1}$ *exists and is nonnegative for all nonnegative diagonal matrices* B *with at least one positive element.*
6. *There exists a positive diagonal matrix P such that* $PE + E^T P$ *is positive semidefinite, that is, matrix -E is Lyapunov stable.*

The next results verify some other properties of irreducible M-matrices [21]. Though singular, it is very easy to turn irreducible matrices into nonsingular matrices by adding positive diagonal terms.

Corollary 2.3. *Let* $E \in Z$ *be a singular but irreducible M-matrix. Then* $\bar{E} = E + diag\{0, 0, \cdots, \varepsilon\}$ *is a nonsingular M-matrix for any* $\varepsilon > 0$. *Moreover,* $-\bar{E}$ *is stable.*

Corollary 2.4. L*et* $E \in Z$ *be a singular but irreducible M-matrix. Let* $\varepsilon_i \geq 0$ *and at least one of them be positive. Then* $\overline{E} = E + diag\{\varepsilon_i\}$ *is a nonsingular M-matrix. Moreover,* $-\overline{E}$ *is stable.*

2.9. Lyapunov Functions for Cooperative Control on Graphs

The properties of M-matrices are extremely important in constructing Lyapunov functions for analysis of dynamic systems on graphs. It is seen in this section that Lyapunov functions for cooperative control systems depend on the communication graph topology, specifically through the graph Laplacian matrix L. That is, stability analysis for cooperative control systems must take into account the connectivity properties of the graph, i.e., the way in which the individual systems communicate.

Recall that a system $\dot{x} = Fx$ is marginally stable if there exist symmetric matrices $P > 0, Q \geq 0$ such that

$$F^T P + PF = -Q \tag{2.120}$$

This equation is called a Lyapunov equation for system $\dot{x} = Fx$. Moreover, if this equation holds with $Q > 0$, then the system is asymptotically stable.

Recall that then,

$$V = x^T Px > 0 \tag{2.121}$$

is a Lyapunov function, since one has $\dot{V} = x^T(PF + F^T P)x \leq 0$ for marginal stability and $\dot{V} = x^T(PF + F^T P)x < 0$ for asymptotic stability.

Consider a graph with adjacency matrix A and Laplacian matrix $L = D - A$. If the graph is strongly connected, then A is irreducible and so is L. Then, L is an irreducible M-matrix. Then, According to Theorem 2.11 item 6, one has

$$PL + L^T P = Q \geq 0 \tag{2.122}$$

for a symmetric matrix $P > 0$. This is a Lyapunov equation that verifies the marginal stability of the consensus system

$$\dot{x} = -Lx \tag{2.123}$$

Let B be a diagonal matrix with non-negative diagonal entries and at least one of them positive. Then from Theorem 2.11 item 5 $L+B$ is a nonsingular M-matrix. Now according to Theorem 2.9 item 6 there exists a symmetric matrix $P > 0$ such that

$$P(L+B)+(L+B)^T P = Q > 0 \qquad (2.124)$$

This Lyapunov equation verifies the asymptotic stability of the system

$$\dot{x} = -(L+B)x . \qquad (2.125)$$

This system is of importance in subsequent chapters where we study cooperative tracker design.

Dependence of Lyapunov Functions on Graph Topology

The next results show specifically how to construct matrices *P* in Lyapunov function (2.121) that satisfy Lyapunov equations (2.122) and (2.124). Matrix *P* is seen to depend on graph Laplacian matrix *L*. Thus, these results tie the Lyapunov analysis of cooperative control systems directly to the graph topology. The first lemma constructs matrix *P* for the singular irreducible case in (2.122).

Lemma 2.3. Lyapunov Equation for Irreducible Laplacian Matrix [13]. *Let the graph be strongly connected so that the Laplacian matrix L is an irreducible singular M-matrix. Let $w_1 = \begin{bmatrix} p_1 & p_2 & \cdots & p_N \end{bmatrix}^T > 0$ be the left eigenvector of L associated with eigenvalue $\lambda_1 = 0$, i.e., $w_1^T L = 0$. Define*
$$P = diag\{p_i\}$$
Then $P > 0$ and $Q = PL+L^T P \geq 0$.

We present two methods for constructing a matrix *P* for Lyapunov function (2.121) that satisfies the Lyapunov equation (2.124). Both depend on the graph topology through the Laplacian matrix.

Lemma 2.4. Lyapunov Equation for Nonsingular Graph Matrix *L+B* [13]. *Let the graph be strongly connected so that the Laplacian matrix L is an irreducible singular M-matrix. Let B be a diagonal matrix with non-negative diagonal entries and at least one of them positive. Then $L+B$ is a nonsingular M-Matrix. Define*

$$q = [q_1, \cdots, q_N]^T = (L+B)^{-1} \underline{1}$$

$$P = diag\{p_i\} = diag\{1/q_i\}$$

Then $P > 0$ and $Q = P(L+B)+(L+B)^T P > 0$

Lemma 2.5. Lyapunov Equation for Nonsingular Graph Matrix $L+B$ [13].
Let the graph be strongly connected so that the Laplacian matrix L is an irreducible singular M-matrix. Let B be a diagonal matrix with non-negative diagonal entries and at least one of them positive. Then $L+B$ is a nonsingular M-Matrix. Let $p, q > 0$ be vectors such that

$$(L+B)q > 0, \quad p^T(L+B) > 0$$

Define

$$P = diag\{\mathrm{p}_i/q_i\}$$

Then $P > 0$ and $Q = P(L+B)+(L+B)^T P > 0$

Note that vectors $p, q > 0$ exist according to Theorem 2.9 item 5.

It is seen from these results that Lyapunov functions for studying the stability properties of cooperative control protocols depend on the graph topology. Therefore, a given cooperative control protocol may be stable on one graph topology but not on another. This reveals the close interactions between the performance of locally design control protocols and the manner in which the agents are allowed to communicate.

In subsequent chapters we shall rely heavily on the results of this section to analyze the stability of cooperative control systems on graphs.

2.10.   Conclusions and Setting for the Subsequent Chapters

This chapter has roughly followed the development of cooperative control results in the early literature since the first papers [22],[7],[10],[9],[16]. To understand the relationships between the communication graph topology and the local design of distributed feedback control protocols, it was natural to study first-order systems and then second-order systems. Early applications were made to formation control [18],[14],[16],[13]. These studies brought an understanding of the limitations and caveats imposed by the graph communication restrictions and opened the way for many well-known results from systems theory to be extended to the case of multi-agent systems on graphs.

The relations discovered between the communication graph topology and the design of distributed control protocols have resulted in new design techniques for cooperative feedback control. New intriguing interactions have been discovered between graph topology and local control protocol design that reveal the richness of the study of cooperative control on graphs, where new phenomena are seen that do not occur in control of single-agent systems.

It has been seen that Lyapunov functions for studying the stability properties of cooperative control protocols depend on the graph topology. In the remainder of the book we explore these relationships for optimal design and adaptive control on graphs.

To the Springer Editorial Team

The references in Chapters 2 through 10 need to be listed alphabetically as they are listed below.

Thank you.

Author Lewis

References

[1] J. Brewer, "Kronecker products and matrix calculus in system theory," IEEE Trans. Circuits and Systems, vol. 25, no. 9, pp. 772-781.

[2] W.L. Brogan, *Modern Control Theory*, 3rd edition, Prentice-Hall, NJ, 1990.

[3] R. Diestel, *Graph Theory,* Springer-Verlag, New York,2000.

[4] J.A. Fax and R. M. Murray. 2004, "Information Flow and Cooperative Control of Vehicle Formations," *IEEE Trans. Automatic Control* 49, no. 9: 1465-1476.

[5] C. Godsil and G.F. Royle, *Algebraic Graph Theory*, Springter-Verlag, New York, 2001.

[6] R.A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[7] A. Jadbabaie, J. Lin, and S. Morse. 2003. "Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules," *IEEE Trans. Automatic Control* 48, no. 6: 988-1001.

[8] D. Lee and M.W. Spong, "Stable flocking of multiple inertial agents on balanced graphs," IEEE Trans. Automatic Control, vol. 52, no. 8, pp. 1469-1475, 2007.

[9] L. Moreau, "Stability of multiagent systems with time-dependent communication links," IEEE Trans. Automatic Control, vol. 50, no. 2, pp. 169-182, Feb. 2005.

[10] R. Olfati-Saber and R.M. Murray, "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Transaction of Automatic Control*, vol. 49, no. 9, pp. 1520-1533, 2004.

[11] R. Olfati-Saber, J.A. Fax, and R.M. Murray, "Consensus and cooperation in networked multi-agent systems," Proc. IEEE, vol. 95, no. 1, pp. 215-233, Jan. 2007.

[12] A. Olshevsky and J.N. Tsitsiklis**, "**Convergence speed in distributed consensus and averaging", *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 33-55, 2009.

[13] Z. Qu, *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. London: Springer-Verlag, 2009.

[14] W. Ren, "Consensus strategies for cooperative control of vehicle formations," *IET Control Theory and Applications*, vo. 1, no. 2, pp. 505-512, 2007.

[15] W. Ren and R.W. Beard, Consensus Seeking in Multiagent Systems Under Dynamically Changing Interaction Topologies," *IEEE Trans. Automatic Control* vol. 50, no. 5, pp. 655-661, 2005.

[16] W. Ren and R.W. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control,* Springer-Verlag, London, 2008.

[17] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," *Proc. Amer. Control Conf.*, pp. 1859–1864, Portland, OR, 2005.

[18] W. Ren and E. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *Int. J. Robust and Nonlinear Control*, vol. 17, pp. 1002-1033, 2007.

[19] C. Reynolds, "Flocks, herds, and schools: A distributed behavior model," *Proceedings of SIGGRAPH 87,* 1987.

[20] Y. Shoham and K. Leyton-Brown, *Multiagent Systems*, Cambridge Press, New York, 2009.

[21] O. Taussky, "A recurring theorem on determinants," American Mathematical Monthly, vol. 56, no. 10, pp. 672-676, 1949.

[22] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, 1986.

[23] C.W. Wu, *Synchronization in Complex Networks of Nonlinear Dynamical Systems*, World Scientific, Singapore, 2007.

[24] G. Xie and L. Wang, "Consensus control for a class of networks of dynamic agents," *Int. J. Robust and Nonlinear Control*, vol. 17, pp. 941-959, 2007.

Part I
Local Optimal Design for Cooperative Control in Multi-agent Systems on Graphs

Cooperative control studies the dynamics of multi-agent dynamical systems linked to each other by a communication graph. The objective of cooperative control is to devise control protocols for the individual agents that guarantee synchronized behavior of the states of all the agents in some prescribed sense. In cooperative systems, any control protocol must be distributed in the sense that it respects the prescribed graph topology. That is, the control protocol for each agent is allowed to depend only on information about that agent and its neighbors in the graph.

In Part I of the book we will study local and global optimal control for cooperative multi-agent systems linked to each other by a communication graph. In Chapter 3 we study continuous-time systems, and we shall see that local optimal design at each agent guarantees global synchronization of all agents to the same state values on any suitably connected digraph. Chapter 4 considers discrete-time systems and shows that an extra condition relating the local agent dynamics and the graph topology must be satisfied to guarantee global synchronization using local optimal design.

In cooperative control systems on graphs it turns out that local optimality for each agent and global optimality for all the agents are not the same. The relations between stability and optimality are well understood for single-agent systems. However, there are more intriguing relations between stability and optimality in cooperative control than appear in the single-agent case, since local stability and global team stability are not the same, and local agent optimality and global team optimality are not the same. New phenomena appear that are not present for single-agent systems. Moreover, throughout everything the synchronization of the states of all agents must be guaranteed.

A common problem in optimal decentralized control is that global optimization problems generally require global information from all the agents, which is not available to distributed controllers. In Chapter 5 we shall see that globally optimal controls of distributed form may not exist on a given graph. To obtain globally optimal performance using distributed protocols that only depend on local agent information in the graph, the global performance index must be selected to depend on the graph properties in a certain way, specifically, through the graph Laplacian matrix.

In Chapter 6 we will define a different sort of global optimality for which distributed control solutions always exist on suitably connected graphs. There, we study multi-agent graphical games and show that if each agent optimizes its own local performance index, a Nash equilibrium is obtained.

3
Riccati Design for Synchronization of Continuous-Time Systems

This chapter studies cooperative tracking control of multi-agent dynamical systems interconnected by a fixed communication graph topology. Each agent or node is mathematically modeled by identical continuous linear time-invariant (LTI) systems, which includes the single integrator and double integrator as special cases. The communication network among the agents is described by a directed graph. A command generator or leader node generates the desired tracking trajectory to which all agents should synchronize. Only a few nodes are aware of information from the leader node. A locally optimal Riccati design approach is introduced here to synthesize the distributed cooperative control protocols. A framework for cooperative tracking control is proposed, including full state feedback control protocols, observer design, and dynamic output regulator control. The classical system theory notion of duality is extended to networked cooperative systems on graphs. It is shown that the local Riccati design method guarantees synchronization of multi-agent systems regardless of graph topology, as long as certain connectivity properties hold. This is formalized through the notion of synchronization region. It is shown that the Riccati design method yields unbounded synchronization regions and so achieves synchronization on arbitrary digraphs containing a spanning tree.

The results in this chapter were first published in [23].

3.1. Duality, Stability, and Optimality for Cooperative Control

In system theory, the relations between state variable feedback design, observer design, and output regulator design are well known and provide a beautiful overall theory that reveals deep structural relations for dynamical systems in terms of duality theory, the separation principle, and other well developed notions. In intercommunicating cooperative control systems, however, the design of feedback control protocols is complicated by the fact that the communication graph topology can severely restrict what can be accomplished by using distributed feedback control protocols, where the control policy of each agent is allowed to depend only on its own information and local information available from its neighbors in the graph. In this chapter we show that ideas of state feedback, observer design, and output regulator design can be extended to the case of cooperative control on

graphs in a natural by using locally optimal design in terms of local Riccati equations.

The relations between stability and optimality have long been debated in the community and are now for the most part understood for single-agent systems. However, relations of stability and optimality in multi-agent cooperative control systems are only now beginning to be clarified. There are more intriguing relations between stability and optimality in cooperative control than appear in the single-agent case, since local stability and global team stability are not the same, and local optimality and global team optimality are not the same.

In this chapter we discuss the design of cooperative control protocols that are distributed, in the sense that each agent is allowed to use only information from itself and its neighbors in the prescribed communication graph topology. The objective is for the states of all agents to synchronize to the state of a leader node, which can be viewed as a command generator exosystem that generates a desired trajectory. Only a few agents have access to information from the leader node. First we consider the case of cooperative control using full state variable feedback, then cooperative observer design in the case of output feedback or incomplete state measurements, then cooperative output regulator protocol design.

In Section 3.2 we assume that the full state of each agent is available for feedback control design by itself and its neighbors in the graph. Locally optimal design in terms of solving local Riccati equations is shown to guarantee synchronization on arbitrary graph topologies that have a spanning tree. The idea of synchronization region is given in Section 3.3. In Section 3.4 we design cooperative state observers based on output feedback, or reduced state information. A duality theory for cooperative control systems on graphs is given in Section 3.5, and it involves the reverse graph. In Section 3.6 we show three methods for designing cooperative dynamic regulators that guarantee synchronization using only output feedback. These three methods depend on different methods of exchanging information between neighbors in the graph.

3.2. State Feedback Design of Cooperative Control Protocols

In this section we design cooperative controllers using state variable feedback (SVFB), assuming that the full state of each agent is available for feedback control design by itself and its neighbors in the communication graph. We require that any control protocol be distributed in the sense that the control for agent i depends only on its own information and that from its neighbors in the graph. It is shown that locally optimal design in terms of a local Riccati equation, along with selection of a suitable distributed control protocol, guarantees synchronization on arbitrary communication graphs that have a spanning tree.

3.2.1 Synchronization of Multi-agent Systems on Graphs

Much attention was paid early on in the cooperative control literature to the leaderless consensus problem of single integrator and double integrator dynamics [10],[15],[17]. Under proper selection of a 'local neighborhood voting protocol', it was shown that if the graph has a spanning tree all nodes reach a consensus value that is a weighted average of the initial conditions of the agent dynamics. In this chapter, we shall study the more general case where each agent has dynamics of a continuous linear time-invariant (LTI) system. LTI systems form an important class of systems; they include the single integrator, double integrator, and higher-order integrator dynamics as special cases. A large class of engineering systems can be modeled by LTI systems, such as mechanical systems (e.g. mass-spring-damper systems), electrical systems (e.g. RLC circuits, op-amp circuits), and electromechanical systems (e.g. armature controlled direct current servomotor with a load) [1],[16].

We call the case where consensus is sought among agents when there is no leader node the *cooperative regulator* problem. Here, we consider the *cooperative tracking* problem, where there are N identical follower agents or nodes and one leader node. Each follower node i has identical dynamics and is modeled by an LTI system

$$\dot{x}_i = Ax_i + Bu_i, \quad y_i = Cx_i, \quad i \in \mathcal{N} \tag{3.1}$$

where $x_i \in \mathbb{R}^n$ is the state, $u_i \in \mathbb{R}^m$ is the input, $y_i \in \mathbb{R}^p$ is the measured output, and $\mathcal{N} = \{1, 2, \ldots, N\}$. The triple (*A, B, C*) is assumed to be stabilizable and detectable. These follower nodes can communicate in a certain way, and the communication network is represented by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with N nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, and a set of edges or arcs $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Let the associated adjacency matrix be $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$, where $a_{ij} \geq 0$ is the weight for edge (v_j, v_i). If there is an edge from node j to node i (i.e. $(v_j, v_i) \in \mathcal{E}$), $a_{ij} > 0$. This means that node j is a neighbor of node i, and node i can get information from node j. The neighbor set of node i is denoted as $N_i = \{j \mid a_{ij} > 0\}$. We assume there is no self loop, i.e., $a_{ii} = 0$.

The dynamics of a leader or control node, labeled *0*, is given by

$$\dot{x}_0 = Ax_0, \quad y_0 = Cx_0, \tag{3.2}$$

where $x_0 \in \mathbb{R}^n$ is the state and $y_0 \in \mathbb{R}^p$ is the measured output. The leader node can be considered as a command generator exosystem that generates the desired target trajectory. It is an autonomous system and is not affected by any follower node.

The objective of the cooperative tracking problem is to design local distributed controllers u_i for all follower nodes i ($\forall i \in \mathcal{N}$), such that all follower nodes track the state trajectory of the leader node, i.e., $\lim_{t\to\infty}(x_i(t)-x_0(t))=0$, $\forall i \in \mathcal{N}$. Then we say that the states of all agents synchronize to the state of the command generator.

Remark 3.1. Though the leader dynamics (3.2) is autonomous and has no control input, it can generate a large class of useful command trajectories. The interesting cases are when matrix A is not stable. The types of trajectories that can be generated by the leader include unit step (position command), the ramp (useful in velocity tracking systems, e.g., satellite antenna pointing), sinusoidal waveforms (useful, e.g., in hard disk drive control), and more. In this chapter, A can be either stable, marginally stable, or even unstable, as long as (A,B) is stabilizable.

Define the augmented graph as $\bar{\mathcal{G}} = \{\bar{\mathcal{V}},\bar{\mathcal{E}}\}$, where $\bar{\mathcal{V}} = \{v_0, v_1,\ldots,v_N\}$ is the node set, including the leader node and all follower nodes, and $\bar{\mathcal{E}} \subseteq \bar{\mathcal{V}} \times \bar{\mathcal{V}}$ is the edge set. Note that $\mathcal{E} \subset \bar{\mathcal{E}}$. For the cooperative tracking problem, we need the following assumption on the graph topology.

Assumption 3.1. *The augmented graph $\bar{\mathcal{G}}$ contains a spanning tree with the root node being the leader node. In other words, there is a directed path (not necessarily unique) from the leader node to every follower node.*

Assumption 3.1 is a necessary condition for solving the cooperative tracking problem. This is illustrated in the following remark in an intuitive way.

Remark 3.2. Suppose Assumption 3.1 does not hold and there is a set of k ($1 \le k < N$) nodes in graph $\mathcal{G}$, i.e. $S_k = \{n_1, n_2,\ldots,n_k\}$ and $S_k \subset \mathcal{N}$, which do not have directed paths from the leader node 0. This further implies that the set of nodes S_k do not have access to information of the rest of the nodes $\bar{S}_k = \{0,\mathcal{N}\} \setminus S_k$. It is obvious that there must exist some node (nodes) in the set S_k , which is (are) either isolated (i.e. isolated single node or isolated subgroup), or has an edge (have edges) to at least one node in $\bar{S}_k$. These cases are depicted in Fig. 3.1. Obviously, in these cases, synchronization to the leader node cannot be achieved.
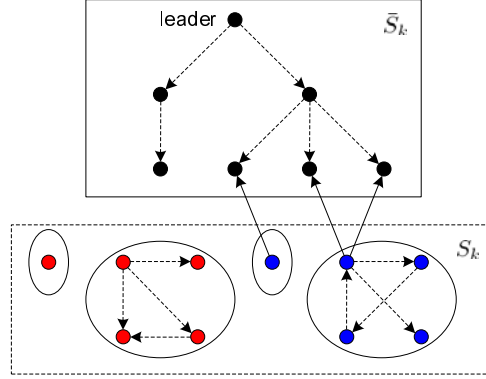
Fig. 3.1. Graph topologies discussed in Remark 3.2

Remark 3.3. Assumption 3.1 includes the following classes of graph topologies (see Fig. 3.2) as special cases.

(a) Graph $\mathcal{G}$ is strongly connected and at least one follower node in graph $\mathcal{G}$ can get information from the leader node [4],[5].

(b) Graph $\mathcal{G}$ contains a spanning tree and at least the root node can get access to the leader node [14],[23].

(c) The augmented graph $\bar{\mathcal{G}}$ is itself a spanning tree with the root being the leader node [3].

(d) The augmented graph $\bar{\mathcal{G}}$ has a hierarchical structure [6].

(e) The augmented graph $\bar{\mathcal{G}}$ has a spanning tree, but graph $\mathcal{G}$ is disconnected and each separated subgroup either is a single node or contains a spanning tree.

The statement that a graph *is* a spanning tree (e.g. Fig. 3.2 (c)) is different from the statement that a graph *has* (or *contains*) a spanning tree (e.g. Fig. 3.2 (a)-(e)). The former is a special case of the latter. A graph is a spanning tree if it has a spanning tree, and each node has exactly one parent node, except for the root node which does not have parent node.
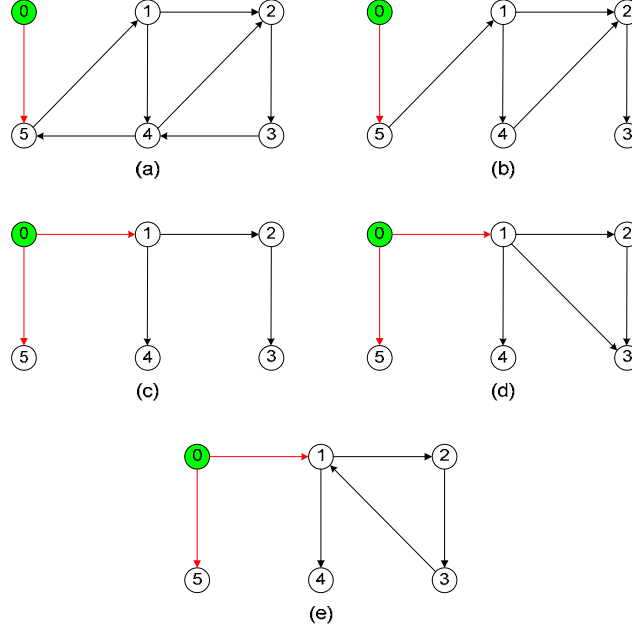
Fig. 3.2. Five classes of graph topologies described in Remark 3.3

3.2.2 Cooperative State Variable Feedback Control

In this Section 3.2, we assume that full state variable of each node is measurable, and design a cooperative state variable feedback (SVFB) control law of distributed form using a Riccati design approach [21], [23]. This is the case $C=I$ in (3.1). The case when $C \neq I$ and the measured output only contains reduced state information is addressed in Sections 3.4, 3.5, 3.6.

We assume that the leader node can only be observed by a small subset of the follower nodes. This is often the case in practice when follower nodes are sparsely populated and limited sensing ability is taken into account. If node i observes the leader, an edge (v_0, v_i) is said to exist with a weight $g_i > 0$. The weights g_i have been called pinning gains in [7],[14],[22] and if $g_i > 0$ then node i is said to be pinned to the leader.

To make the controller fully distributed, the control law of each agent must respect the graph topology and can only use the local neighborhood information of that agent. Define the *local neighborhood tracking error* of node i as [11]

$$\varepsilon_i = \sum_{j=1}^{N} a_{ij}(x_j - x_i) + g_i(x_0 - x_i). \tag{3.3}$$

This can also be written in the form

$$\varepsilon_i = -(d_i + g_i)\left(x_i - \frac{\sum_{j=1}^{N} a_{ij}x_j + g_i x_0}{d_i + g_i}\right) \tag{3.4}$$

where $d_i = \sum_{j=1}^{N} a_{ij}$ is the in-degree of node i. While (3.3) can be interpreted as a weighted sum of differences between the states of node i and its neighbors, (3.4) describes the difference between the state of node i and the weighted average center of its neighbors' states.

Consider a static SVFB control protocol for each node $i \in \mathcal{N}$ given as

$$u_i = cK\varepsilon_i \tag{3.5}$$

with coupling gain $c > 0$ and feedback gain matrix $K \in \mathbb{R}^{m \times n}$. These controllers are distributed in the sense that they are implemented at each node using only the local neighborhood tracking error information ε_i. The task is to find suitable c and K such that all agents synchronize to the leader node for the given graph topology.

The closed-loop system of node i is

$$\dot{x}_i = Ax_i + cBK\left(\sum_{j=1}^{N} a_{ij}(x_j - x_i) + g_i(x_0 - x_i)\right).$$

For ease of analysis, write the overall global closed-loop dynamics as

$$\dot{x} = \left(I_N \otimes A - c(L+G) \otimes BK\right)x + (c(L+G) \otimes BK)\underline{x}_0,$$

where the global state $x = col(x_1, x_2, \ldots, x_N) \in \mathbb{R}^{nN}$ is the columnwise concatenation of local state vectors $x_1, \ldots, x_N$, and $\underline{x}_0 = col(x_0, x_0, \ldots, x_0) \in \mathbb{R}^{nN}$. The Kronecker product is $\otimes$. The identity matrix is $I_N \in \mathbb{R}^{N \times N}$ and $G = diag(g_1, g_2, \ldots, g_N)$ is the diagonal matrix of pinning gains which describes the connections between the leader node and follower nodes. The Laplacian matrix associated with graph $\mathcal{G}$ is defined by $L = [l_{ij}] = D - \mathcal{A}$, with $D = diag(d_1, d_2, \ldots, d_N)$ the diagonal matrix of in-degrees. It is apparent that

$$l_{ij} = \begin{cases} -a_{ij}, & j \neq i, \\ \sum_{k=1}^{N} a_{ik}, & j = i. \end{cases}$$

Define the global system and control input matrices

$$A_c = I_N \otimes A - c(L+G) \otimes BK,$$
$$B_c = c(L+G) \otimes BK.$$

Matrix A_c reflects the local agent closed-loop matrix $A - BK$ as modified on the graph structure $L+G$. Note that the graph structure $L+G$ appears on the left-hand side of the Kronecker product, while the local state information appears on the right.

Denote the disagreement error with respect to the leader for each node as $\delta_i = x_i - x_0$. Define the global disagreement vector [15] as

$$\delta = col(\delta_1, \delta_2, \ldots, \delta_N) = x - \underline{x}_0 \in \mathbb{R}^{nN}, \tag{3.6}$$

Then the cooperative tracking problem is solved if $\lim_{t\to\infty} \delta(t) = 0$. The global disagreement error dynamics is given by

$$\dot{\delta} = \dot{x} - \underline{\dot{x}}_0 = A_c \delta. \tag{3.7}$$

Note that the disagreement error $\delta_i = x_i - x_0$ is not available to node i unless it is pinned to the leader node (that is, $g_i \neq 0$), whereas the local neighborhood tracking error (3.3) is known to each node i.

Before moving on, we need the following lemmas. In these lemmas, the notations A and a_{ij} stand for a general matrix A and its entries, and are not related to the system matrix A or the graph adjacency matrix entries a_{ij}.

Lemma 3.1. Geršgorin Disc Theorem *[9]. Let a matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$. Then all eigenvalues of A reside in the union of the n discs*

$$\bigcup_{i=1}^{n} \left\{ \lambda \in C \middle| |\lambda - a_{ii}| \leq \sum_{j=1; j\neq i}^{n} |a_{ij}| \right\}.$$

Lemma 3.2. *[20] Let a matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ and*

$$J = \left\{ i \in \{1, 2, \ldots, n\} \middle| |a_{ii}| > \sum_{j=1; j \neq i}^{n} |a_{ij}| \right\} \neq \varnothing.$$

If for each $i \notin J$, there is a sequence of nonzero elements of A of the form $a_{ii_1}, a_{i_1i_2}, \ldots, a_{i_lj}$ with $j \in J$. Then A is nonsingular.

Lemma 3.3. *Under Assumption 3.1, the matrix $L+G$ is nonsingular. Moreover, all its eigenvalues are located in the open right-half plane.*

Proof. Under Assumption 3.1, at least one node in $\mathcal{G}$ can get information directly from the leader node, i.e., $g_i > 0$ for at least one $i \in \mathcal{N}$. Without loss of generality, we assume there are two nodes r_1 and r_2 such that $g_{r_1} > 0$ and $g_{r_2} > 0$. Since $\sum_{j=1}^{N} l_{ij} = 0$, $l_{ii} \geq 0$ and $l_{ij} \leq 0$, $|l_{ii}| = \sum_{j=1; j \neq i}^{N} |l_{ij}|$. Then for matrix $L+G$, $|l_{ii} + g_i| \geq \sum_{j=1; j \neq i}^{N} |l_{ij}|$ for all $i \in \mathcal{N}$ and strict inequality holds for $i \in \{r_1, r_2\}$. Assumption 3.1 implies that, for any other node i which does not have direct access to the leader node (i.e., $i \in \mathcal{N} \setminus \{r_1, r_2\}$), there must be a direct path either originated from node r_1 or node r_2. According to Lemma 3.2, $L+G$ is nonsingular. By Geršgorin disc theorem, all eigenvalues of $L+G$ lie within the union $\{\lambda \in \mathbb{C} \,|\, Re(\lambda) > 0\} \cup \{0\}$. The fact that $L+G$ is nonsingular further implies all its eigenvalues are located in the open right-half plane.

The next result (cf. [8]) provides a necessary and sufficient condition for asymptotic stability of the disagreement error dynamics (3.7).

Lemma 3.4. *Let λ_i ($i \in \mathcal{N}$) be the eigenvalues of $L+G$, which may or may not be distinct. Then system (3.7) is asymptotically stable if and only if all the matrices*

$$A - c\lambda_i BK, \quad i = 1, 2, \ldots, N \tag{3.8}$$

are asymptotically stable.

Proof. The zero state of system (3.7) is asymptotically stable if and only if A_c is Hurwitz, i.e., all eigenvalues of A_c lie in the open left-half plane. There exists a nonsingular matrix $S \in \mathbb{R}^{N \times N}$, such that

$$S^{-1}(L+G)S = J = \begin{bmatrix} J_{n_1}(\bar{\lambda}_1) & & & \\ & J_{n_2}(\bar{\lambda}_2) & & \\ & & \ddots & \\ & & & J_{n_k}(\bar{\lambda}_k) \end{bmatrix},$$

where $n_1 + n_2 + \ldots + n_k = N$ and $J_{n_1}, J_{n_2}, \ldots, J_{n_k}$ are Jordan blocks of sizes $n_1, n_2, \ldots$. Note that the eigenvalues $\bar{\lambda}_i$ of $L+G$ need not be distinct. Similarity transformation of matrix A_c gives a block triangular matrix

$$
\begin{aligned}
\bar{A}_c &= (S \otimes I_n)^{-1} A_c (S \otimes I_n) \\
&= (S \otimes I_n)^{-1} (I_N \otimes A - c(L+G) \otimes BK)(S \otimes I_n) \\
&= I_N \otimes A - cJ \otimes BK.
\end{aligned}
\tag{3.9}
$$

Denote the diagonal entries of J as $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$, i.e. $[\lambda_1, \lambda_2, \ldots, \lambda_N] = [\underbrace{\bar{\lambda}_1, \ldots, \bar{\lambda}_1}_{n_1}, \ldots, \underbrace{\bar{\lambda}_k, \ldots, \bar{\lambda}_k}_{n_k}]$. Since the eigenvalues of a block triangular matrix are the union of the sets of eigenvalues of the diagonal blocks, $\bar{A}_c$ is Hurwitz if and only if all the diagonal entries $A - c\lambda_i BK$ ($i = 1, 2, \ldots, N$) are Hurwitz. Hence under this condition A_c is Hurwitz. This completes the proof. □

If matrix A is stable, then even with a coupling gain of $c=0$ all nodes synchronize to the zero state. If the command generator matrix A is not stable (see Remark 3.1), then it is inferred from Lemma 3.4 that all eigenvalues λ_i of $L+G$ must be nonzero. By Lemma 3.3, if the graph has a spanning tree with the leader as the root node, then all eigenvalues λ_i of $L+G$ have positive real parts. Then, proper design of c, K to satisfy condition (3.8) guarantees synchronization.

Lemma 3.4 implies that an arbitrary SVFB control gain K that stabilizes the single-agent dynamics $A - BK$ may fail to stabilize the global dynamics A_c for a prescribed graph structure. That is, though individual agents may have stable dynamics in isolation, when they are connected together by a communication graph, their stability may be destroyed. This is illustrated in Example 3.1 in Section 3.3. The condition (3.8) is difficult to guarantee and has been a stumbling block to the design of distributed control protocols that guarantee synchronization on a given graph topology.

3.2.3 Local Riccati Design of Synchronizing Protocols

Our objective now is to overcome the coupling between the local agent feedback control design and the graph topological properties that is displayed in Lemma 3.4, and to provide a design method for feedback matrix K in protocol (3.5) that is independent of the graph topology. That is, we wish to decouple the design of the control protocol from the graph properties, and show how to design a protocol that guarantees synchronization on any directed graph with suitable properties.

The next result shows how to select SVFB control gain K to guarantee stability on arbitrary digraphs satisfying Assumption 3.1 by using a local Riccati design approach [13] and proper choice of the coupling gain c.

Theorem 3.1. *Consider local distributed control protocols* (3.5). *Suppose (A,B) is stabilizable and let design matrices $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ be positive definite. Design the SVFB control gain K as*

$$K = R^{-1}B^T P, \tag{3.10}$$

where P is the unique positive definite solution of the control algebraic Riccati equation (ARE)

$$0 = A^T P + PA + Q - PBR^{-1}B^T P. \tag{3.11}$$

Then under Assumption 3.1, the global disagreement error dynamics (3.7) *is asymptotically stable if the coupling gain satisfies*

$$c \geq \frac{1}{2\min_{i \in \mathcal{N}} \operatorname{Re}(\lambda_i)} \tag{3.12}$$

with λ_i ($i \in \mathcal{N}$) the eigenvalues of $L + G$. Then, all agents synchronize to the leader node state trajectory.

Proof. Let the eigenvalues of $L + G$ be $\lambda_i = \alpha_i + j\beta_i$, where $\alpha_i, \beta_i \in \mathbb{R}$. Then by Lemma 3.3, $\alpha_i > 0$, $\forall i \in \mathcal{N}$. Considering (3.10) and (3.11), straightforward computation gives the Lyapunov equation

$$(A - c\lambda_i BK)^* P + P(A - c\lambda_i BK) = -Q - (2c\alpha_i - 1)K^T RK,$$

where the superscript $*$ denotes conjugate transpose of a complex matrix. Since $P > 0$ and $Q > 0$, by Lyapunov theory [2] matrix $A - c\lambda_i BK$ is Hurwitz if $c \geq 1/2\alpha_i$, $\forall i \in \mathcal{N}$. Then Lemma 3.4 completes the proof.

The importance of this result is that it decouples the local SVFB gain design at each agent from the details of the interconnecting communication graph structure. The SVFB gain is designed using the control ARE (3.11), and then the graph topology comes into the choice of the coupling gain c through condition (3.12).

Note that the SVFB gain in Theorem 3.1 is the optimal gain [13] that minimizes the local performance index

$$J_i = \frac{1}{2}\int_0^\infty x_i^T Q x_i + u_i^T R u_i \ dt \tag{3.13}$$

subject to the local agent dynamics (3.1). Therefore, the theorem shows that locally optimal SVFB control design by each agent guarantees the synchronization of all agents on any graph that has a spanning tree with the leader as the root node.

The locally optimal design method in Theorem 3.1 is scalable, since it only depends on the local agent dynamics, which have state space of dimension n, and does not depend on the size N of the graph.

It is stressed that the locally optimal design that minimizes the local agent performance index (3.13) does not guarantee the global optimality of the motion of all the agents as a team. Globally optimal design for multi-agent systems is discussed in Chapter 5.

3.3. Region of Synchronization

From Lemma 3.4, one sees that without using Riccati design, one can still find suitable coupling gain c and control gain K such that asymptotic tracking can be achieved by determining a SVFB gain that stabilizes (3.8) for all eigenvalues λ_i of $L+G$. The Riccati design in Theorem 3.1 decouples the design of the local agent feedback protocols from the graph properties and so provides a simple way of finding such a gain. This section shows another point of view of this issue by describing the concept of synchronization region [7],[14],[22]. This concept provides a natural way to evaluate the performance of synchronization control laws by revealing how synchronizability depends on structural parameters of the communication graph.

Definition 3.1. Synchronization Region. *Given system (A,B) and feedback matrix K, the synchronization region is the region in the complex plane defined as* $S = \{s \in \mathbb{C} \mid A - sBK \text{ is Hurwitz}\}$.

Our application of this definition is in regards to the distributed control protocol (3.5). By Lemma 3.4 and Definition 3.1, synchronization is achieved if $c\lambda_i$ fall into the synchronization region S for all $i \in \mathcal{N}$. Clearly, the synchronization region depends on both the control matrix K and the coupling gain c. It is shown in [7] that, for unsuitable choices of K the synchronization region may be disconnected. A synchronization region that forms a connected unbounded right-half plane is a desirable property of a consensus protocol [14], since, according to Lemma 3.3, such a region contains all possible graph eigenvalues for appropriate choice of coupling gain c. By Lemma 3.4, an unbounded synchronization region

implies that the same control gain K guarantees synchronization for any digraph containing a spanning tree.

Corollary 3.1. *For protocol (3.5) with the Riccati design-based control gain (3.10), the synchronization region is unbounded. More specifically, a conservative estimate for the synchronization region is $S = \{\alpha + j\beta \mid \alpha \in [1/2, \infty), \beta \in (-\infty, \infty)\}$.*

Proof. Let $s = \alpha + j\beta$. Using the same development as in Theorem 3.1, we have

$$(A - sBK)^* P + P(A - sBK) = -Q - (2\alpha - 1)K^T RK.$$

Since $P > 0$, $Q > 0$ and $R > 0$, $A - sBK$ is Hurwitz if and only if $-Q - (2\alpha - 1)K^T RK < 0$. A sufficient condition is $\alpha \geq 1/2$. This completes the proof.

The following example shows that a random stabilizing control gain K may yield a bounded synchronization region, while the Riccati design-based control gain (3.10) renders an unbounded synchronization region.

Example 3.1. Consider the agent dynamics (3.1) with matrices [14]

$$A = \begin{bmatrix} -2 & -1 \\ 2 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The feedback gain $K = [0.5, 0.5]$ stabilizes A-BK. The synchronization region for this gain matrix is $S_1 = \left\{ x + jy \,\middle|\, x < 2; \frac{x}{2}\left(1 - \frac{x}{2}\right)^2 - \left(1 - \frac{x}{8}\right)y^2 > 0 \right\}$, which is shadowed in Fig. 3.3.

If, for a given graph, the eigenvalues of $L+G$ fall within the region S_1, synchronization occurs. However, this is a severe restriction on the allowable forms of communication between the agents. Note in particular that if the communication graph is a cycle, the eigenvalues of L are uniformly distributed about a circle in the right-half plane (Chapter 2), and so would in likelihood not fall within this synchronization region.

Now consider the Riccati design-based feedback gain $K = [1.544, 1.8901]$ provided by Theorem 3.1 with $Q = I_2$ and $R = 1$. The synchronization region is $S_2 = \left\{ x + jy \,\middle|\, 1 + 1.544x > 0; (5.331x - 1.5473)y^2 + 2.2362(1 + 1.54x)^2 x > 0 \right\}$, which is unbounded as shadowed in Fig. 3.4. Lemma 4 in [14] is used in computing the synchronization region. Note that the region indicated in Fig. 3.4 is the actual synchronization region determined from solving the inequality given. The region

given by Corollary 3.1 is the region $\{x+jy \mid x \in [1/2, \infty),\ y \in (-\infty, \infty)\}$, which is a conservative estimate of S_2.
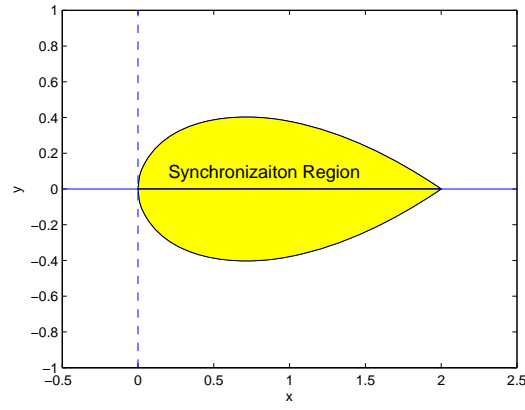
Fig. 3.3. Bounded synchronization region for a random stabilizing SVFB gain
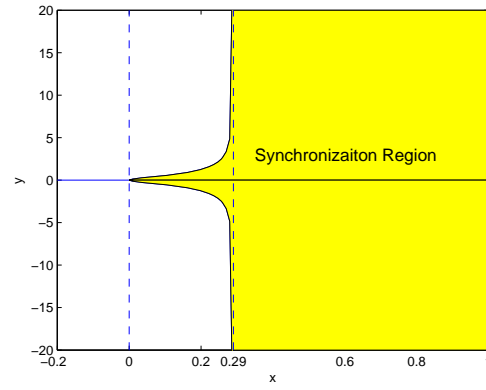
Fig. 3.4. Unbounded synchronization region for Riccati-based SVFB gain

3.4. Cooperative Observer Design

In Section 3.2, a cooperative tracking controller is designed by assuming that the full state information of all nodes can be measured. Unfortunately, nature seldom cooperates so eagerly. In practice, only limited output information can be measured. In this case, for single-agent systems it is well understood how to design a dynamical output regulator that stabilizes a system or enforces a desired tracking behavior. Some of these regulator designs are based on state observers [12], which take the available output measurements and provide estimates of the full system state. In this section, we investigate cooperative observer design for networked multi-agent systems, where the graph topology interacts with the local agent observer design [21],[23]. In Section 3.6 we show how to use the state feedback protocols in Section 3.2 along with these cooperative observers to design dynamic regulators that guarantee synchronization using only the available output measurements.

Consider system (3.1) for the i-th agent, that is

$$\dot{x}_i = Ax_i + Bu_i, \quad y_i = Cx_i, \quad i \in \mathcal{N} \tag{3.14}$$

Denote $\hat{x}_i \in \mathbb{R}^n$ as the estimate of the state x_i, $\hat{y}_i = C\hat{x}_i$ as the consequent estimate of the output y_i. Define the state estimation error $\tilde{x}_i = x_i - \hat{x}_i$ and the output estimation error $\tilde{y}_i = y_i - \hat{y}_i$ for node i.

Standard observer design for a single system often uses the output estimation error information $\tilde{y} = y - \hat{y}$. Similarly, for cooperative observer design, taking the graph topology into account, we define the *local neighborhood output estimation error* for node i as

$$\zeta_i = \sum_{j=1}^{N} a_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(\tilde{y}_0 - \tilde{y}_i) \tag{3.15}$$

which depends on the output estimation errors of node i and its neighbors. The cooperative observer for each node i ($i \in \mathcal{N}$) is designed in the form

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\zeta_i. \tag{3.16}$$

where $c > 0$ is the coupling gain, and $F \in \mathbb{R}^{n \times p}$ is the observer gain. These observers are completely distributed in the sense that each observer only requires its own output estimation error information and that of its neighbors.

The cooperative observer design problem is to select the coupling gain c and the observer gain F so that the state estimation error $\tilde{x}_i = x_i - \hat{x}_i$ of all agents go to zero.

Since the leader node *0* acts as a command generator, it is reasonable to assume that the leader node knows its own state, i.e., $\hat{x}_0 = x_0$ and $\tilde{y}_0 = 0$. Then the global cooperative observer dynamics is

$$\dot{\hat{x}} = A_0 \hat{x} + (I_N \otimes B)u + c[(L+G) \otimes F]y,$$

where $\hat{x} = col(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N)$, $y = col(y_1, y_2, \ldots, y_N)$, $u = col(u_1, u_2, \ldots, u_N)$ and

$$A_0 = I_N \otimes A - c(L+G) \otimes FC.$$

Matrix A_0 reflects the local agent observer matrix $A - FC$ as modified by the graph structure $L+G$. Note that graph topology information appears on the left-hand side of the Kronecker product while local agent information appears on the right.

Let the global state estimation error be

$$\tilde{x} = x - \hat{x} \tag{3.17}$$

Then straightforward computation gives the dynamics of the state estimation error

$$\dot{\tilde{x}} = A_0 \tilde{x} \tag{3.18}$$

Now the task for cooperative observer design is to find a suitable coupling gain c and observer gain F, such that the matrix A_0 is Hurwitz. Then the state estimation error goes to zero. The next result is dual to Lemma 3.4 and provides a necessary and sufficient condition for asymptotical stability of the global cooperative observer dynamics.

Lemma 3.5. *Let λ_i ($i \in \mathcal{N}$) be the eigenvalues of $(L+G)$. Then the state estimation error dynamics* (3.18) *is asymptotically stable if and only if all the matrices*

$$A - c\lambda_i FC, \quad \forall i \in \mathcal{N}, \tag{3.19}$$

are Hurwitz, i.e., asymptotically stable.
Proof. See the proof of Lemma 3.4.

If the command generator matrix *A* is not stable (see Remark 3.1), then it is inferred from Lemma 3.5 that all eigenvalues λ_i of *L+G* must be nonzero. By Lemma 3.3, if the graph has a spanning tree with the leader as the root node, then all eigenvalues λ_i of *L+G* have positive real parts. Then, proper design of *c, F* to

satisfy condition (3.19) guarantees that all state estimation errors converge to zero using the cooperative observers (3.16).

Lemma 3.5 shows how the graph topology interferes with the design of the local agent observers. This is undesirable. We now wish to devise a design method that decouples the local agent observer design from the graph properties.

Similar to Theorem 3.1, the next result shows how to select the observer gain F to guarantee that (3.19) is satisfied on arbitrary digraphs satisfying Assumption 3.1 by using a local Riccati design approach [12] and proper choice of the coupling gain c.

Theorem 3.2. *Suppose (A,C) is detectable and let design matrices $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{p \times p}$ be positive definite. Design the observer gain F as*

$$F = PC^T R^{-1} \tag{3.20}$$

where P is the unique positive definite solution of the observer ARE

$$0 = AP + PA^T + Q - PC^T R^{-1} CP . \tag{3.21}$$

Then the estimation error dynamics (3.18) *are asymptotically stable if the coupling gain satisfies*

$$c \geq \frac{1}{2 \min_{i \in \mathcal{N}} \mathrm{Re}(\lambda_i)} \tag{3.22}$$

with λ_i ($i \in \mathcal{N}$) be the eigenvalues of $(L+G)$.

Proof. The development is the same as in Theorem 3.1. Except that here we have

$$(A - c\lambda_i FC)P + P(A - c\lambda_i FC)^* = -Q - (2c\alpha_i - 1)FRF^T$$

The importance of this theorem is that it decouples the local observer gain design at each agent from the details of the interconnecting communication graph structure. The observer gain is designed using the observer ARE (3.21) [12], and then the graph topology comes into the choice of the coupling gain c through condition (3.22).

The local Riccati design method in Theorem 3.2 is scalable, since it only depends on the local agent dynamics, which have state space of dimension n, and does not depend on the size N of the graph.

3.5. Duality for Cooperative Systems on Graphs

As is well known in classical control theory, controller design and observer design are dual problems [2]. One outcome of duality is that these two problems are equivalent to the same algebraic problem. Formally, given system (3.1), (3.14), which we denote as *(A,B,C)*, the dual system is defined to be

$$\dot{x}_i = A^T x_i + C^T u_i, \quad y_i = B^T x_i \qquad (3.23)$$

which we denote as (A^T, C^T, B^T). A standard result is that system (3.14) is reachable (which depends on the properties of the input-coupling pair *(A,B)*) if and only if system (3.23) is detectable (which depends on the properties of the output-coupling pair (A^T, B^T). Moreover, replacing *(A,B)* in (3.11) by (A^T, C^T) results in (3.21), so that the control ARE (3.11) is dual to the observer ARE (3.21). Therefore, the observer gain *F* in Theorem 3.2 can be found by computing the SVFB gain *K* in Theorem 3.1, with *(A,B)* replaced by (A^T, C^T) and then setting $F = K^T$ [12].

Another remarkable insight provided by duality theory for standard single systems is that control design is dual to observer design and estimation if the time is reversed. Thus, optimal control is fundamentally a backwards-in-time problem [13], as captured by the idea of dynamic programming, which proceeds backwards from a desired goal state. On the other hand, optimal estimation is fundamentally a forwards-in-time problem [12]. A complete theory of duality is provided in the theory of electric circuits, where such notions are formally captured.

Unfortunately, in inter-communicating cooperative multi-agent systems, the design of cooperative feedback control protocols and cooperative observers is complicated by the fact that the communication graph topology can severely restrict what can be accomplished by using distributed protocols, where the policy of each agent is allowed to depend only on its own information and local information available from its neighbors in the graph. Therefore, the theory of duality has not been fully developed for cooperative control systems.

In this section, we extend the important concept of duality to networked cooperative linear systems. We show that cooperative SVFB controller design and cooperative observer design are also dual problems under suitable definitions. The next definition captures the notion of time reversal in cooperative systems on graphs.

Definition 3.2. Reverse Digraph. *Given a digraph $\mathcal{G}$, the reverse digraph $\mathcal{G}'$ is the graph with the same nodes as digraph $\mathcal{G}$, and with the directions of all edges reversed.*

Thus, if the adjacency matrix of digraph $\mathcal{G}$ is $\mathcal{A} = [a_{ij}]$, then the adjacency matrix of its reverse digraph $\mathcal{G}'$ is $\mathcal{A}^T = [a_{ji}]$. This means that the row sums in the reverse graph correspond to column sums in the original graph, so that the roles of in-neighbors and out-neighbors are exchanged.

Lemma 3.6. *If the digraph $\mathcal{G}$ is balanced, then the transpose of its Laplacian matrix L is the same as the Laplacian matrix L' of its reverse digraph $\mathcal{G}'$, i.e. $L^T = L'$.*

Theorem 3.3. Duality of cooperative systems on graphs. *Consider a networked system of N identical linear dynamics (A, B, C) on a balanced communication graph $\mathcal{G}$. Suppose the SVFB gain K stabilizes the synchronization error dynamics (3.7). Then the observer gain K^T stabilizes the state estimation error dynamics (3.18) for a networked dual system of N identical linear dynamics (A^T, C^T, B^T) on the reverse graph $\mathcal{G}'$.*

Proof. Consider the networked system with (A, B, C) and graph $\mathcal{G}$. Under SVFB gain K, the synchronization error dynamics (3.7) is

$$\dot{\delta} = [I_N \otimes A - c(L+G) \otimes BK]\delta = A_c \delta. \tag{3.24}$$

For the networked dual system with (A^T, C^T, B^T) and reverse graph $\mathcal{G}'$, under the observer gain K^T, the state estimation error dynamics (3.18) is

$$\dot{\tilde{x}} = [(I_N \otimes A^T) - c(L'+G) \otimes (K^T B^T)]\tilde{x} = A_o^{\mathcal{G}'} \tilde{x}.$$

Since $G = diag(g_i)$, it is follows that $A_0^{\mathcal{G}'} = (A_c)^T$. This proves the duality.

The equivalence between these dual problems is summarized as follows

$$A \leftrightarrow A^T$$
$$B \leftrightarrow C^T$$
$$C \leftrightarrow B^T$$
$$\mathcal{G} \leftrightarrow \mathcal{G}'$$
$$L \leftrightarrow L'$$

Duality of SVFB control and output feedback (OPFB) control is shown in [21] for leaderless consensus on the same graph. This is different from the duality pre-

sented in Theorem 3.3, i.e., duality of SVFB control design and observer design for cooperative tracking problems on reverse graphs. Theorem 3.3 extends the classical concept of duality in control theory [2] to networked systems on graphs.

3.6. Cooperative Dynamic Regulators for Synchronization

One approach for the design of controllers based on reduced state or output feedback (OPFB) is to integrate an observer with a state feedback controller. Then, the controller employs state estimate information instead of actual measured state information. Since this state estimate information is generated by an observer, which is a dynamical system using only system input and output information, this integrated controller is called a dynamic output feedback regulator or tracker. The dynamics of the regulator are provided by the observer.

In this section, we propose three cooperative dynamic output regulators to solve the cooperative tracking problem. The Riccati design-based control gain (3.10) and/or Riccati design based observer gain (3.20) are used in these three dynamic output feedback tracking controllers, all of which yield unbounded synchronization regions. This highlights the importance of local Riccati design approach in cooperative control of networked systems. It also highlights the increased richness of structure inherent in cooperative control on graphs, which admit three natural structures of regulators instead of the single standard regulator based on SVFB and full observer design in the single system case.

The work in this section comes from [23]. Various papers have designed variants of the three regulators described in this section including [14].

3.6.1 Neighborhood Controller and Neighborhood Observer

Recall the full SVFB cooperative control protocol (3.5), that is $u_i = cK\varepsilon_i$. Recall the *local neighborhood output estimation error* (3.15) for node i, re-written here as

$$\zeta_i = \sum_{j=1}^{N} a_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(\tilde{y}_0 - \tilde{y}_i) \tag{3.25}$$

Define the *local neighborhood state estimation tracking error* as

$$\hat{\varepsilon}_i = \sum_{j \in N_i} a_{ij}(\hat{x}_j - \hat{x}_i) + g_i(\hat{x}_0 - \hat{x}_i). \tag{3.26}$$

These two local neighborhood errors represent the complete information available to node *i* based on measured output information from itself and its neighbors in the graph.

In the first regulator protocol, both controller and observer are designed using this complete available local neighborhood information. This cooperative regulator is designed using OPFB as

$$u_i = cK\hat{\varepsilon}_i, \tag{3.27}$$

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\zeta_i. \tag{3.28}$$

Then, the closed-loop dynamics of node *i* ($i \in \mathcal{N}$) is

$$\dot{x}_i = Ax_i + cBK\left(\sum_{j \in N_i} a_{ij}(\hat{x}_j - \hat{x}_i) + g_i(\hat{x}_0 - \hat{x}_i)\right),$$

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\left(\sum_{j \in N_i} a_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(\tilde{y}_0 - \tilde{y}_i)\right).$$

Assume $\hat{x}_0 = x_0$ for the leader node. Considering the identity $(L \otimes BK)\underline{x}_0 = 0$ which roots in the fact $L1_N = 0$, with 1_N the *N*-vector of 1's, the global closed-loop dynamics can be written as

$$\dot{x} = (I_N \otimes A)x - c[(L+G) \otimes BK](\hat{x} - \underline{x}_0), \tag{3.29}$$

$$\begin{aligned}\dot{\hat{x}} &= [I_N \otimes A - c(L+G) \otimes FC]\hat{x} + c[(L+G) \otimes F]y \\ &= -c[(L+G) \otimes BK](\hat{x} - \underline{x}_0)\end{aligned} \tag{3.30}$$

Theorem 3.4. *Let (A,B,C) be stabilizable and detectable and the augmented graph $\bar{\mathcal{G}}$ has a spanning tree with the leader node as the root. Let the cooperative dynamic OPFB tracking control law be (3.27) and (3.28). Design the SVFB gain K and coupling gain c according to Theorem 3.1 and the observer gain F according to Theorem 3.2. Then all nodes $i = 1, 2, \ldots, N$ synchronize to the leader node 0 asymptotically and the state estimation error $\tilde{x}$ in (3.17) approaches zero asymptotically.*

Proof. Equation (3.29) can be written as

$$\dot{x} = A_c x + B_c(\tilde{x} - \underline{x}_0).$$

Further computation gives

$$\dot{\delta} = A_c \delta + B_c \tilde{x}.$$

The global state estimation error dynamics (3.18) is

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}} = [I_N \otimes A - c(L+G) \otimes FC]\tilde{x} = A_o \tilde{x}.$$

Then one has

$$\begin{bmatrix} \dot{\delta} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A_c & B_c \\ 0 & A_o \end{bmatrix} \begin{bmatrix} \delta \\ \tilde{x} \end{bmatrix}.$$

The eigenvalues of a block triangular matrix are the combined eigenvalues of its diagonal blocks. Theorem 3.1 and Theorem 3.2 guarantee A_c and A_o are Hurwitz, respectively. This completes the proof.

Next, we analyze the performance of this proposed OPFB cooperative regulator using the concept of synchronization region. The synchronization region for this dynamic output feedback control law can be defined as follows.

Definition 3.3. *Consider the OPFB cooperative regulator (3.27) and (3.28). The synchronization region is the complex region defined as*

$$S_{OPFB} \triangleq \{s \in C \mid A - sBK \text{ is Hurwitz}\} \cap \{s \in C \mid A - sFC \text{ is Hurwitz}\}.$$

Corollary 3.2. *Consider the OPFB cooperative regulator (3.27) and (3.28). When the control gain K and observer gain F are designed as in Theorem 3.1 and Theorem 3.2, the synchronization region is an unbounded right-half plane. More specifically, a conservative synchronization region is* $S_{OPFB} = \{\alpha + j\beta \mid \alpha \in [1/2, \infty), \beta \in (-\infty, \infty)\}$.

Proof. Let $s = \alpha + j\beta$. Using the same development as in Theorem 3.1 and Theorem 3.2, we have

$$(A - sBK)^* P + P(A - sBK) = -Q - (2\alpha - 1)K^T RK.$$
$$(A - sFC)P + P(A - sFC)^* = -Q - (2\alpha - 1)FRF^T.$$

Therefore, $P > 0$, $Q > 0$ and $R > 0$, $\alpha \geq 1/2$ guarantees that both $A - sBK$ and $A - sFC$ are Hurwitz.

3.6.2 Neighborhood Controller and Local Observer

The OPFB cooperative regulator just presented uses the complete neighborhood information available for both the controller design (3.27) and the observer design (3.28). In cooperative multi-agent systems on graphs, to achieve synchronization it is not necessary to use neighbor information in both the control protocol and the observer protocol, as long as one or the other contains neighbor information. This reveals a richness of structure not appearing in the case of standard single agent systems.

The OPFB cooperative regulator designed in this section uses the neighborhood estimate tracking error information $\hat{\varepsilon}_i$ for the controller design and only the local agent output estimation error information $\tilde{y}_i$ for the observer design. Here by "local", we mean the information only about node i itself.

For node i, the controller uses complete neighbor information and is designed as

$$u_i = cK\hat{\varepsilon}_i = cK\left(\sum_{j \in N_i} a_{ij}(\hat{x}_j - \hat{x}_i) + g_i(\hat{x}_0 - \hat{x}_i)\right), \tag{3.31}$$

whereas the observer use only local agent information and is designed as

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\tilde{y}_i. \tag{3.32}$$

The closed-loop dynamics of node i ($i \in \mathcal{N}$) is

$$\dot{x}_i = Ax_i + cBK\left(\sum_{j \in N_i} a_{ij}(\hat{x}_j - \hat{x}_i) + g_i(\hat{x}_0 - \hat{x}_i)\right),$$

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\left(y_i - \hat{y}_i\right).$$

Here, the second equation is that of a local agent observer not coupled to information from any neighbors. The global closed-loop dynamics can be written as

$$\dot{x} = (I_N \otimes A)x - c[(L+G) \otimes BK](\hat{x} - \underline{x}_0),$$

$$\dot{\hat{x}} = [I_N \otimes (A+cFC)]\hat{x} - c[(L+G) \otimes BK](\hat{x} - \underline{x}_0) - (I_N \otimes cFC)x.$$

It can be shown that

$$\begin{bmatrix} \dot{\delta} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A_c & B_c \\ 0 & I_N \otimes (A+cFC) \end{bmatrix} \begin{bmatrix} \delta \\ \tilde{x} \end{bmatrix}. \tag{3.33}$$

Theorem 3.5. *Let (A,B,C) be stabilizable and detectable and the augmented graph* $\bar{\mathcal{G}}$ *has a spanning tree with the leader node as the root. Let the OPFB cooperative dynamic tracking control law be* (3.31) *and* (3.32). *Design the SVFB gain K and coupling gain c according to Theorem 3.1. Design the observer gain F such that the local agent observer* $(A + cFC)$ *is Hurwitz. Then all nodes* $i = 1, 2, ..., N$ *synchronize to the leader node 0 asymptotically and the state estimation error* $\tilde{x}$ *approaches zero asymptotically.*

Proof. Similar to Theorem 3.4, thus omitted.

Since (A, C) is detectable, an F can be selected such that $(A + cFC)$ is Hurwitz. Also, F can be designed using the local Riccati equation as in Theorem 3.2, except that

$$F = -PC^T R^{-1}.$$

A similar synchronization region analysis for this OPFB cooperative regulator can be carried out as that in Section 3.6.1.

3.6.3 Local Controller and Neighborhood Observer

This third OPFB cooperative regulator design uses only the local agent "state estimate" for controller design and the complete neighborhood information for "observer" design. For node *i*, the controller portion is designed using local agent feedback based on a sort of 'estimate' of the leaders' state

$$u_i = K(\hat{x}_i - \hat{x}_{0i}), \tag{3.34}$$

$$\dot{\hat{x}}_{0i} = (A + BK)\hat{x}_{0i} \tag{3.35}$$

and using complete neighbor information in the 'observer' dynamics

$$\dot{\hat{x}}_i = (A + BK)\hat{x}_i - cF\left(\sum_{j \in N_i} a_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(y_0 - C\hat{x}_{0i} - \tilde{y}_i) \right) \tag{3.36}$$

This third design is somewhat surprising since the controller uses no information about the neighbors, yet the following result shows that synchronization is reached using this rather odd OPFB cooperative regulator.

Theorem 3.6. *Let (A,B,C) be stabilizable and detectable and the augmented graph* $\bar{\mathcal{G}}$ *has a spanning tree with the leader node as the root. Let the cooperative dy-*

namic OPFB tracking control law be (3.34), (3.35) *and* (3.36). *Design the SVFB gain K such that $A + BK$ is Hurwitz. Design the "observer" gain F and the coupling gain c according to Theorem 3.2. Then all nodes $i = 1,2,...,N$ synchronize to the leader node 0 asymptotically.*

Proof. Define $\mu = col(\mu_1, \mu_2, ..., \mu_N)$ with $\mu_i = x_i - x_0$, $\eta = col(\eta_1, \eta_2, ..., \eta_N)$ with $\eta_i = \hat{x}_i - \hat{x}_{0i}$, and $\theta = col(\mu, \eta)$. Then asymptotic stability of θ implies that $lim_{t \to \infty}(x_i(t) - x_0(t)) = 0$ and $lim_{t \to \infty}(\hat{x}_i(t) - \hat{x}_{0i}(t)) = 0$, $\forall i \in \mathcal{N}$. Let $\hat{x}_{0i}(t_0) = \hat{x}_{0j}(t_0)$, then $\hat{x}_{0i}(t) = \hat{x}_{0j}(t)$, $\forall i, j \in \mathcal{N}$ and $\forall t \geq t_0$. Straightforward computation yields the dynamics of θ as

$$\dot{\theta} = \begin{bmatrix} I_N \otimes A & I_N \otimes BK \\ c(L+G) \otimes FC & I_N \otimes BK + A_o \end{bmatrix} \theta = A_\theta \theta. \tag{3.37}$$

Matrix A_θ is similar to the matrix

$$\begin{bmatrix} I_N \otimes (A + BK) & I_N \otimes BK \\ 0 & A_o \end{bmatrix}.$$

In fact, $T^{-1} A_\theta T = \begin{bmatrix} I_N \otimes (A + BK) & I_N \otimes BK \\ 0 & A_o \end{bmatrix}$, where $T = \begin{bmatrix} I_{nN} & 0 \\ I_{nN} & I_{nN} \end{bmatrix}$ and $T^{-1} = \begin{bmatrix} I_{nN} & 0 \\ -I_{nN} & I_{nN} \end{bmatrix}$.

Therefore, θ is asymptotically stable if and only if both $I_N \otimes (A + BK)$ and A_0 are Hurwitz, which are equivalent to the conditions that $A + BK$ and $A - c\lambda_i FC$, $\forall i \in \mathcal{N}$ are Hurwitz (see Lemma 3.5). Theorem 3.2 guarantees stability of $A - c\lambda_i FC$. A suitable K can be selected since (A, B) is stabilizable.

Similar synchronization region analysis can be carried out as in Section 3.6.1.

Note that the derivation of (3.37) requires the conditions $\hat{x}_{0i}(t_0) = \hat{x}_{0j}(t_0)$, $\forall i, j \in \mathcal{N}$. A special case is when $\hat{x}_{0i}(t_0) = 0$, $\forall i \in \mathcal{N}$. Then $\hat{x}_{0i}(t) = 0$, $\forall i \in \mathcal{N}$, $t \geq t_0$ and the control law (3.34) and (3.36) are simplified as

$$u_i = K\hat{x}_i,$$

$$\dot{\hat{x}}_i = (A + BK)\hat{x}_i - cF\left(\sum_{j \in N_i} a_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(y_0 - \tilde{y}_i) \right).$$

Remark 3.4. The "observer" (3.36) was proposed in [14]. It is not a true observer, in the sense that $\lim_{t\to\infty}(\hat{x}_i - x_i) \neq 0$ and $\lim_{t\to\infty}(\hat{x}_{0i} - x_0) \neq 0$. The variables $\hat{x}_i$ and $\hat{x}_{0i}$ only take roles of intermediate variables in the controller design.

3.7. Simulation Examples

A large scope of industrial applications can be modeled as the mass-spring system, including vibration in mechanical systems, animation of deformable objects, etc. In this example, we consider a cooperative tracking problem with one leader node and 6 follower nodes. Each node is a two-mass-spring system with a single force input, except for the leader node, which is unforced. The system is shown in Fig. 3.5, where m_1 and m_2 are masses, k_1 and k_2 are spring constants, u_i is the force input for mass 1, and $y_{i,1}$ and $y_{i,2}$ are the displacements of the two masses.

Define the state vector for node i as $x_i = [x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}]^T = [y_{i,1}, \dot{y}_{i,1}, y_{i,2}, \dot{y}_{i,2}]^T$ and the measured output information as $y_i = [y_{i,1}, y_{i,2}]^T$. Then, this two-mass-spring system can be modeled by

$$\dot{x}_i = Ax_i + Bu_i \tag{3.38}$$

$$y_i = Cx_i \tag{3.39}$$

with $A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k_1-k_2}{m_1} & 0 & \frac{k_2}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_2}{m_2} & 0 & \frac{-k_2}{m_2} & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ \frac{1}{m_1} \\ 0 \\ 0 \end{bmatrix}$, and $C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$.
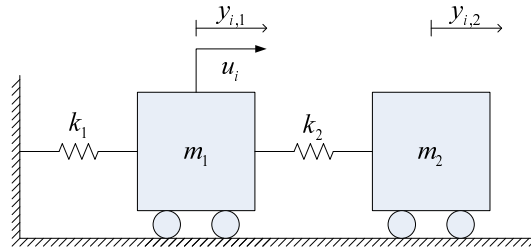
Fig. 3.5. Two-mass-spring system

Let the command generator leader node consist of an unforced two-mass-spring system, producing a desired state trajectory. Six two-mass-spring systems act as follower nodes and these nodes receive state or output information from their neighbors, according to the communication graph topology described in Fig. 3.6. The edge weights of the links are shown in the figure.
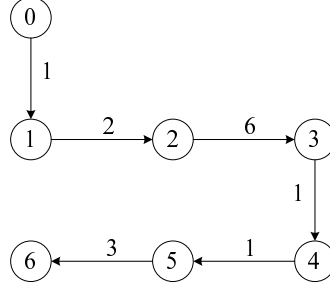
Fig. 3.6. Communication graph topology

The objective of the cooperative tracking control problem is to design distributed controllers u_i for the follower nodes, such the displacements for both of the two masses synchronize to that of the leader node, i.e. $lim_{t\to\infty}(y_{i,1} - y_{0,1}) = 0$ and $lim_{t\to\infty}(y_{i,2} - y_{0,2}) = 0$ for $i = 1,2,...,6$. The displacement $y_{i,2}$ of mass 2 is difficult to control using only the force input to mass 1.

In the following, we shall verify the proposed cooperative control algorithms, namely, the SVFB control law from Theorem 3.1 and the three types of dynamic OPFB cooperative trackers in Section 3.6. In the simulations, $m_1 = 1.1$ kg, $m_2 = 0.9$ kg, $k_1 = 1.5$ N/m and $k_2 = 1$ N/m. The coupling gain $c = 2$, which satisfies the condition (3.12), (3.22) for this graph topology. The initial values x_i ($i = 0,1,...,6$) are random and are generated by the Matlab function *randn*.

a. SVFB Cooperative tracker

First, we assume full SVFB, so that the displacements and velocities of both mass 1 and mass 2 are measurable for all agents. The distributed control law is (3.5) with *c, K* found for each agent by using the Riccati equation design method from Theorem 3.1. The tracking performances are depicted in Fig. 3.7 - Fig. 3.8. These figures show that displacements $y_{i,1}, y_{i,2}$ $(i = 1,...,6)$ synchronize to the displacements $y_{0,1}, y_{0,2}$ of the leader node within a few seconds. Velocity tracking is also achieved and the figures are omitted here to avoid redundancy.

b. OPFB Dynamic Cooperative Trackers

When the velocities of mass 1 and mass 2 are not measurable, but only position feedback is possible, we design three types of dynamic OPFB cooperative controllers. We use the three design methods for dynamic cooperative trackers given in Section 3.6. For convenience, we label the three cases as

- Case 1: OPFB with neighborhood controller and neighborhood observer (Section 3.6.1).
- Case 2: OPFB with neighborhood controller and local observer (Section 3.6.2).
- Case 3: OPFB with local controller and neighborhood observer (Section 3.6.3).

In the simulation, for Case 2, the observer gain is taken as $F = \begin{bmatrix} -1.2124 & -0.1259 \\ -0.2428 & -0.0944 \\ -0.1259 & -1.4329 \\ -0.2386 & -0.5345 \end{bmatrix}$, which makes $A + cFC$ Hurwitz. For Case 3, the control

gain is taken as $K = \begin{bmatrix} -0.9907 & -1.7831 & 0.4292 & -0.9807 \end{bmatrix}$, which makes $A + BK$ Hurwitz.

Using these types of dynamic OPFB cooperative trackers, the displacement tracking performance and estimation performance are illustrated, respectively, in Fig. 3.9-Fig. 3.17. Note that Fig. 3.17 shows that, for Case 3, $\hat{x}_i$ is not a true observer for x_i, nor $\hat{x}_{0i}$ an observer for x_0. These variables simply act as intermediate variables in the control protocol and are part of the dynamics of the OPFB cooperative tracker.
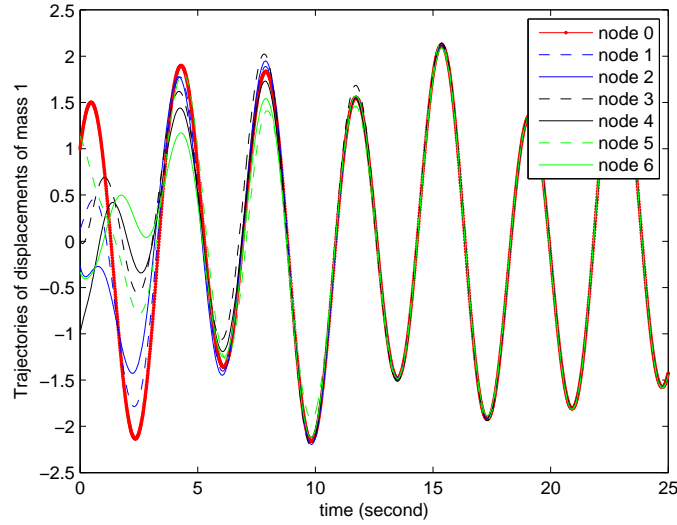
Fig. 3.7. Cooperative control using SVFB and local Riccati design. Profiles of the displacements $y_{i,1}$ $(i = 0, \ldots, 6)$ of mass 1
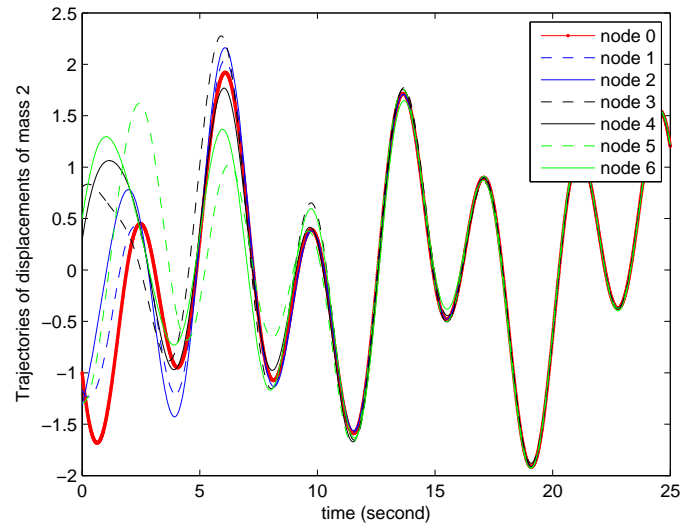
Fig. 3.8. Cooperative control using SVFB and local Riccati design. Profiles of the displacements $y_{i,2}$ $(i = 0, \ldots, 6)$ of mass 2
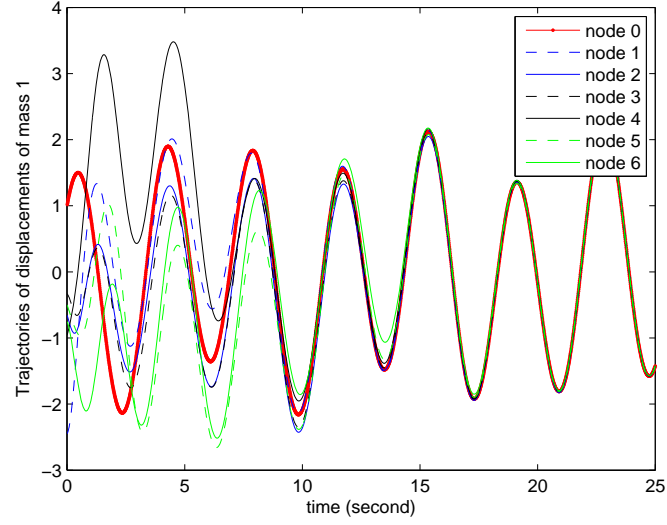
Fig. 3.9. Dynamic OPFB cooperative tracker of Case 1. Profiles of the displacements $y_{i,1}$ ($i = 0,\ldots,6$) of mass 1
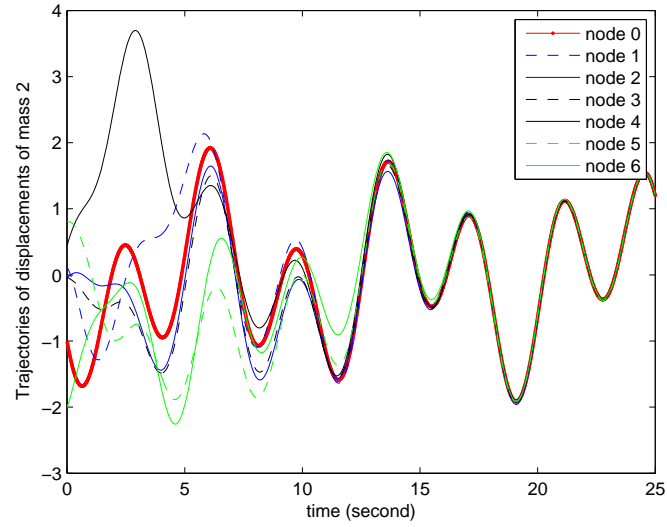
Fig. 3.10. Dynamic OPFB cooperative tracker of Case 1. Profiles of the displacements $y_{i,2}$ ($i = 0,\ldots,6$) of mass 2
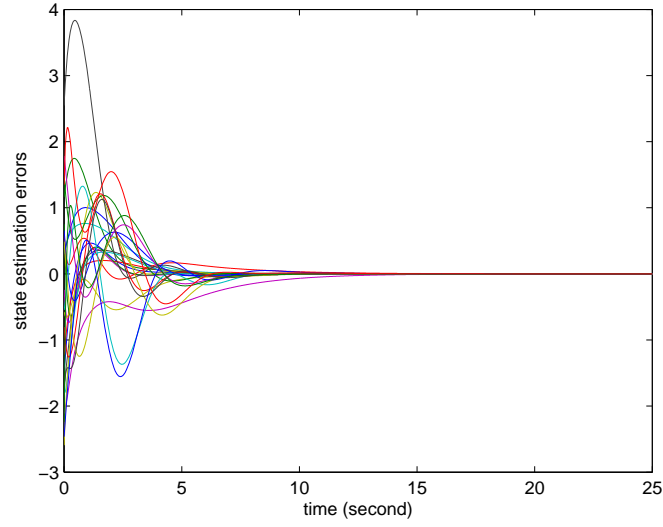
Fig. 3.11. Dynamic OPFB cooperative tracker of Case 1. Profiles of the state estimation errors $\tilde{x}_{i,j}$ ($i = 1, \ldots, 6$; $j = 1, \ldots, 4$)
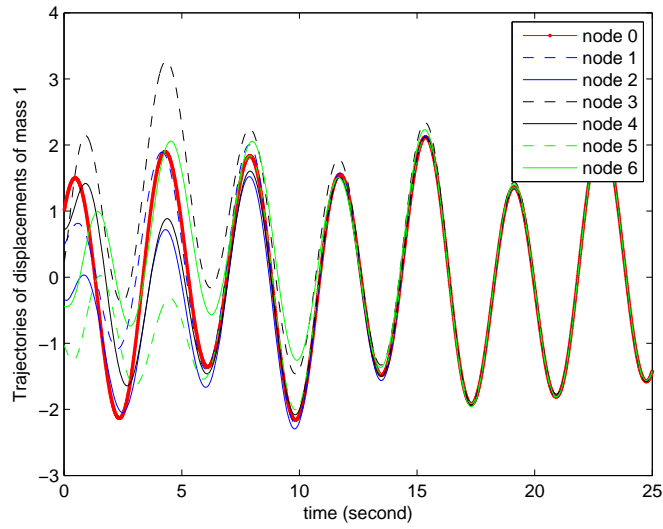
Fig. 3.12. Dynamic OPFB cooperative tracker of Case 2. Profiles of the displacements $y_{i,1}$ ($i = 0, \ldots, 6$) of mass 1
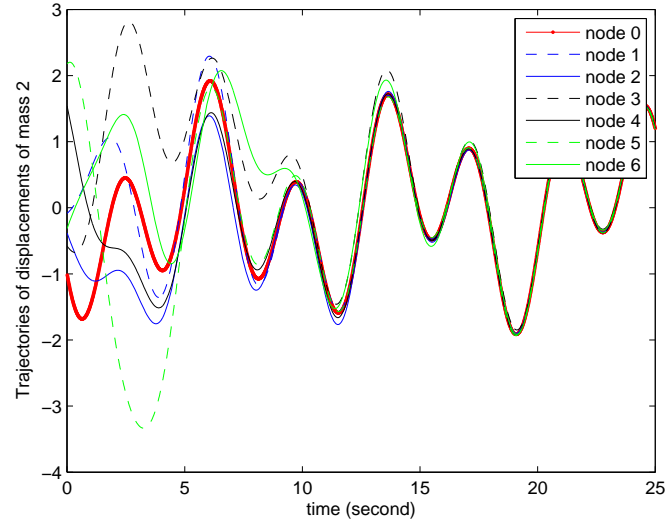
Fig. 3.13. Dynamic OPFB cooperative tracker of Case 2. Profiles of the displacements $y_{i,2}$ (
$i = 0, \ldots, 6$) of mass 2
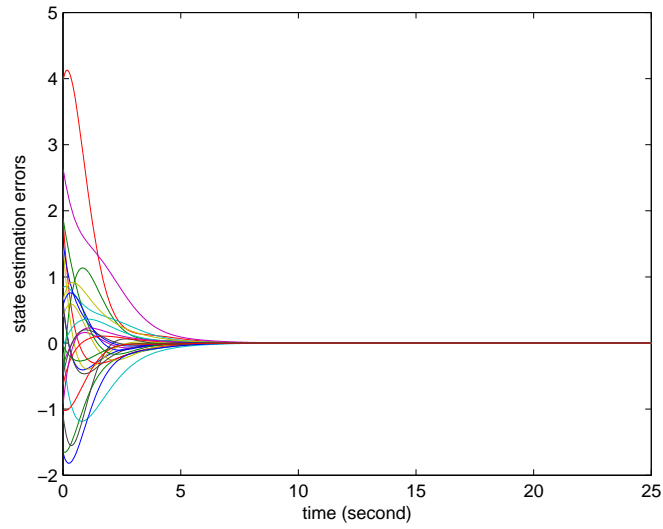
Fig. 3.14. Dynamic OPFB cooperative tracker of Case 2. Profiles of the state estimation errors
$\tilde{x}_{i,j}$ ($i = 1, \ldots, 6$; $j = 1, \ldots, 4$)
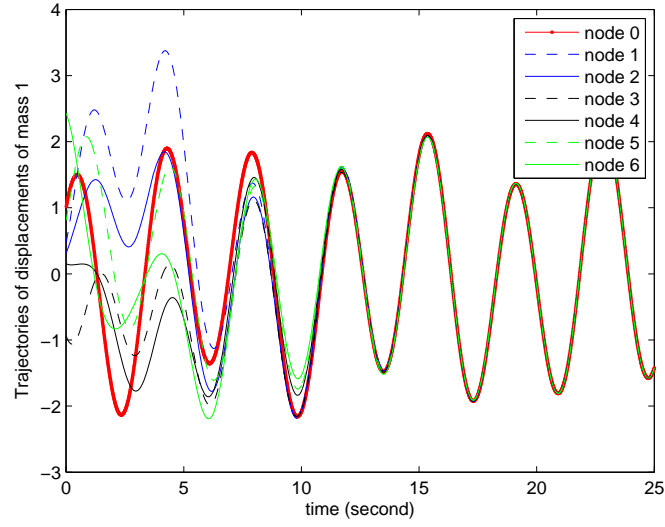
Fig. 3.15. Dynamic OPFB cooperative tracker of Case 3. Profiles of the displacements $y_{i,1}$ ($i = 0,\ldots,6$) of mass 1
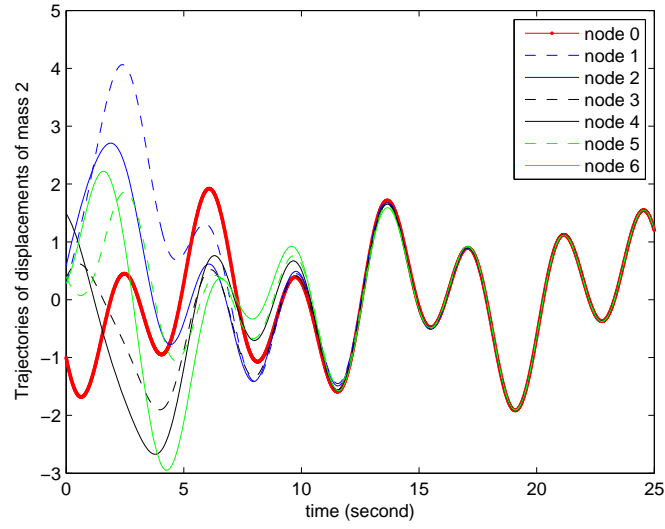
Fig. 3.16. Dynamic OPFB cooperative tracker of Case 3. Profiles of the displacements $y_{i,2}$ ($i = 0,\ldots,6$) of mass 2
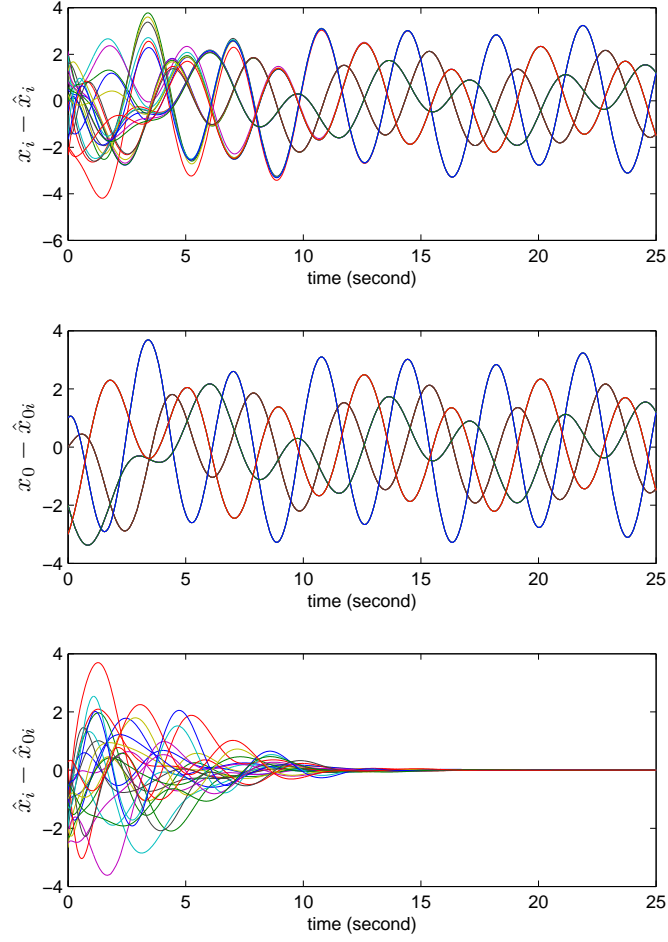
Fig. 3.17. Dynamic OPFB cooperative tracker of Case 3. Profiles of $x_{i,j} - \hat{x}_{i,j}$, $x_{0,j} - \hat{x}_{0i,j}$ and $\hat{x}_{i,j} - \hat{x}_{0i,j}$ ($i = 1, \ldots, 6$; $j = 1, \ldots, 4$)

To the Springer Editorial Team

The references in Chapters 2 through 10 need to be listed alphabetically as they are listed below.

Thank you.

Author Lewis

References

[1] P.J. Antsaklis and A.N. Michel. *A linear systems primer*. Birkhauser, 2007.

[2] C.T. Chen. *Linear System Theory and Design*. Holt, Rinehart and Winston, New York, 1984.

[3] G. Chen, F.L. Lewis, and L. Xie. Finite-time distributed consensus via binary control protocols. *Automatica*, 47(9): 1962–19685, 2011

[4] A. Das and F.L. Lewis. Distributed adaptive control for synchronization of unknown nonlinear networked systems. *Automatica*, 46(12): 2014–2021, 2010.

[5] A. Das and F.L. Lewis. Cooperative adaptive control for synchronization of second-order systems with unknown nonlinearities. *Int. J. Robust. Nonlinear Control*, 21(13): 1509–1524, 2011.

[6] H. Du, S. Li, and C. Qian. Finite-time attitude tracking control of spacecraft with application to attitude synchronization. *IEEE Trans. Autom. Control*, 56(11): 2711-2717, 2011.

[7] Z. Duan, G. Chen, and L. Huang. Disconnected synchronized regions of complex dynamical networks. *IEEE Trans. Autom. Control*, 54(4): 845–849, 2009.

[8] J.A. Fax and R.M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Trans. Autom. Control*, 49(9): 1465–1476, 2004.

[9] R.A. Horn and C.R. Johnson. *Matrix analysis*. Cambridge Univ Press, 1990.

[10] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, 48(6): 988–1001, 2003.

[11] S. Khoo, L. Xie, and Z. Man, "Robust finite-time consensus tracking algorithm for multirobot systems," *IEEE Trans. Mechatron.*,14(2): 219–228, 2009.

[12] F.L. Lewis, L. Xie, and D. Popa, *Optimal & Robust Estimation: With an Introduction to Stochastic Control Theory*, 2nd ed., CRC Press, Boca Raton, 2007.

[13] F.L. Lewis, D. Vrabie, and V.L. Syrmos. *Optimal Control*. 3rd ed., Wiley, New York, 2012.

[14] Z. Li, Z. Duan, G. Chen, and L. Huang. Consensus of multiagent systems and synchronition of complex networks: a unified viewpoint. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 57(1): 213–224, 2010.

[15] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control*, 49(9): 1520–1533, 2004.

[16] L. Qiu and K. Zhou. *Introduction to feedback control*. Prentice Hall, 2010.

[17] W. Ren, R.W. Beard, and E.M. Atkins. A survey of consensus problems in multi-agent coordination. In *Proc. Amer. Control Conf.*, pages 1859–1864, Portland, OR, 2005.

[18] L. Scardovi and R. Sepulchre. Synchronization in networks of identical linear systems. *Automatica*, 45(11): 2557–2562, 2009.

[19] J.H. Seo, H. Shim, and J. Back. Consensus of high-order linear systems using dynamic output feedback compensator: low gain approach. *Automatica*, 45(11): 2659–2664, 2009.

[20] P.N. Shivakumar and K.H. Chew. A sufficient condition for nonvanishing of determinants. *Proc. Amer. Math. Soc.*, 43(1): 63–66, 1974.

[21] S.E. Tuna. LQR-based coupling gain for synchronization of linear systems. *Arxiv preprint arXiv: 0801.3390*, 2008.

[22] X.. Wang and G. Chen. Pinning control of scale-free dynamical networks. *Physica A*, 310(3): 521–531, 2002.

[23] H. Zhang, F.L. Lewis, and A. Das. Optimal design for synchronization of cooperative systems: state feedback, observer and output feedback. *IEEE Trans. Autom. Control*, 56(8): 1948–1952, 2011.

4
Riccati Design for Synchronization of Discrete-Time Systems

In this chapter design methods are given for synchronization control of discrete-time multi-agent systems on directed communication graphs. The graph is assumed to have fixed topology and contain a spanning tree. The graph properties complicate the design of synchronization controllers due to the interplay between the eigenvalues of the graph Laplacian matrix and the required stabilizing gains. A method is given that decouples the design of the synchronizing feedback gains from the detailed graph properties. It is based on computation of the agent feedback gains using local Riccati equation design. Conditions are given for synchronization based on the relation of the graph eigenvalues to a bounded circular region in the complex plane that depends on the agent dynamics and the Riccati solution. The notion of 'synchronization region' is used. Convergence to consensus and robustness properties are investigated. This chapter also investigates the design of distributed observers for identical agents using local Riccati design. Cooperative observer design guaranteeing convergence of the estimates of all agents to their actual states is proposed. The notion of convergence region for distributed observers on graphs is introduced.

The feedback control synchronization region and observer convergence region for discrete-time systems are inherently bounded, so that the conditions for state synchronization and observer convergence are stricter than the results for the continuous-time counterparts given in Chapter 3. This is, in part, remedied by using weighting by different feedback coupling gains for every agent based on its in-degree.

A duality principle that involves the notion of reverse graph is shown for state feedbacks and observers for distributed discrete-time systems on balanced graph topologies. Three different cooperative controller/observer architectures are proposed for dynamic output feedback regulator design, and all are shown to guarantee convergence of the estimate to the true state as well as synchronization of all the agents' states to the command state trajectory. This provides design methods for cooperative regulators based on a separation principle. Examples show the effectiveness of these design methods for guaranteeing synchronization in cooperative discrete-time systems.

The results in this chapter were first published in [5].

4.1. Graph Properties

The information flow between agents is modeled here as a communication graph. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a nonempty finite set of N vertices or nodes $\mathcal{V} = \{v_1, \cdots, v_N\}$ and a set of edges or arcs $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. It is assumed that the graph is simple, *i.e.* there are no repeated edges or self-loops, $(v_i, v_i) \notin \mathcal{E}, \forall i$. General directed graphs (digraphs) are considered and it is taken that information propagates through the graph along directed arcs. Denote the connectivity or adjacency matrix as $E = [e_{ij}]$ with $e_{ij} > 0$ if $(v_j, v_i) \in \mathcal{E}$ and $e_{ij} = 0$ otherwise. Note that diagonal elements satisfy $e_{ii} = 0$ because the graph is simple. The set of neighbors of node v_i is denoted as $\mathcal{N}_i = \{v_j : (v_j, v_i) \in \mathcal{E}\}$, *i.e.* the set of nodes with arcs incoming into v_i. Define the in-degree matrix as a diagonal matrix $D = diag(d_1 \ldots d_N)$ with $d_i = \sum_j e_{ij}$ the (weighted) in-degree of node i (*i.e.* the *i-th* row sum of E). Define the graph Laplacian matrix as $L = D - E$, which has all row sums equal to zero.

A directed path from node v_{i_1} to node v_{i_k} is a sequence of edges $(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), \ldots (v_{i_{k-1}}, v_{i_k})$, with $(v_{i_{j-1}}, v_{i_j}) \in \mathcal{E}$ for $j = \{2, \ldots, k\}$. The graph is said to contain a (directed) spanning tree if there exists a vertex such that every other vertex in $\mathcal{V}$ can be connected by a directed path starting from that vertex. Such a special vertex is then called a root.

Given the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the reverse graph $\mathcal{G}'$ is a graph having the same set of vertices $\mathcal{V}' = \mathcal{V}$ and the set of edges with the property $E' = [e'_{ij}] = E^T = [e_{ji}]$. That is, the edges of $\mathcal{G}$ are reversed in $\mathcal{G}'$.

We denote the real numbers by $\mathbb{R}$, the positive real numbers by $\mathbb{R}^+$, and the complex numbers by $\mathbb{C}$. For any matrix A, $\sigma_{min}(A), \sigma_{max}(A)$ are the minimal and maximal singular values of A respectively. $C(O, r)$ denotes an open circle in the complex plane with its center at $O \in \mathbb{C}$ and radius r. The corresponding closed circle is denoted as $\bar{C}(O, r)$. For any square matrix A the spectral radius $\rho(A) = \max |eig(A)|$ is the maximal magnitude of the eigenvalues of A.

4.2. State Feedback Design of Discrete-time Cooperative Controls

In this section we design cooperative controllers for discrete-time (DT) multi-agent systems using state variable feedback (SVFB), that is, assuming that the full state of each agent is available for feedback control design by itself and its out-neighbors in the communication graph. We require that any control protocol be distributed in the sense that the control for agent *i* depends only on its own information and that from its (in-)neighbors in the graph. It is shown that locally optimal design in terms of a local DT Riccati equation, along with selection of a suitable distributed control protocol, guarantees synchronization on arbitrary communication graphs that have a spanning tree.

4.2.1. Synchronization of Discrete-time Multi-agent Systems on Graphs

The starting point for consideration are the dynamics of the multi-agent system which are now specified. Given a communication graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ that describes the topology of the agent interactions, let each of its N nodes be endowed with a state vector $x_i \in \mathbb{R}^n$ and a control input $u_i \in \mathbb{R}^m$, and consider at each node the discrete-time linear time-invariant dynamics

$$x_i\left(k+1\right) = Ax_i\left(k\right) + Bu_i\left(k\right) \tag{4.1}$$

Assume that $\left(A,B\right)$ is stabilizable and B has full column rank m. This assumption is necessary for the subsequent feedback design.

A leader node, control node, or command generator has the autonomous drift dynamics with no input

$$x_0\left(k+1\right) = Ax_0\left(k\right) \tag{4.2}$$

with $x_0 \in \mathbb{R}^n$.

Though the leader dynamics (4.2) is autonomous and has no control input, it can generate a large class of useful command trajectories. The interesting cases are when matrix *A* is not stable. The types of trajectories that can be generated by the leader include unit step (position command), the ramp (useful in velocity tracking systems, *e.g.*, satellite antenna pointing), sinusoidal waveforms (useful, *e.g.*, in hard disk drive control), and more. In this chapter, *A* can be either stable, marginally stable, or even unstable, as long as (*A,B*) is stabilizable.

The *cooperative tracker* or *synchronization problem* is to select the control signals u_i, using only measurements from the neighbors of node i, such that all nodes synchronize to the state of the control node, that is, $\lim_{k\to\infty}\|x_i(k) - x_0(k)\| = 0, \forall i$. These requirements should be fulfilled for all initial conditions $x_i(0)$. If the trajectory $x_0(k)$ approaches a fixed point, this is normally called the consensus problem.

To achieve synchronization, define the local neighborhood tracking errors

$$\varepsilon_i = \sum_{j\in N_i} e_{ij}(x_j - x_i) + g_i(x_0 - x_i) \tag{4.3}$$

where pinning gain $g_i \geq 0$ is nonzero if node v_i can directly sense the state of control node. The intent is that only a small percentage of nodes have the control node as a neighbor with $g_i > 0$, yet all nodes should synchronize to the trajectory of the control node using local neighborhood control protocols. It is assumed that at least one pinning gain is nonzero, with pinning into a root node. Note that the local neighborhood tracking error represents the information available to agent i for control purposes.

Choose the input of agent i as the weighted local control protocol

$$u_i = c(1 + d_i + g_i)^{-1} K\varepsilon_i, \tag{4.4}$$

where c is a coupling gain to be detailed later. By 'weighted', we mean that the protocol is multiplied by the weighting factor $(1 + d_i + g_i)^{-1}$, which captures the local properties of the graph with respect to node i. Then, the closed-loop dynamics of the individual agents are given by

$$x_i(k+1) = Ax_i(k) + c(1 + d_i + g_i)^{-1} BK\varepsilon_i(k). \tag{4.5}$$

Defining global tracking error and state vectors $\varepsilon = \begin{bmatrix} \varepsilon_1^T & \cdots & \varepsilon_N^T \end{bmatrix}^T \in \mathbb{R}^{nN}$ $x = \begin{bmatrix} x_1^T & \cdots & x_N^T \end{bmatrix}^T \in \mathbb{R}^{nN}$, one writes the global error as

$$\varepsilon(k) = -(L+G)\otimes I_n x(k) + (L+G)\otimes I_n \overline{x}_0(k) \tag{4.6}$$

where $G = diag(g_1, \ldots, g_N)$ is the diagonal matrix of pinning gains and $\overline{x}_0(k) = \underline{1} \otimes x_0(k)$ with $\underline{1} \in \mathbb{R}^N$ the vector of 1's. The Kronecker product is $\otimes$. The global dynamics of the N-agent system is given by

$$x(k+1) = \left[I_N \otimes A - c(I+D+G)^{-1}(L+G) \otimes BK \right] x(k) + c(I+D+G)^{-1}(L+G) \otimes BK\overline{x}_0(k) . (4.7)$$

Define the *global disagreement error* $\delta(k) = x(k) - \overline{x}_0(k)$, which has the global disagreement error dynamics

$$\delta(k+1) = A_c\delta(k) \tag{4.8}$$

where the closed-loop system matrix is

$$A_c = \left[I_N \otimes A - c(I+D+G)^{-1}(L+G) \otimes BK \right]. \tag{4.9}$$

Matrix A_c reflects the local agent closed-loop matrix $A - BK$ as modified on the graph structure $L+G$. Note that the graph structure $L+G$ appears on the left-hand side of the Kronecker product, while the local state information appears on the right.

We refer to the matrix

$$\Gamma = (I+D+G)^{-1}(L+G) \tag{4.10}$$

as the (weighted) *graph matrix*. Its eigenvalues Λ_k, $k = 1, \ldots, N$ are important in the upcoming discussion, and we refer to them as the *graph matrix eigenvalues*.

Assumption 4.1. Assume the graph contains a directed spanning tree and has at least one nonzero pinning gain g_i into a root node *i*.

Under this assumption, the graph matrix Γ is nonsingular, since $L+G$ is nonsingular.

The importance of using weighting of $(I+D+G)^{-1}$ in protocol (4.4) is shown by the following result, which shows that all graph eigenvalues of Γ are restricted to a closed circle of radius 1 in the right-half complex plane.

Lemma 4.1. *Given the control protocol (4.4), the eigenvalues of Γ satisfy $\Lambda_k \subseteq \overline{C}(1,1)$, $k = 1 \ldots N$, for any graph.*

Proof. Follows directly from the Geršgorin circle criterion applied to $\Gamma = (I + D + G)^{-1}(L + G)$, which has Geršgorin circles $\overline{C}\left(\dfrac{d_i + g_i}{1 + d_i + g_i}, \dfrac{d_i}{1 + d_i + g_i}\right)$. These are all contained in $\overline{C}(1,1)$. $\blacksquare$

Lemma 4.1 reveals that weighting restricts the possible positions of the graph matrix eigenvalues to the bounded known circular region $\overline{C}(1,1)$. The non-weighted protocol $u_i = cK\varepsilon_i$ yields graph eigenvalues in the region $\bigcup_i \overline{C}(d_i + g_i, d_i)$, which is larger than $\overline{C}(1,1)$. The importance of using this weighting is further shown in Example 4.1.

Under Assumption 4.1, the graph eigenvalues are inside the circle $\overline{C}(1,1)$, and all have positive real parts. The next result was presaged in [1].

Lemma 4.2. *The multi-agent systems (4.5) synchronize if and only if* $\rho\left(A - c\Lambda_k BK\right) < 1$ *for all eigenvalues* Λ_k, $k = 1...N$, *of graph matrix (4.10).*

Proof. Let J be the Jordan form of Γ, *i.e.*, there exists a nonsingular matrix $R \in \mathbb{R}^{N \times N}$ such that $R\Gamma R^{-1} = J$. From (4.9), it follows that

$$(R \otimes I_n)A_c(R^{-1} \otimes I_n) = I_N \otimes A - cJ \otimes BK = \begin{bmatrix} A - c\Lambda_1 BK & \times & \times \\ 0 & \ddots & \times \\ 0 & 0 & A - c\Lambda_N BK \end{bmatrix}, \quad (4.11)$$

where '$\times$' denotes possibly nonzero elements. By (4.11), one has $\rho(A_c) < 1$ if and only if $\rho\left(A - c\Lambda_k BK\right) < 1$ for all Λ_k. From (4.8), the multi-agent systems (4.5) synchronize if and only if $\rho(A_c) < 1$. Thus, the multi-agent systems (4.5) synchronize if and only if $\rho\left(A - c\Lambda_k BK\right) < 1$ for all Λ_k. $\blacksquare$

For synchronization one requires the asymptotic stability of the error dynamics (4.8). It is assumed that (A, B) is stabilizable. If the matrix A is unstable or marginally stable, then Lemma 4.2 requires that $\Lambda_k \neq 0$, $k = 1...N$ which is guaranteed if the interaction graph contains a spanning tree with at least one nonzero pinning gain into a root node.

4.2.2 Synchronization Region

Building on Lemma 4.2 we introduce the notion of the synchronization region, which allows the analysis of the global multi-agent system cooperative stability by considering the robust stability of a single agent. The following definitions are required.

Definition 4.1: *For a matrix pencil* $A - sBK$, $s \in \mathbb{C}$ *the synchronization region is a subset* $S_c \subseteq \mathbb{C}$ *such that* $S_c = \left\{ s \in \mathbb{C} \, \middle| \, \rho(A - sBK) < 1 \right\}$.

Given the choice of K, the synchronization region S_c of the matrix pencil $A - sBK$ is determined. The synchronization region was discussed in [16],[23],[1],[9]. A concept related to synchronization region, yet of a more special form, is introduced in the following definition.

Definition 4.2: *Given a system* (A, B), *the complex gain margin region* $U \subseteq \mathbb{C}$ *for a given stabilizing feedback matrix* K, *is a connected region containing 1 such that* $\rho(A - sBK) < 1$, $\forall s \in U$.

The distinction between the more general synchronization region and the complex gain margin region is that the latter is connected, whereas the synchronization region may not be [10]. This means that the complex gain margin region can more easily be determined via, *e.g.* Lyapunov stability analysis. Clearly, the complex gain margin region is contained in the synchronization region.

Lemma 4.3. *Matrices* $A - c\Lambda_k BK$ *are stable for all eigenvalues* Λ_k *if and only if they satisfy* $c\Lambda_k \in S_c$, $\forall k \in \{1, \dots, N\}$.

According to this result the cooperative stability of the entire multi-agent system is reduced to a robust stabilization problem of the single agent system (A, B).

4.2.3 Local Riccati Design of Synchronizing Protocols

Given a single agent system (A, B) the synchronization region depends on the local feedback matrix K. Therefore, the distributed control design problem is to find a K that yields a guaranteed synchronization region or complex gain margin region. It will now be shown that selecting the local feedback matrix K as a locally optimal feedback, derived from a local algebraic Riccati equation, gives a guaranteed gain margin region. Then, it remains to check the relation of the graph matrix eigenvalues to the guaranteed synchronization region. Sufficient conditions for synchronization are given in the next theorem.

Theorem 4.1. *Suppose* (A, B) *is stabilizable. Assume that the interaction graph contains a spanning tree with at least one pinning gain nonzero that connects into the root node. Let $P > 0$ be a solution of the discrete-time Riccati-like equation*

$$A^T P A - P + Q - A^T P B (B^T P B)^{-1} B^T P A = 0 \tag{4.12}$$

for some prescribed $Q = Q^T > 0$. Define

$$r := \left[\sigma_{\max} (Q^{-1/2} A^T P B (B^T P B)^{-1} B^T P A Q^{-1/2}) \right]^{-1/2}. \tag{4.13}$$

Then protocol (4.4) guarantees synchronization of multi-agent systems (4.5) for some K if there exists a covering circle $C(c_0, r_0)$ of the graph matrix eigenvalues Λ_k, $k = 1 \ldots N$ such that

$$\frac{r_0}{c_0} < r. \tag{4.14}$$

Moreover, if condition (4.14) is satisfied then the choice of feedback matrix

$$K = \left(B^T P B \right)^{-1} B^T P A \tag{4.15}$$

and coupling gain

$$c = \frac{1}{c_0} \tag{4.16}$$

guarantee synchronization. ∎

The above reasoning is motivated by geometrical considerations as shown in Figure 4.1. Condition (4.14) means that the covering circle $C(c_0, r_0)$ of the graph eigenvalues can be projected by radial scaling from the origin into a circle concentric with and contained in the interior of $C(1, r)$. It will be shown in Lemma 4.4 that the synchronization region of the local Riccati feedback gain selected as in Theorem 4.1 contains $C(1, r)$. Therefore, synchronization is guaranteed by any value of c radially projecting $C(c_0, r_0)$ into the interior of $C(1, r)$. One such value is given by (4.16). If $C(c_0, r_0)$ can be projected by radial scaling from the origin into $C(1, r)$, then $C(c_0, r_0)$ is said to be *homothetic* to $C(1, r)$.

The importance of Theorem 4.1 is that it decouples the local SVFB gain design at each agent from the details of the interconnecting communication graph struc-

ture. The SVFB gain is designed by each agent using the local Riccati equation (4.12), and then the graph topology comes into the choice of the coupling gain c through condition (4.16). Unlike the continuous-time case discussed in Chapter 3, it is not always possible to select K and c to guarantee synchronization on any graph with a spanning tree. Indeed, condition (4.14) must be satisfied. It will be seen that this condition is a relation between the properties of the communication graph and the instability properties of the local agent dynamics.

The theorem shows that local Riccati SVFB control design by each agent guarantees the synchronization of all agents on any graph satisfying condition (4.14). Note that the SVFB gain in Theorem 4.1 is related to the optimal gain [8] that minimizes the local performance index

$$J_i = \frac{1}{2}\sum_{k=0}^{\infty} x_i^T(k)Qx_i(k) + u_i^T(k)R\rho u_i(k) \qquad (4.17)$$

subject to the local agent dynamics (4.1), in the limit as $\rho \to 0$.

The local Riccati design method in Theorem 4.1 is scalable, since it only depends on the local agent dynamics, which have state space of dimension n, and does not depend on the size N of the graph.
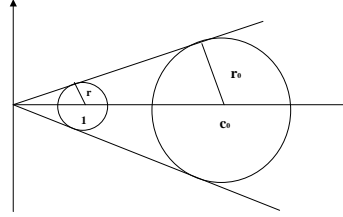
Fig. 4.1. Riccati design circle $C(1,r)$ and covering circle $C(c_0, r_0)$ of graph matrix eigenvalues.

The following technical lemmas are needed for the proof of Theorem 4.1.

Lemma 4.4. *The synchronization region for the choice of K given earlier contains the open circle $C(1,r)$.*

Proof. By choosing the state feedback matrix as (4.15) with $P > 0$ a solution of equation (4.12) one has

$$A^T PA - P + Q - K^T \left(B^T PB \right) K = 0 \qquad (4.18)$$

From this equation one obtains by completing the squares

$$A^T PA - P + Q - K^T B^T PBK = (A - BK)^T P(A - BK) - P + Q = 0. \quad (4.19)$$

Therefore, given the quadratic Lyapunov function $V(y) = y^T Py$, $y \in \mathbb{R}^n$, the choice of feedback gain stabilizes the system (A, B). At least a part of the synchronization region S can be found as

$$(A - sBK)^* P(A - sBK) = (A - \operatorname{Re} sBK)^T P(A - \operatorname{Re} sBK) + \operatorname{Im}^2 s(BK)^T PBK < P, \quad (4.20)$$

where * denotes complex conjugate transpose (Hermitian adjoint), $s \in \mathbb{C}$. Therefore from equations (4.18) and (4.20) one has

$$
\begin{aligned}
&(A - sBK)^* P(A - sBK) - P \\
&= A^T PA - P - s^* K^T B^T PA - sA^T PBK + s^* sK^T B^T PBK \\
&= A^T PA - P - 2\operatorname{Re} sK^T B^T PBK + |s|^2 K^T B^T PBK \\
&= A^T PA - P - (2\operatorname{Re} s - |s|^2)K^T B^T PBK \\
&= A^T PA - P - (1 - |s-1|^2)(BK)^T PBK
\end{aligned}
\quad (4.21)
$$

Inserting (4.19) into (4.21) yields

$$
\begin{aligned}
(A - sBK)^* P(A - sBK) - P &= -Q + K^T B^T PBK - (1 - |s-1|^2)K^T B^T PBK \\
&= -Q + |s-1|^2 K^T B^T PBK
\end{aligned}
$$

This is the condition of stability if

$$Q - |s-1|^2 K^T B^T PBK > 0, \quad (4.22)$$

which gives a simple bounded complex region, more precisely a part of the complex gain margin region U,

$$Q > |s-1|^2 K^T B^T PBK \Rightarrow 1 > |s-1|^2 \sigma_{\max}\left(Q^{-1/2}(BK)^T P(BK)Q^{-1/2}\right). \quad (4.23)$$

This is an open circle $C(1, r)$ specified by

$$|s-1|^2 < \frac{1}{\sigma_{max}\left(Q^{-1/2}K^T B^T PBKQ^{-1/2}\right)}. \tag{4.24}$$

Furthermore expressing K as the H_2 Riccati equation state feedback completes the proof. ∎

Lemma 4.5. *Given the circle $C(1,r)$ in the complex plane, which is contained in the synchronization region S for the Riccati choice of gain (4.15), the system is guaranteed to synchronize for some value of $c > 0$ if the graph matrix eigenvalues Λ_k, $k = 1...N$, are located in such a way that*

$$|c\Lambda_k - 1| < r, \forall k, \tag{4.25}$$

for that particular c in (4.16).

Proof. Follows from Lemmas 4.2, 4.3, and 4.4. ∎

Based on these constructions the proof of Theorem 4.1 can now be given.

Proof of Theorem 4.1. Given $C(1,r)$ contained in the synchronization region S_c of matrix pencil $A - sBK$, and the properties of dilation (homothety), and assuming there exists a directed spanning tree in the graph with a nonzero pinning gain into the root node, it follows that synchronization is guaranteed if all eigenvalues Λ_k are contained in a circle $C(c_0, r_0)$ similar with respect to homothety to a circle concentric with and contained within $C(1,r)$.

The center of the covering circle c_0 can be taken on the real axis due to symmetry and the radius equals $r_0 = \max_k |\Lambda_k - c_0|$. Taking these as given, one should have

$$\frac{r_0}{c_0} < \frac{r}{1}. \tag{4.26}$$

If this equation is satisfied then choosing $c = 1/c_0$ maps with homothety the covering circle of all eigenvalues $C(c_0, r_0)$ into a circle $C(1, r_0/c_0)$ concentric with and, for $r > r_0/c_0$, contained in the interior of the circle $C(1,r)$. ∎

If there exists a solution $P > 0$ to Riccati equation (4.12), B must have full column rank. Assuming B has full column rank, there exists a positive definite so-

lution P to (4.12) only if (A, B) is stabilizable. This is a necessary condition for the applicability of Theorem 4.1.

4.3. Robustness Property of Local Riccati Design

The following lemma is motivated by the conjecture that if the conditions of Theorem 4.1 hold there is in fact an open interval of admissible values for the coupling constant c. This has the interpretation of robustness for the local Riccati design in sense that synchronization is still guaranteed under small perturbations of c from the value given by (4.16).

Lemma 4.6. *Taking* $\alpha_{max} := \arccos \sqrt{1-r^2} \geq 0$ *and denoting the angle of an eigenvalue of the graph matrix* Γ *by* $\phi_k := \arg \Lambda_k$, *a necessary condition for the existence of at least one admissible value of* c *is given by*

$$\frac{\operatorname{Re}\Lambda_k}{|\Lambda_k|} > \sqrt{1-r^2}, \quad \forall k, \tag{4.27}$$

which is equivalent to

$$|\phi_k| < \alpha_{max}, \quad \forall k. \tag{4.28}$$

Furthermore the interval of admissible values of c, *if non-empty, is an open interval given as a solution to a set of inequalities*

$$\frac{\cos\phi_k - \sqrt{\cos^2\phi_k - \cos^2\alpha_{max}}}{|\Lambda_k|} < c < \frac{\cos\phi_k + \sqrt{\cos^2\phi_k - \cos^2\alpha_{max}}}{|\Lambda_k|}, \forall k. \tag{4.29}$$

Finally if (4.14) holds then the solution of (4.29) is non-empty and contains the value (4.16).

Proof. If solution c exists, the equations

$$|c\Lambda_k - 1| < r \Rightarrow c^2|\Lambda_k|^2 - 2c\operatorname{Re}\Lambda_k + 1 - r^2 < 0 \tag{4.30}$$

are satisfied simultaneously for every k for at least one value of c. Therefore $\forall k$ (4.30) has an interval of real solutions for c. From that, and the discriminant of (4.30), relation (4.27) follows.

One therefore has $\cos\phi_k = \dfrac{\mathrm{Re}\,\Lambda_k}{|\Lambda_k|} > \sqrt{1-r^2} = \cos\alpha_{max}$ meaning $|\phi_k| < \alpha_{max}$. Bearing

that in mind, expression (36) can further be equivalently expressed as

$$c^2\left|\Lambda_k\right|^2 - 2c\left|\Lambda_k\right|\cos\phi_k + \cos^2\alpha_{max} < 0,$$

Solving this equation for $c\left|\Lambda_k\right|$ yields N intervals in (4.29). The intersection of these N open intervals is either an open interval or an empty set.

Now given that $\Lambda_k \in C(c_0, r_0)$ one finds that

$$\frac{\cos\phi_k - \sqrt{\cos^2\phi_k - \cos^2\alpha_{max}}}{|\Lambda_k|} < \frac{\cos\phi_k - \sqrt{\cos^2\phi_k - \cos^2\alpha_{max}}}{\left|\Lambda_k\right|_{min}} \qquad (4.31)$$

$$\frac{\cos\phi_k + \sqrt{\cos^2\phi_k - \cos^2\alpha_{max}}}{\left|\Lambda_k\right|_{max}} < \frac{\cos\phi_k + \sqrt{\cos^2\phi_k - \cos^2\alpha_{max}}}{|\Lambda_k|}$$

where $\left|\Lambda_k\right|_{min}, \left|\Lambda_k\right|_{max}$ are extremal values for fixed ϕ_k, as determined by $C(c_0, r_0)$. Namely, for $\left|\Lambda_k\right|_{min}, \left|\Lambda_k\right|_{max}$ one has

$$\left|\Lambda_k - c_0\right| = r_0 \Rightarrow \frac{\left|\Lambda_k\right|^2}{c_0^2} - 2\frac{\left|\Lambda_k\right|}{c_0}\cos\phi_k + 1 - \frac{r_0^2}{c_0^2} = 0$$

which gives $\left|\Lambda_k\right|_{max,min} = c_0\left[\cos\phi_k \pm \sqrt{\cos^2\phi_k - \left(1 - \dfrac{r_0^2}{c_0^2}\right)}\right]$. If (4.14) is satisfied then

$\cos^2\alpha_{max} < 1 - \dfrac{r_0^2}{c_0^2}$, used in (4.31) together with expression for $\left|\Lambda_k\right|_{min}, \left|\Lambda_k\right|_{max}$ implies

the non-emptiness of the interval solution of (4.29), and guarantees that $c = 1/c_0$ is a member of that interval. Furthermore if the assumptions of Theorem 4.1 were satisfied with equality then the lower and higher limit of every subinterval (4.31) would in fact become equal to $1/c_0$. ∎

Condition (4.28) means that the graph matrix eigenvalues Λ_k must be inside the cone in the complex plane shown in Figure 4.1. It is evident from the geometry of the problem that eigenvalues Λ_k located outside the cone determined by (4.27), (4.28) cannot be made to fit into the region $C(1, r)$ by scaling with real values of

c. It should be noted that condition (4.14) is stronger than condition (4.28) and condition (4.29) has also been derived in [26].

4.4. Application to Real Graph Matrix Eigenvalues and Single-input Systems

In this section condition (4.14) of Theorem 4.1 is studied in several special simplified cases and it is shown how this condition relates to known results.

4.4.1 Case of Real Graph Eigenvalues

The special case of real eigenvalues of Γ and single input systems at each node allows one to obtain necessary and sufficient condition for synchronization.

Corollary 4.1. *Let graph* $\mathcal{G}(\mathcal{V},\mathcal{E})$ *have a spanning tree with pinning into the root node, and all eigenvalues of* Γ *be real and positive, that is,* $\Lambda_k > 0$ *for all* k . *Define* $0 < \Lambda_{\min} \leq \cdots \leq \Lambda_k \leq \cdots \leq \Lambda_{\max}$. *A covering circle for* Λ_k *that also minimizes* r_0 / c_0 , *is* $C(c_0, r_0)$ *with*

$$\frac{r_0}{c_0} = \frac{\Lambda_{\max} - \Lambda_{\min}}{\Lambda_{\max} + \Lambda_{\min}}.$$
(4.32)

Then, condition (4.14) in Theorem 4.1 becomes

$$\frac{\Lambda_{\max} - \Lambda_{\min}}{\Lambda_{\max} + \Lambda_{\min}} < r.$$
(4.33)

   Moreover, given that this condition is satisfied, the coupling gain choice (4.16) reduces to

$$c = \frac{2}{\Lambda_{\min} + \Lambda_{\max}}.$$
(4.34)

Proof. For graphs having a spanning tree with pinning into the root node, and all eigenvalues of Γ real, one has $0 < \Lambda_{\min}$. Note that in that case $\phi_k = 0 \ \forall k$ and $\cos^2 \alpha_{\max} = 1 - r^2$ so that necessary condition (4.28) is satisfied. Furthermore, inequalities (4.29) become

$$\frac{1-r}{\left|\Lambda_k\right|} < c < \frac{1+r}{\left|\Lambda_k\right|}.$$

From this, a necessary and sufficient condition for the existence of a nonempty intersection of these N intervals is

$$\frac{1-r}{\Lambda_{min}} < \frac{1+r}{\Lambda_{max}},$$

where the extremal values $\Lambda_{min} = \left|\Lambda_k\right|_{min}, \Lambda_{max} = \left|\Lambda_k\right|_{max}$ $\Lambda_{min,max} = \left|\Lambda_k\right|_{min,max}$ are the same for every k and are determined as in the proof of Lemma 4.6.

The sought interval is given as $c \in \left\langle \dfrac{1-r}{\Lambda_{min}}, \dfrac{1+r}{\Lambda_{max}} \right\rangle$. From this interval, one finds condition (4.33). Therefore, the sufficient condition (4.14) is equivalent to (4.33). Examining the positions of eigenvalues on the real axis it is found that for the minimal covering circle $C(r_0, c_0)$ one has

$$c_0 = \frac{\Lambda_{min} + \Lambda_{max}}{2},$$

$$r_0 = \Lambda_{max} - c_0 = c_0 - \Lambda_{min} = \frac{\Lambda_{max} - \Lambda_{min}}{2}.$$

So

$$\frac{\Lambda_{max} - \Lambda_{min}}{\Lambda_{max} + \Lambda_{min}} = \frac{r_0}{c_0} < r.$$

Note that in the case of real eigenvalues Λ_k their minimal covering circle has also the minimal ratio r_0/c_0 of all covering circles $C(c_0, r_0)$. This shows that (4.33) expresses the sufficient condition (4.14) specialized to Γ having all real eigenvalues. Theorem 4.1 then also gives the choice of c (4.34). ∎

This special case when the eigenvalues of the graph matrix Γ are real bears strong resemblance to the case presented and studied in [26] concerning undirected graphs. For, note that if one uses the non-weighted protocol $u_i = cK\varepsilon_i$ instead of (4.4), then, the eigenvalues Λ_k used in the analysis of this chapter are

those of $(L+G)$, not Γ in (4.10). However, the condition that the eigenvalues of $(L+G)$ be real is equivalent to the graph being undirected. It is noted that, even if all eigenvalues of $(L+G)$ are real, the eigenvalues of Γ may be complex, and vice versa. The importance of weighting is discussed in Lemma 4.1.

4.4.2 Case of Single-input Agent Dynamics

The radius r in (4.13) for Riccati-based design is important since it is instrumental in determining sufficient condition (4.14) in Theorem 4.1 as well as the condition in Lemma 4.6. In the case of single-input agent systems the expression (4.13) simplifies.

Remark 4.1. If the node dynamics (4.1) are single-input, define r by (4.13) with the choice of $Q = P^* - A^T P^* A + \dfrac{A^T P^* B B^T P^* A}{B^T P^* B}$, where $P^* > 0$ is a positive definite solution of the Riccati equation

$$P^* - A^T P^* A + \frac{A^T P^* B B^T P^* A}{B^T P^* B + 1} = 0. \tag{4.35}$$

Then

$$r = \frac{1}{\prod_u \left| \lambda^u(A) \right|} \tag{4.36}$$

where $\lambda^u(A)$ are the unstable eigenvalues of the system matrix A indexed by u.

This follows from Theorem 2.2 of [2]. Namely putting Q as defined into (4.12) yields the solution $P = P^*$. In case of single-input systems $B \in \mathbb{R}^n$ and u is a scalar, so the result in [2] applies. Note that if the solution to (4.35) is only positive semidefinite, a regularizing term must be added to (4.35) as is done in [2].

Note that P^* defined as a solution to Riccati equation (4.35) becomes the solution of (4.12) for the choice of Q given in Remark 4.1. This means, in the context of this chapter, that H_2 Riccati feedback gain for that specific choice of Q, applied to single-input systems, maximizes the value of r considered here to be

the radius of $C(1,r)$ in the complex plane, rather than just the real interval [2]. The following remark combines the above results.

Remark 4.2. If systems (4.1) are single-input and the Γ matrix of the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ has all eigenvalues real, then selecting Q as in Remark 4.1, gives the condition (4.14) in the form

$$\prod_u \left|\lambda^u(A)\right| < \frac{\Lambda_{max} + \Lambda_{min}}{\Lambda_{max} - \Lambda_{min}}.$$
(4.37)

Moreover this condition is necessary and sufficient for synchronization for any choice of the feedback matrix K if all the eigenvalues of A lie on or outside the unit circle. Sufficiency follows by Corollary 4.1 and Remark 4.1, assuming the conditions of Theorem 4.1 hold. Necessity follows from [26].

4.4.3 Mahler Measure, Graph Condition Number, and Graph Channel Capacity

The term appearing in Remark 4.2 involving the product of unstable eigenvalues

$$M(A) = \prod_i \left|\lambda_i^u(A)\right|$$
(4.38)

deserves further attention. $M(A)$ is known as the Mahler measure of the respective characteristic polynomial of A. It is related to the intrinsic entropy rate of a system $\sum_i \log_2 \left|\lambda_i^u(A)\right|$ describing the minimum data rate in a networked control system that enables stabilization of an unstable system [26].

It is well recognized that intrinsic entropy is of importance in the stability analysis of Kalman filtering with intermittent observations [20] and the quadratic stabilization of an uncertain linear system (cf. Theorem 2.1 of [4]).

We define the graph condition number as

$$\kappa(\Gamma) = \frac{\Lambda_{max}}{\Lambda_{min}}$$
(4.39)

In terms of the graph condition number condition (4.37) is written as

$$M(A) < \frac{1 + \kappa^{-1}}{1 - \kappa^{-1}}$$
(4.40)

The topological entropy of a system is defined as

$$h(A) = \log(M(A)) \tag{4.41}$$

Motivated by this, we define the graph channel capacity as

$$C(\Gamma) = \log \frac{1 + \kappa^{-1}}{1 - \kappa^{-1}} \tag{4.42}$$

4.5. Cooperative Observer Design

In this section we define cooperative observers that have a distributed structure on the interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. That is, the observer dynamics for each agent are distributed in the sense that they respect the communication structure imposed by the graph and can only use information from their neighbors. We provide a local Riccati design method for the observer gains that guarantees convergence of the observer errors to zero if a certain condition holds. Pinning control is used to provide a fixed reference in terms of a node 0 to some of the agents. All other, non-pinned, agent observers rely only on the relative output information of the neighbors

N agents are assumed coupled on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and have identical dynamics given as the discrete-time systems

$$x_i(k+1) = Ax_i(k) + Bu_i(k)$$
$$y_i(k) = Cx_i(k) \tag{4.43}$$

It is assumed that (A, C) is detectable and that C has full row rank. A control or leader node 0 has the command generator dynamics

$$x_0(k+1) = Ax_0(k)$$
$$y_0(k) = Cx_0(k) \tag{4.44}$$

The state, input, and output vectors are $x_i, x_0 \in \mathbb{R}^n$, $u_i \in \mathbb{R}^m$, $y_i, y_0 \in \mathbb{R}^p$. Note that matrix A may not be stable. The autonomous command generator describes many reference trajectories of interest including periodic trajectories, ramp-type trajectories, *etc*.

4.5.1 Distributed Neighborhood Observer Dynamics

It is desired to design cooperative observers that estimate the states $x_i(k)$ given only measurements of the outputs $y_j(k)$ of the agent i itself and of its neighbors. Given an estimated output $\hat{y}_i = C\hat{x}_i$ at node i, define the *local output estimation error*

$$\tilde{y}_i = y_i - \hat{y}_i.$$

To construct an observer that takes into account information from the neighbors of node i, define the *local neighborhood output disagreement*

$$\varepsilon_i^o = \sum_j e_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(\tilde{y}_0 - \tilde{y}_i) \tag{4.45}$$

where the observer pinning gains are $g_i \geq 0$, with $g_i > 0$ only for a small percentage of nodes who have direct measurements of the output error $\tilde{y}_0$ of the control node. With the global state vector defined as $x = \begin{bmatrix} x_1^T & \dots & x_N^T \end{bmatrix}^T \in \mathbb{R}^{nN}$ and global output $y = \begin{bmatrix} y_1^T & \dots & y_N^T \end{bmatrix}^T \in \mathbb{R}^{pN}$, the global output disagreement error $\varepsilon^o(k) \in \mathbb{R}^{pN}$ is

$$\varepsilon^o(k) = -(L+G) \otimes I_m \tilde{y}(k) + (L+G) \otimes I_m(\underline{1} \otimes \tilde{y}_0(k)). \tag{4.46}$$

Throughout this chapter it is assumed that $x_0 = \hat{x}_0$, implying $\tilde{y}_0 \equiv 0$, *i.e.* the control node state is accurately known to its neighbors. This means that pinned nodes have a fixed reference of 0, while others use relative information only. Then, define the *distributed neighborhood observer* dynamics

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + Bu_i(k) - c_1(1+d_i+g_i)^{-1} F\varepsilon_i^o(k). \tag{4.47}$$

where F is an observer gain matrix and c_1 a coupling gain. This algorithm has local neighborhood output disagreement (4.45) weighted by $(1+d_i+g_i)^{-1}$. The importance of using this weighting is shown in Lemma 4.1 and illustrated in Example 4.1. The global distributed observer dynamics then has the form

$$\hat{x}(k+1) = I_N \otimes A\hat{x}(k) + I_N \otimes Bu(k) + c_1(I+D+G)^{-1}(L+G) \otimes F\tilde{y}(k)$$

$$\hat{x}(k+1) = \left[I_N \otimes A - c_1 \left(I + D + G \right)^{-1} \left(L + G \right) \otimes FC \right] \hat{x}(k) + I_N \otimes Bu(k)$$
$$+ c_1 \left(I + D + G \right)^{-1} \left(L + G \right) \otimes Fy(k)$$

where $G = diag\left(g_1, \ldots, g_N \right)$ is the diagonal matrix of pinning gains.

Define the *i-th* state *observer error*

$$\eta_i(k) = x_i(k) - \hat{x}_i(k)$$

Then, the *global observer error* $\eta = \begin{bmatrix} \eta_1^T & \cdots & \eta_N^T \end{bmatrix}^T \in \mathbb{R}^{nN}$ has the dynamics

$$\eta(k+1) = \left[I_N \otimes A - c_1 \left(I + D + G \right)^{-1} \left(L + G \right) \otimes FC \right] \eta(k). \tag{4.48}$$

Denote the global observer system matrix as

$$A_o = I_N \otimes A - c_1 (I + D + G)^{-1} (L + G) \otimes FC. \tag{4.49}$$

The weighted graph matrix is

$$\Gamma = \left(I + D + G \right)^{-1} \left(L + G \right) \tag{4.50}$$

Lemma 4.7. *If there exists a spanning tree in the communication graph $\mathcal{G}$ with pinning into a root node, then $L + G$, and hence Γ, is nonsingular, and both have all their eigenvalues in the open right half plane.*

It is now desired to design the observer gains F to guarantee stability of the observer errors, hence convergence of the estimates $\hat{x}_i(k)$ to the true states $x_i(k)$. Unfortunately, the stability of the observer error dynamics depends on the graph topology through the eigenvalues $\Lambda_k, k = 1 \ldots N$ of graph matrix Γ as shown by the following result.

Lemma 4.8. *The global observer error dynamics (4.48) are asymptotically stable if and only if $\rho\left(A - c_1 \Lambda_k FC \right) < 1$ for all eigenvalues Λ_k, $k = 1 \ldots N$ of the graph eigenvalue matrix $\Gamma = \left(I + D + G \right)^{-1} \left(L + G \right)$.*

Proof. Perform on (4.48) a state-space transformation $z^o = (T \otimes I_n)\eta$, where T is a matrix satisfying $T^{-1}\Gamma T = \Lambda$ with Λ being a block triangular matrix. The transformed global observer error system is

$$z^o(k+1) = [I_N \otimes A - c_1\Lambda \otimes FC] z^o(k) \tag{4.51}$$

Therefore, (4.48) is stable if and only if $I_N \otimes A - c_1\Lambda \otimes FC$ is stable. This is equivalent to matrices $A - c_1\Lambda_k FC$ being stable for every k since Λ_k are the diagonal elements of matrix Λ. ∎

A necessary condition for the stability of (4.48) is detectability of (A, C). According to Lemma 4.7, stability of $A - c_1\Lambda_k FC$ implies detectability of (A, C) under the assumption that the graph has a spanning tree.

4.5.2 Convergence Region

The concept of the synchronization region introduced in Section 4.2.2 has proven important for the design of state feedback gains for continuous-time and discrete-time multi-agent systems that guarantee stability on arbitrary graph topologies. The next definition provides a dual concept for observer design.

Definition 4.3: *For a matrix pencil $A - sFC$, $s \in \mathbb{C}$, the convergence region is a subset $S_o \subseteq \mathbb{C}$ such that $S_o = \{s \in \mathbb{C} | \rho(A - sFC) < 1\}$.*

The convergence region may not be connected, but it is intimately related to the complex gain margin region defined next.

Definition 4.4: *The complex gain margin region $U_o \subseteq \mathbb{C}$ for some stabilizing observer gain F, given a system (A, C) is a simply connected region in $\mathbb{C}$, $U_o = \{s \in \mathbb{C} | \rho(A - sFC) < 1\}$ containing $s = 1$.*

Note that for a stabilizing observer gain F one has the relation $U_o \subseteq S_o$. The next result ties together Lemma 4.8 and the concepts introduced here.

Lemma 4.9. *Matrices $A - c_1\Lambda_k FC$ are stable for all eigenvalues Λ_k if and only if $c_1\Lambda_k \in S_o; \forall k$.*

4.5.3 Local Riccati Cooperative Observer Design

Given a single agent system (A, C) the convergence region depends on the local observer gain matrix F. Therefore, the distributed observer design problem is to find an F that yields a guaranteed convergence region or complex gain margin region. It will now be shown that selecting the local observer matrix F based on a local algebraic Riccati equation gives a guaranteed gain margin region. Then, it remains to check the relation of the graph matrix eigenvalues to the guaranteed gain margin region. Sufficient conditions for stable observer design are given in the next theorem.

Theorem 4.2. *Given multi-agent systems (4.43) with (A, C) detectable, assume the interaction graph contains a spanning tree with at least one pinning gain non-zero that connects into the root node. Let $P > 0$ be a solution of the discrete-time observer Riccati equation*

$$APA^T - P + Q - APC^T \left(CPC^T \right)^{-1} CPA^T = 0, \tag{4.52}$$

where $Q = Q^T > 0$. Choose the observer gain matrix as

$$F = APC^T \left(CPC^T \right)^{-1}. \tag{4.53}$$

Define

$$r_{obs} := \left[\sigma_{\max} \left(Q^{-1/2} APC^T \left(CPC^T \right)^{-1} CPA^T Q^{-1/2} \right) \right]^{-1/2}. \tag{4.54}$$

Then the observer error dynamics (4.48) are stable if there exists a covering circle $\bar{C}(c_0, r_0)$ of the graph matrix eigenvalues Λ_k, $k = 1...N$ such that

$$\frac{r_0}{c_0} < r_{obs}. \tag{4.55}$$

If (4.55) is satisfied then taking the coupling gain

$$c_1 = \frac{1}{c_0} \tag{4.56}$$

makes the observer dynamics (4.48) stable. ∎

The importance of Theorem 4.2 is that it decouples the local observer gain design at each agent from the details of the interconnecting communication graph structure. The observer gain is designed by each agent using the local Riccati equation (4.52), and then the graph topology comes into the choice of the coupling gain through condition (4.55). Unlike the continuous-time case discussed in Chapter 3, it is not always possible to select F and c_1 to guarantee synchronization on any graph with a spanning tree. Indeed, condition (4.55) must be satisfied. This condition is a relation between the properties of the communication graph and the instability properties of the local agent dynamics.

The local Riccati design method in Theorem 4.2 is scalable, since it only depends on the local agent dynamics, which have state space of dimension n, and does not depend on the size N of the graph.

The next technical lemma is needed in the proof of Theorem 4.2.

Lemma 4.10. *The convergence region S_o for the observer gain F given in Theorem 4.2 contains the circle $C\left(1, r_{obs}\right)$.*

Proof. The observer Riccati equation (4.52) can be written as

$$APA^T - P + Q - F\left(CPC^T\right)^{-1} F^T = 0. \tag{4.57}$$

Select any symmetric matrix $P > 0$. Then

$$
\begin{aligned}
&(A^T - sC^T F^T)^* P(A^T - sC^T F^T) - P \\
&= APA^T - P - s^* FCPA^T - sAPC^T F^T + s^* sFCPC^T F^T \\
&= APA^T - P - 2\operatorname{Re} sAPC^T (CPC^T)^{-1} CPA^T + |s|^2 APC^T (CPC^T)^{-1} CPA^T \\
&= APA^T - P - (2\operatorname{Re} s - |s|^2) FCPC^T F^T \\
&= APA^T - P - (1 - |s-1|^2) FCPC^T F^T
\end{aligned} \tag{4.58}
$$

A sufficient condition for the stability of $(A - sFC)$ is that this be less than zero. Inserting (4.57) this is equivalent to

$$-Q + |s-1|^2 FCPC^T F^T < 0 \tag{4.59}$$

This equation furnishes the bound

$$|s-1|^2 < \frac{1}{\sigma_{\max}\left(Q^{-1/2} F(CPC^T) F^T Q^{-1/2}\right)} \tag{4.60}$$

Furthermore expressing F as (4.53) one obtains

$$|s-1|^2 < \frac{1}{\sigma_{max}\left(Q^{-1/2}APC^T\left(CPC^T\right)^{-1}CPA^TQ^{-1/2}\right)} = r_{obs}^2 \qquad (4.61)$$

This is an open circle $C(1, r_{obs})$, and for s in that circle $\rho(A^T - sC^TF^T) < 1$, but this also means that $\rho(A - sFC) < 1$ since transposition does not change the eigenvalues of a matrix. ∎

The guaranteed circle of convergence $C(1, r_{obs})$, being simply connected in the complex plane and containing $s = 1$, is contained within the observer gain margin region U_o.

Proof of Theorem 4.2. Given $C(1, r_{obs})$ contained in the convergence region S_o of matrix pencil $A - sFC$, and the properties of dilation (homothety), assuming there exists a directed spanning tree in the graph with a nonzero pinning gain into the root node, it is clear that synchronization is guaranteed if all eigenvalues Λ_k are contained in a circle $C(c_0, r_0)$ similar with respect to homothety to a circle concentric with and contained within $C(1, r_{obs})$. Homothety as understood here, refers to a radial projection with respect to the origin, *i.e.* zero in complex plane. That is, it must be possible to bring $C(c_0, r_0)$ into $C(1, r_{obs})$ by scaling the radii of all graph eigenvalues by the same constant multiplier.

The center of the covering circle c_0 can be taken on the real axis due to symmetry and the radius equals $r_0 = \max_k |\Lambda_k - c_0|$. Taking these as given, it is straightforward that one should have

$$\frac{r_0}{c_0} < \frac{r_{obs}}{1}.$$

If this equation is satisfied then choosing $c = 1/c_0$ maps with homothety the covering circle of all eigenvalues $C(c_0, r_0)$ into a circle $C(1, r_0/c_0)$ concentric with and, for $r > r_0/c_0$, contained in the interior of the circle $C(1, r_{obs})$. ∎

Note that in (4.52) there is a condition $P > 0$, and so one must have C of full row rank. A necessary condition for the existence of a solution $P > 0$ to (4.52) is

detectability of (A, C). Also is should be remarked that if condition (4.55) is satisfied then the coupling gain has a robustness property just as detailed in Lemma 4.6.

For the sake of computational simplicity the covering circle $C(c_0, r_0)$ is used to prove sufficiency. If, however, this covering circle is found not to satisfy (4.55), there could be other covering circles satisfying this condition, *e.g.* those having smaller r_0/c_0 ratio.

4.6. Duality on Graphs for Discrete-time Cooperative Systems

In this section a duality property is given for cooperative controllers and cooperative observers on communication graphs, where $c_1 = c$. In this discussion, it is necessary to consider non-weighted controller and observer protocols given by

$$x_i(k+1) = Ax_i(k) + cBK\left[\sum_{j\in N_i} e_{ij}(x_j - x_i) + g_i(x_0 - x_i)\right](k), \qquad (4.62)$$

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + Bu_i(k) - c_1 F\left[\sum_j e_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(\tilde{y}_0 - \tilde{y}_i)\right](k). \qquad (4.63)$$

The next result details the duality property for the distributed controllers (4.62) and observers (4.63). For this result, it is required that the interaction graph of the observer be the same as the interaction graph for the control design, as assumed throughout this chapter. Recall that the reverse graph $\mathcal{G}'$ has nodes and edges the same as $\mathcal{G}$, but with all edge arrows reversed.

Theorem 4.3. *Consider a networked system of N identical linear agents on a communication graph $\mathcal{G}$ with dynamics (A, B, C) given by (4.43). Let the graph $\mathcal{G}$ be balanced. Suppose the feedback gain K in protocol (4.62) stabilizes the global tracking error dynamics having closed-loop system matrix*

$$A_c = I_N \otimes A - c(L+G) \otimes BK. \qquad (4.64)$$

Then the observer gain $F = K^T$ in protocol (4.63) stabilizes the global observer error dynamics (4.48) with the closed-loop observer matrix

$$A_o = I_N \otimes A^T - c(L+G)^T \otimes FB^T, \qquad (4.65)$$

for a networked dual system of N identical linear agents (A^T, C^T, B^T) on the reverse communication graph $\mathcal{G}'$.

Proof. Similar to the proof given in Section 3.5 for continuous-time systems. ∎

Though the duality result requires that weighting of $(1+d_i+g_i)^{-1}$ is not used in this section, the importance of weighting is illustrated in Example 4.1.

4.7. Cooperative Dynamic Regulators for Synchronization

This section introduces cooperative dynamic regulators that guarantee state synchronization by using knowledge of the local neighborhood output disagreement. Complete state information is not assumed, and observation, either distributed neighborhood or local, is used to estimate the local neighborhood state tracking errors (4.3). Three different observer-controller architectures at node i are presented, each of which yields a cooperative dynamic output feedback regulator that guarantees cooperative tracking.

The observer graph and controller graph do not need to be the same. In the case that the observer and controller graph are not the same the sufficient conditions of Theorem 4.1 and 4.2 apply separately to the respective graphs. The following results are derived under a reasonable simplifying assumption of having those two graphs equal, but apply similarly to the more general case. The development of this section is a discrete-time version of the continuous-time results of [28] 3.6. The cooperative regulators should have an observer at each node and a control law at each node based on the observed outputs. Also the separation principle should hold. This is the case for the following three system architectures.

4.7.1 Neighborhood Controller and Neighborhood Observer

In this architecture, the controller uses a distributed neighborhood feedback law

$$u_i = c\left(1+d_i+g_i\right)^{-1} K\hat{\varepsilon}_i, \qquad (4.66)$$

where

$$\hat{\varepsilon}_i = \sum_j e_{ij}\left(\hat{x}_j - \hat{x}_i\right) + g_i\left(\hat{x}_0 - \hat{x}_i\right) \qquad (4.67)$$

is the estimated local neighborhood tracking error. The observer is also a distributed neighborhood observer of the form (4.47). Thus, both the controller and the observer depend on the neighbor nodes.

The global state and estimate dynamics are given as

$$x(k+1) = (I_N \otimes A)x(k) - c_2(I+D+G)^{-1}(L+G) \otimes BK(\hat{x}-\overline{x}_0)(k) \quad (4.68)$$

$$\hat{x}(k+1) = \left[I_N \otimes A - c_1(I+D+G)^{-1}(L+G) \otimes FC \right]\hat{x}(k) + c_1(I+D+G)^{-1}(L+G) \otimes Fy(k) \quad (4.69)$$
$$- c_2(I+D+G)^{-1}(L+G) \otimes BK(\hat{x}-\overline{x}_0)(k)$$

Note that the assumption $\hat{x}_0 = x_0$ was used in (4.68). The global state error then follows from

$$x(k+1) = \left[I_N \otimes A - c_2(I+D+G)^{-1}(L+G) \otimes BK \right]x(k)$$
$$+ c_2(I+D+G)^{-1}(L+G) \otimes BK(\eta+\overline{x}_0)(k)$$

which yields the global error dynamics

$$\delta(k+1) = (I_N \otimes A - c_2(I+D+G)^{-1}(L+G) \otimes BK)\delta(k)$$
$$+ c_2(I+D+G)^{-1}(L+G) \otimes BK\eta(k) \qquad , \qquad (4.70)$$
$$= A_c\delta(k) + B_c\eta(k)$$

and

$$\eta(k+1) = (I_N \otimes A - c_1(I+D+G)^{-1}(L+G) \otimes FC)\eta(k) \qquad (4.71)$$
$$= A_o\eta(k)$$

The entire error system is then

$$\begin{bmatrix} \delta \\ \eta \end{bmatrix}(k+1) = \begin{bmatrix} A_c & B_c \\ 0 & A_o \end{bmatrix}\begin{bmatrix} \delta \\ \eta \end{bmatrix}(k). \qquad (4.72)$$

This system is asymptotically stable, implying $\delta, \eta \to 0$ asymptotically, under the conditions detailed in Theorem 4.1 and 4.2. This guarantees synchronization of all agents to the control node state.

4.7.2 Neighborhood Controller and Local Observer

In this architecture, the controller is of the distributed neighborhood form (4.66), (4.67). On the other hand, the observers are now local, depending only on the local agent dynamics according to

$$\hat{x}_i(k+1) = A\hat{x}_i(k) + Bu_i(k) + F\tilde{y}_i(k) \tag{4.73}$$

and do not use information from the neighbors. Then, the state observation error $\eta_i = x_i - \hat{x}_i$ dynamics is

$$\eta_i(k+1) = (A - FC)\eta_i(k),$$

or globally

$$\eta(k+1) = I_N \otimes (A - FC)\eta(k). \tag{4.74}$$

Then, the overall error dynamics are

$$\begin{bmatrix} \delta \\ \eta \end{bmatrix}(k+1) = \begin{bmatrix} A_c & B_c \\ 0 & I_N \otimes (A - FC) \end{bmatrix} \begin{bmatrix} \delta \\ \eta \end{bmatrix}(k). \tag{4.75}$$

Now, the observer gain F is simply selected using, for instance, Riccati design so that $(A - FC)$ is asymptotically stable. The feedback gain is selected by Theorem 4.1. Then, under the hypotheses of Theorem 4.1, synchronization is guaranteed.

4.7.3 Local Controller and Neighborhood Observer

In this architecture the controller is local of the form

$$u_i = -K\hat{x}_i, \tag{4.76}$$

which does not depend on the neighbors. On the other hand, the observer is distributed and given by protocol (4.47), with

$$\varepsilon_i^o = \sum_j e_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(y_0 - \tilde{y}_i).$$

Note that instead of $\tilde{y}_0$ in (4.45), which is assumed identically equal to zero, here one uses the control node output y_0. In global form this yields

$$x(k+1) = I_N \otimes Ax(k) - I_N \otimes BK\hat{x}(k)$$
$$\hat{x}(k+1) = I_N \otimes (A-BK)\hat{x}(k) + c_1\Gamma \otimes FC\eta(k) - c_1\Gamma \otimes FC\overline{x}_0(k) \tag{4.77}$$

Expressing $x, \hat{x}$ through $x, \eta = x - \hat{x}$ gives

$$x(k+1) = I_N \otimes (A-BK)x(k) + I_N \otimes BK\eta(k)$$
$$\eta(k+1) = I_N \otimes (A-BK)x(k) + I_N \otimes BK\eta(k) - I_N \otimes (A-BK)\hat{x}(k)$$
$$- c_1\Gamma \otimes FC\eta(k) + c_1\Gamma \otimes FC\overline{x}_0(k)$$
$$= I_N \otimes (A-BK)\eta(k) + I_N \otimes BK\eta(k) - c_1\Gamma \otimes FC(\eta - \overline{x}_0)(k) \tag{4.78}$$
$$= (I_N \otimes A - c_1\Gamma \otimes FC)\eta(k) + c_1\Gamma \otimes FC\overline{x}_0(k)$$
$$= A_o\eta(k) + c_1\Gamma \otimes FC\overline{x}_0(k)$$

Global tracking error dynamics now follows as

$$\delta(k+1) = (I_N \otimes A)\delta(k) - (I_N \otimes BK)\hat{x}(k). \tag{4.79}$$

Using that $\hat{x} = x - \eta = x - \overline{x}_0 - \eta + \overline{x}_0 = \delta - (\eta - \overline{x}_0)$ one finds

$$\delta(k+1) = I_N \otimes (A-BK)\delta(k) + (I_N \otimes BK)(\eta(k) - \overline{x}_0(k)). \tag{4.80}$$

Neither (4.78) nor (4.80) are autonomous systems since an exogenous input is present in form of control node state $\overline{x}_0(k)$. However if one looks at the dynamics of $\vartheta(k) = \eta(k) - \overline{x}_0(k)$ it follows that $\vartheta(k+1) = A_o\vartheta(k)$, and $\delta(k+1) = I_N \otimes (A-BK)\delta(k) + (I_N \otimes BK)\vartheta(k)$. Or, more clearly written in matrix form

$$\begin{bmatrix} \delta \\ \vartheta \end{bmatrix}(k+1) = \begin{bmatrix} I_N \otimes (A-BK) & I_N \otimes BK \\ 0 & A_o \end{bmatrix} \begin{bmatrix} \delta \\ \vartheta \end{bmatrix}(k). \tag{4.81}$$

The control gain K is simply selected using, *e.g.* Riccati design, so that $(A-BK)$ is asymptotically stable. The observer gain is selected by Theorem 4.2. Then, under the hypotheses of Theorem 4.2, synchronization $\delta \to 0$ is guaranteed.

Note that the observer in (46) is biased since $\vartheta \to 0$ implies $\eta \to \overline{x}_0$. Thus, the observers effectively estimate the tracking errors, converging to $\hat{x}_i(k) = x_i(k) - x_0(k) = \delta_i(k)$.

Remark 4.3. The three proposed observer-controller architectures differ in the amount of information that must be exchanged between neighbors for observation and control. When both the observer and controller are distributed, each agent requires a significant amount of computation and communication to produce the estimate of its own state, since neighbor outputs need to be measured and state, *i.e.* output, estimates need to be communicated between neighbors. Also the control input requires all the estimated neighbor states. The local observers and distributed controller architecture require less computations and communication, since the state of each agent is estimated using only its own inputs and outputs, and only the state estimates need to be communicated between neighbors. The third architecture using distributed estimation and local controllers is interesting because for control purposes the agents do not need to communication from their neighborhoods, though communication is needed for distributed estimation. The specific eigenvalues of $A - BK$, $A - FC$, A_o, A_c that appear in the augmented state equations (4.72), (4.75), (4.81) determine the time constants in each of the three architectures.

4.8. Simulation Examples

This section gives numerical examples that confirm the validity of the proposed control laws for distributed synchronization and observer convergence. A multi-agent system consisting of 5 identical discrete-time linear time invariant agents is considered. For these systems, feedback gains guaranteeing cooperative synchronization control and observer gains for state observation convergence are designed. The performance of the three proposed distributed dynamic compensators architectures in Section 4.7 is also investigated, with performance comparisons provided.

4.8.1 Example 4.1. Importance of Weighting in Control and Observer Protocols

This example shows the importance of using the weighting $\left(1 + d_i + g_i\right)^{-1}$ in control law (4.4) and observer algorithm (4.47). Consider a set of 5 agents with dynamics given by (4.43) with $\left(A, B, C\right)$ in controllable canonical form

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.2 & 0.2 & 1.1 \end{bmatrix}, \ B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \ C = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

The control node has the dynamics (4.44). The control node is pinned into two of the nodes, and the graph structure is given by $G = \mathrm{diag}(30, 0, 0, 0, 30)$ and

$$L = D - E = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 3 & 0 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

Figure 4.2a shows the non-weighted graph eigenvalues, that is, the eigenvalues of $(L + G)$, along with their covering circle. Figure 4.2b shows the magnified part of Figure 4.2a containing the weighted counterparts, that is the eigenvalues of $\Gamma = (I + D + G)^{-1}(L + G)$, their covering circle. It also shows the observer convergence region $C(1, r_{obs})$ displayed in dashes. Dashed lines marking a sector in complex plane depict the region wherein the eigenvalue covering circles should lie so that they be scalable (homothetic) to the observer convergence region. It can be verified that the non-weighted eigenvalues in Figure 4.2a do not satisfy the sufficient condition (4.14) given in Theorem 4.1, while the weighted eigenvalues in Figure 4.2b do.

It is interesting that the non-weighted eigenvalues are real, while the weighted eigenvalues here are complex.
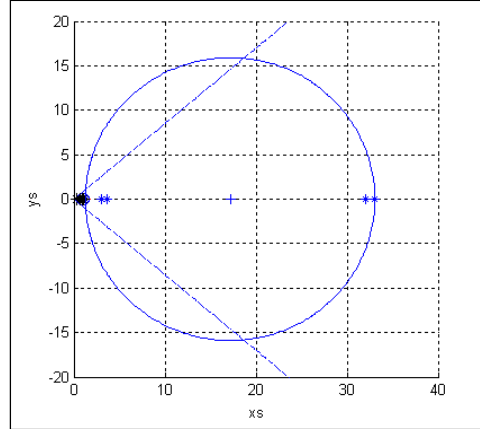
(a)

Fig. 4.2 (a). Non-weighted graph eigenvalues and their covering circle
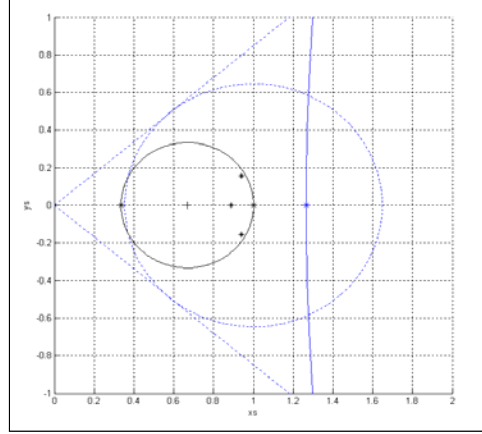
(b)

Fig. 4.2.(b). Weighted graph matrix eigenvalues, their covering circle, and convergence region of the observer (dashed circle).

4.8.2 *Example 4.2. Three Cooperative Dynamic Regulator Designs*

This example gives a set of simulations for the cooperative tracker using the three different controller-observer architectures in Section 4.7. The dynamics and the communication graph used are the same as in Example 4.1.

Example 4.2a. Cooperative Tracker with State Variable Feedback

In this simulation it is assume that perfect state measurements are available. That is, the control law (4.4) is used assuming full SVFB and the design of *c,K* is done using local Riccati design as in Theorem 4.1. . The Q matrix is chosen as $Q = 0.2I_3$. This provides a baseline for comparison for the performance of the three dynamic regulator designs given in Section 4.7. Figure 4.3 shows the state components of the 5 nodes and the leader control node. Figure 4.4 shows the first components of the state tracking errors. All nodes synchronize to the state of the leader.
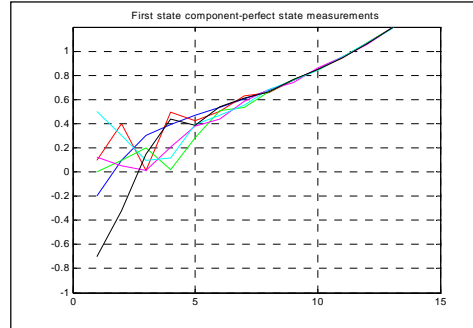
(a)
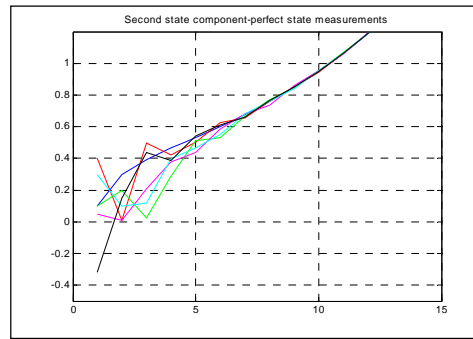
(b)
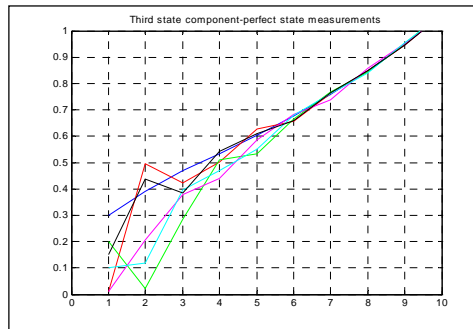
(c)

Fig. 4.3. States of the nodes and the control node with perfect state measurements (a) First state components, (b) Second state components, (c) Third state components.
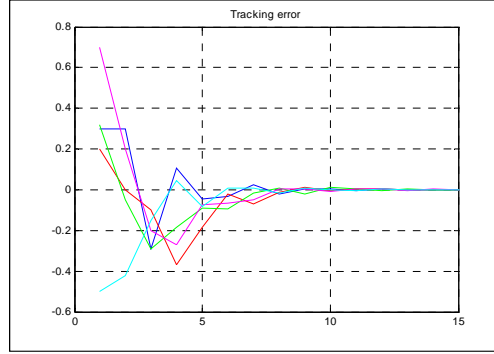
Fig. 4.4. First state tracking error component with perfect state measurements

Example 4.2b. Distributed Neighborhood Controllers and Observers

This simulation is for the case of distributed neighborhood controllers and observers given as in Section 4.7.1. Figure 4.5 depicts the first state components of all 5 agents and the control node, as well as the first components of the tracking errors δ and the first components of the observer errors η. Synchronization is achieved.
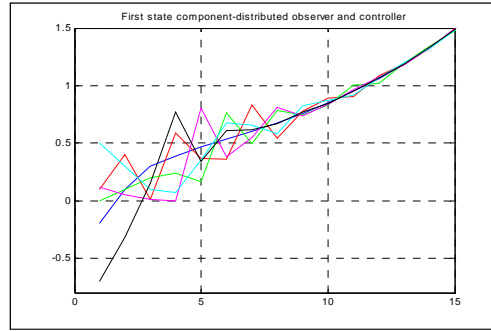
(a)
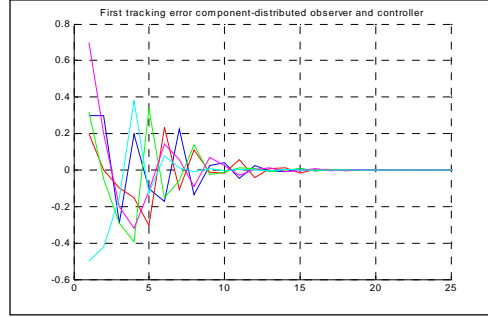
(b)
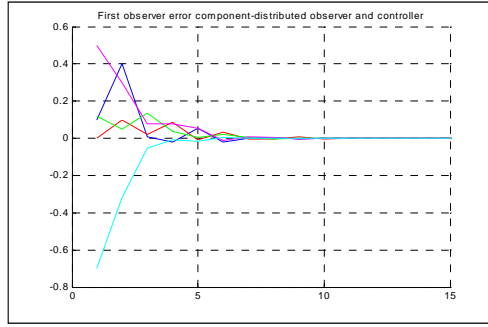
(c)

Fig. 4.5. Neighborhood controllers and observers. (a) First state components, showing synchronization, (b) First tracking error components, (c) First observer error components.

Example 4.2c. Neighborhood Controllers and Local Observers

This simulation is for the case of neighborhood controllers and local observers given in Section 4.7.2. Figure 4.6 depicts the first state components of all 5 agents and the control node, as well as the first components of the tracking errors δ and the first components of the observer errors η. Synchronization is achieved.
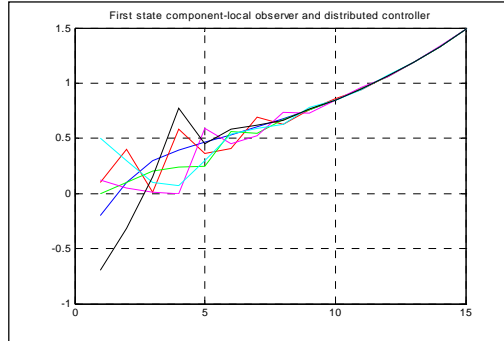
(a)
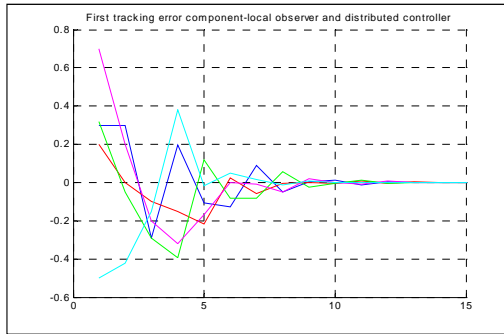
(b)
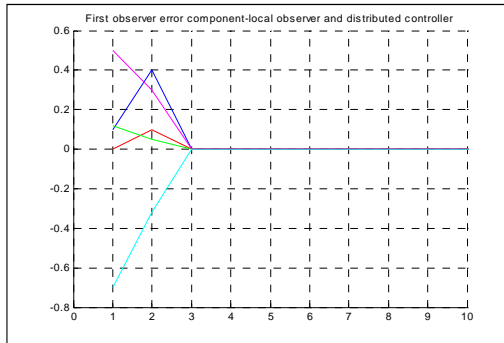
(c)

Fig. 4.6. Local observers and distributed controllers. (a) First
state components, showing synchronization, (b) First tracking error components,
(c) First observer error components.

Example 4.2d. Local Controllers and Neighborhood Observers

This simulation is for the case of local controllers and neighborhood observers given in Section 4.7.3. Figure 4.7 depicts the first state components of all 5 agents and the control node, as well as the first components of the tracking errors δ and the first components of the observer errors η. Synchronization is achieved.

Note that the observer errors in Figure 4.7c do not converge to zero since the observers estimate the tracking errors, and so converge to $\hat{x}_i(k) = x_i(k) - x_0(k) = \delta_i(k)$.
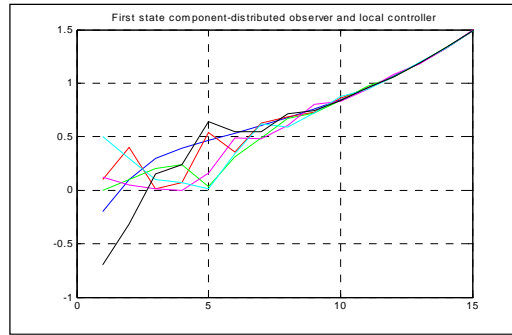
(a)
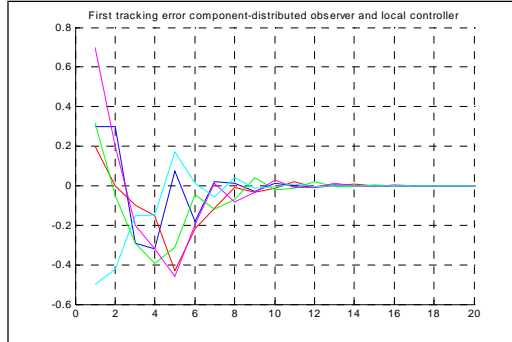
(b)
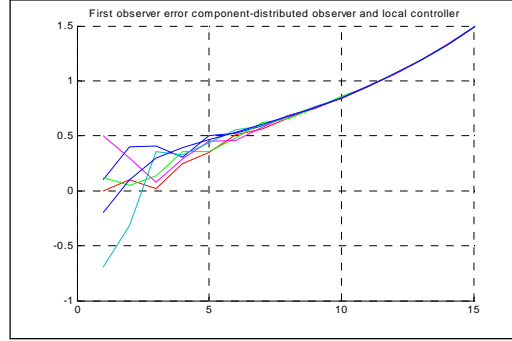
(c)

Fig. 4.7. Local controllers and neighborhood observers. (a) First
state components, showing synchronization, (b) First tracking error components,
(c) First observer error components.

Remark 4.4. It is interesting to compare the time constants of state synchroniza-
tion and estimate convergence for the three dynamic regulator architectures.
These are in accordance with Remark 4.3. Compared to perfect state measure-
ment, distributed observers and controllers take almost twice as long for state syn-
chronization, and almost that same for estimate convergence alone. The case of
neighborhood controller and local observer required the shortest time for state
synchronization, shorter even than in case of local controller and neighborhood
observer. For the system considered, it appears to be more advantageous to make
observation faster by using a local observer, than to simplify the control law. De-
pending on the specific eigenvalues generated from the choice of K and F, per-
haps for some other system it would pay more to use a local controller with dis-
tributed observer.

4.9. Conclusion

This chapter provides conditions for achieving synchronization of identical dis-
crete-time state space agents on a directed communication graph structure based
on local agent Riccati equation design. The concept of the discrete-time synchro-
nization region in the z-plane is used. The result is expressed in terms of a radius
easily computed from the local H_2 type Riccati equation solution given in Theo-
rem 4.1. It is shown in the single-input case that this approach yields known re-
sults in terms of the Mahler measure and graph condition number. In addition, lo-
cal Riccati design algorithms that do not depend on the topological properties of
the graph were given in Theorem 4.2 to design observer gains for cooperative sys-

tems on communication graphs. Riccati design for the controllers and observers guarantees stability if a condition in terms of the graph eigenvalues and stability properties of the local agent dynamics holds. A duality principle for distributed systems on graphs was given. Three cooperative dynamic regulator designs were presented, each using distributed feedback controllers and observers that use only local agent information available in the graph. All three guarantee synchronization of multi-agent systems to the state of a control node using only output feedback. Furthermore, the separation principle is shown to be valid in all three cases for multi-agent systems on graphs. Results presented in this chapter could be extended to time-varying graphs using results available in the literature.

References

[1]   Duan Z, Chen G, Huang L. Disconnected Synchronized Regions of Complex Dynamical Networks. *IEEE Transactions on Automatic Control* 2009; 54, (4): 845-849. DOI: 10.1109/TAC.2008.2009690

[2]   Elia N, Mitter S.K. Stabilization of Linear Systems With Limited Information. *IEEE Transactions on Automatic Control* 2001; 46, (9): 1384-1400. DOI: 10.1109/9.948466

[3]   Fax J, Murray R. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control* 2004; 49, (9): 1465-1476. DOI: 10.1109/TAC.2004.834433

[4]   Fu M, Xie L. The Sector Bound Approach to Quantized Feedback Control. *IEEE Transactions on Automatic Control.* 2005; 50, (11): 1698-1711. DOI: 10.1109/TAC.2005.858689

[5]   Hengster-Movric K, You K, Lewis F.L, Xie L. Synchronization of Discrete-time Multi-agent Systems on graphs using Riccati Design. Automatica, vol. 49, no. 2, pp. 414-423, Feb. 2013.

[6]   Hong Y, Wang X, Jiang Z.P. Multi-Agent Coordination with General Linear Models: A Distributed Output Regulation Approach. *8th IEEE International conference on Control and Automation (ICCA)* 2010; 137-142. DOI: 10.1109/ICCA.2010.5524157

[7]   Jadbabaie A, Lin J, Morse A. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control* 2003; 48, (6): 988–1001. DOI: 10.1109/TAC.2003.812781

[8]   Lewis, F.L., D. Vrabie, and V.L. Syrmos, Optimal Control, 3rd Edition, Wiley, New York, 2012.

[9]   Li Z, Duan Z, Chen G, Huang L. Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint. *IEEE Transactions on Circuits and Systems I, Reg. Papers* 2010; 57, (1): 213–224. DOI: 10.1109/TCSI.2009.2023937

[10]   Li, Z., Duan, Z., Chen, G. Consensus Of Discrete-time Linear Multi-agent Systems With Observer-Type Protocols. *Discrete And Continuous Dynamical Systems*, *Series B,* 2011; 16 (2), pp. 489-505.

[11]   Liu S, Xie L, Lewis F. Synchronization of Multi-Agent Systems with Delayed Control Input Information from Neighbors. *49th IEEE Conference on Decision and Control (CDC)* 2010; 4529 – 4534. DOI: 10.1109/CDC.2010.5716980

[12]   Ma CQ, Zhang JF. Necessary and Sufficient Conditions for Consensusability of Linear Multi-Agent Systems. *IEEE Transactions on Automatic Control* 2010; 55 (5): 1263-1268. DOI: 10.1109/TAC.2010.2042764

[13]   Olfati-Saber R, Fax J.A, Murray R.M. Consensus and Cooperation in Networked Multi-Agent Systems. *Proceedings of the IEEE* 2007; 95 (1): 215-233. DOI: 10.1109/JPROC.2006.887293

[14]   Olfati-Saber R, Murray R.M. Consensus protocols for networks of dynamic agents. *Proceedings of American Control Conference* 2003; 951-956.

[15]   Olfati-Saber R, Murray R.M, Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control* 2004; 49 (9): 1520-1533. DOI: 10.1109/TAC.2004.834113

[16]   Pecora L.M, Carroll T.L. Master Stability Functions for Synchronized Coupled Systems. *Physical Review Letters* 1998; 80 (10): 2109-2112. DOI: 10.1103/PhysRevLett.80.2109

[17]   Qu Z. *Cooperative Control of Dynamical Systems, Applications to Autonomous Vehicles*. Springer-Verlag: London, 2009; 158-173.

[18]   Ren W, Beard R.W. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control* 2005; 50 (5): 655–661. DOI: 10.1109/TAC.2005.846556

[19]   Seo J.H, Shim H, Back J. Consensus of high-order linear systems using dynamic output feedback compensator: low gain approach. *Automatica* 2009; 45, (11): 2659-2664. DOI: 10.1016/J.AUTOMATICA.2009.07.022

[20]   Sinopoli, B., Schenato, L., Franceschetti, M., Poolla K., and Sastry S., "Kalman filtering with intermittent observations", *IEEE Transactions on Automatic Control*, vol. 49, No.9, pp.1453-1464, 2004.

[21]   Tsitsiklis, J. (1984). Problems in Decentralized Decision Making and Computation. Ph.D. dissertation, Dept. Elect. Eng. and Comput. Sci., MIT, Cambridge, MA.

[22]   Tuna S.E. LQR-based coupling gain for synchronization of linear systems. *arXiv:0801.3390v1 [math.OC]* 2008; 22 Jan.

[23]   Wang X.F, Chen G. Pinning control of scale free dynamical networks. *Physica A* 2002; 310, (3-4): 521-531.

[24]   You K, Xie L. Necessary and Sufficient Conditions for Consensusability of Discrete-time Multi-agent Systems. *2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems* 2010; DOI: 10.3182/20100913-2-FR-4014.00019

[25]   You K, Xie L. Consensusability of Discrete-time Multi-agent Systems over Directed Graphs. *5th AOTULE International Postgraduate Students Conference on Engineering* 2010.

[26]   You, K, Xie L. "Network Topology and Communication Data Rate for Consensusability of Discrete-time Multi-agent Systems", *IEEE Transactions on Automatic Control,* 2011; 56 (10), pp. 2262-2275, 2011

[27]   You K, Xie L. Coordination of Discrete-time Multi-agent Systems via Relative Output Feedback. *International journal of Robust and Nonlinear control* 2011. 21: n/a. DOI: 10.1002/rnc.1659

[28]   Zhang H, Lewis F.L, Das A. Optimal Design for Synchronization of Cooperative Systems: State Feedback, Observer and Output Feedback. *IEEE Transactions on Automatic Control* 2011; 56 (6): 1-1. DOI: 10.1109/TAC.2011.2139510

5
Cooperative Globally Optimal Control for Multi-agent Systems on Directed Graph Topologies

In Chapters 3 and 4 we showed that locally optimal design in terms of Riccati equations can guarantee synchronization for cooperative multi-agents on graphs. In this chapter, we examine the design of distributed control protocols that solve *global* optimal problems for all the agents in the graph. In cooperative control systems on graphs it turns out that local optimality for each agent and global optimality for the all agents are not the same. This chapter brings together stability and optimality theory to design distributed cooperative control protocols that guarantee synchronization and are also optimal with respect to a positive semi-idefinite global performance criterion. A common problem in optimal decentralized control is that global optimization problems generally require global information from all the agents, which is not available to distributed controllers. In cooperative control of multi-agent systems on graphs, each agent is only allowed to use distributed information that respects the graph topology, that is, information about itself and its neighbors. Global optimal control for multi-agent systems is complicated by the fact that the communication graph topology interplays with agent system dynamics.

In this chapter we use an inverse optimality approach together with partial stability to consider synchronization with globally optimal performance. Agents are assumed identical with linear time-invariant dynamics. Directed communication graphs of fixed topology are considered. Specially structured quadratic performance indices are derived that capture the topology of the graph, allowing for globally optimal controllers that can be implemented using local distributed protocols.

A new class of digraphs is defined on which global optimal controllers exist that are distributed in form- namely, those graphs whose Laplacian matrix has a diagonal Jordan form. Such graphs are shown to admit distributed controllers that solve global optimality problems for all agents on the entire graph. This structural constraint can in fact be relaxed, and optimal cooperative controllers developed for arbitrary digraphs, by allowing state-control cross-weighting terms in the performance criterion.

The results in this chapter come from [8].

5.1. Stability, Local Optimality, and Global Optimality for Synchronization Control on Graphs

The relations between stability and optimality have long been debated in the community and are now understood for single-agent systems. However, relations of stability and optimality in multi-agent cooperative control systems are only now beginning to be clarified. There are more intriguing relations between stability and optimality in cooperative control than appear in the single-agent case, since local stability and global team stability are not the same, and local agent optimality and global team optimality are not the same. New phenomena appear that are not present for single-agent systems. Moreover, throughout everything synchronization of the states of all agents must be guaranteed.

In cooperative multi-agent control on graphs, any admissible control protocols must be distributed in the sense that they use only information locally available to an agent on the prescribed graph topology. Specifically, distributed control protocols are allowed to depend only on information about the agent and its direct neighbors. The communication restrictions imposed by graph topologies can severely limit what can be accomplished by local distributed control protocols at each agent. Chapters 3 and 4 showed that local optimal design of distributed protocols at each agent can guarantee global stability, or synchronization, in multi-agent systems on communication graphs.

In this chapter we consider the global optimality of local distributed control protocols in terms of the overall performance of the team of all agents. It is seen that globally optimal performance cannot in general be guaranteed for standard LQR performance measures by using distributed control protocols for arbitrary digraphs. For linear distributed agent control protocols to yield global optimal team behaviors, the graph topology must satisfy certain conditions. These conditions are shown to define a new class of digraphs that admit global optimal control design using distributed control protocols. If these conditions are satisfied, moreover, any admissible global performance indices must be selected in a certain manner that depends on the graph topology. This clarifies the interactions between stability, optimal performance, and the communication graph topology in multi-agent systems.

Cooperative optimal control has been considered by many authors-[5],[4],[3],[21],[2],[18], to name just a few. Optimality of a control protocol gives rise to desirable characteristics, such as gain and phase margins, that guarantee robustness in presence of some types of disturbances [11],[7]. The common difficulty, however, is that in the general case optimal control is not distributed [4],[3]. Solution of a global optimization problem generally requires centralized, *i.e.* global, information. In order to have local distributed control that is optimal in some sense it is possible *e.g.* to consider each agent optimizing its own, local, performance index. This is done for receding horizon control in [5], implicitly in [27], and for distributed games on graphs in [25], where the notion of optimality is Nash equilibrium. In [21] the LQR problem is phrased as a maximization problem

of LMI's under the constraint of the communication graph topology. This is a constrained optimization taking into account the local character of interactions among agents. It is also possible to use a local observer to obtain the global information needed for the solution of the global optimal problem, as is done in [4]. In the case of agents with identical linear time-invariant dynamics, [2] presents a suboptimal design that is distributed on the graph topology.

In this chapter are considered fixed topology directed graphs and linear time-invariant agent dynamics. A new class of digraphs is defined, namely those whose Laplacian matrix is simple, *i.e.* has a diagonal Jordan form. On these graphs the globally optimal LQR problem has a distributed linear protocol solution. If this condition is satisfied, then distributed linear protocols exist that solve the global optimal LQR problem only if the performance indices are of a certain form that captures the topology of the graph. That is, the achievable optimal performance depends on the graph topology.

The structure of the chapter is the following; Section 5.2 introduces graph theory concepts and some definitions that are used throughout the chapter. More information about graphs appears in Chapter 2. Section 5.3 deals with stability of possibly noncompact invariant consensus manifolds, and defines the notion partial stability. Section 5.4 introduces conditions of optimality and inverse optimality suitable for the study of synchronization on graphs where the consensus manifold may not be compact. These results are applied in Sections 5.5 and 5.6. Section 5.5 discuses globally optimal leaderless consensus and then pinning control to a leader node for single-integrator agent dynamics. We call these respectively the cooperative regulator and the cooperative tracker problems. Section 5.6 gives global optimality results for general linear time-invariant agent dynamics.

In Sections 5.5 and 5.6 it is found that optimality for the standard LQR performance index is only possible for graphs whose Laplacian matrix satisfies a certain condition. In Section 5.7 this condition is further discussed and shown to be satisfied by certain classes of digraphs, specifically for those graphs whose Laplacian matrix has a diagonal Jordan form. This condition holds in particular for undirected graphs.

In Section 5.8 a more general performance index is considered which has state-control cross-weighting. It is shown that, with proper choice of the performance index, optimal cooperative control of distributed form can be achieved for any directed graph. This shows that performance indices with state-control cross-weighting terms are in some sense more natural for cooperative control on communication graphs.

5.2.   Graph Definitions

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N vertices $\mathcal{V} = \{v_1, \cdots, v_N\}$ and a set of edges or arcs $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. It is assumed that the graph is simple, *i.e.* there are no repeated

edges or self-loops $(v_i, v_i) \notin \mathcal{E}, \forall i$. Directed graphs are considered. Denote the connectivity matrix as $E = [e_{ij}]$ with $e_{ij} > 0$ if $(v_j, v_i) \in \mathcal{E}$ and $e_{ij} = 0$ otherwise. The set of neighbors of node v_i is $\mathcal{N}_i = \{v_j : (v_j, v_i) \in \mathcal{E}\}$, *i.e.* the set of nodes with arcs incoming into v_i. Define the in-degree matrix as $D = diag(d_1 \ldots d_N)$ with $d_i = \sum_j e_{ij}$ the (weighted) in-degree of node i (*i.e.* the *i-th* row sum of E). Define the graph Laplacian matrix as $L = D - E$. A *path* from node v_{i_1} to node v_{i_k} is a sequence of edges $\left(v_{i_1}, v_{i_2}\right), \left(v_{i_2}, v_{i_3}\right), \ldots \left(v_{i_{k-1}}, v_{i_k}\right)$, with $\left(v_{i_{j-1}}, v_{i_j}\right) \in \mathcal{E}$ or $\left(v_{i_j}, v_{i_{j-1}}\right) \in \mathcal{E}$ for $j = \{2, \ldots, k\}$. A *directed path* is a sequence of edges $\left(v_{i_1}, v_{i_2}\right), \left(v_{i_2}, v_{i_3}\right), \ldots \left(v_{i_{k-1}}, v_{i_k}\right)$, with $\left(v_{i_{j-1}}, v_{i_j}\right) \in \mathcal{E}$ for $j = \{2, \ldots, k\}$. The graph is said to be *connected* if every two vertices can be joined by a path. A graph is said to be *strongly connected* if every two vertices can be joined by a directed path. The graph is said to contain a (directed) *spanning tree* if there exists a vertex such that every other vertex in $\mathcal{V}$ can be connected by a (directed) path starting from it. Such a special vertex is then called a root node. The Laplacian matrix L has a simple zero eigenvalue if and only if the undirected graph is connected, or the directed graph contains a spanning tree.

A *bidirectional* graph is a graph satisfying $e_{ij} > 0 \Leftrightarrow e_{ji} > 0$. A *detail balanced graph* is a graph satisfying $\lambda_i e_{ij} = \lambda_j e_{ji}$ for some positive constants $\lambda_1 \ldots \lambda_N$. By summing over the index i it is seen that then $\begin{bmatrix} \lambda_1 & \ldots & \lambda_N \end{bmatrix}$ is a left eigenvector for the zero eigenvalue of L. The Laplacian matrix of a detail balanced graph satisfies $L = \Lambda P$, for some diagonal matrix $\Lambda > 0$ and $P = P^T$ a Laplacian matrix of some undirected graph. That is, the Laplacian L is symmetrizable. In fact $\Lambda = diag(1/\lambda_i)$ works. The concept of a detail balanced graph is related to the reversibility of an associated Markov process.

For any matrix A, $\sigma_{\min}(A), \sigma_{\max}(A)$ are minimal and maximal singular values of A respectively. For a positive semidefinite matrix A, $\sigma_{>0\min}(A)$ denotes the minimal nonzero singular value. Note that for symmetric L, $\sigma_{>0\min}(L) = \lambda_2(L)$, the graph Fiedler eigenvalue.

5.3. Partial Asymptotic Stability

This section presents concepts of partial stability applied to noncompact manifolds. The definitions are based on neighborhoods of a manifold and avoid the need for global coordinates, in contrast to usual formulation of partial stability,

e.g. [7]. The fact that a manifold to which convergence is to be guaranteed is noncompact means that usual approaches for compact manifolds based on proper Lyapunov functions are not applicable as such. Furthermore when dealing with noncompact manifolds one makes the distinction between non-uniform and uniform stability.

Let the system dynamics be given as

$$\dot{x} = f(x,u) = f(x) + g(x)u \,, \tag{5.1}$$

where $f(x)$, $g(x)$, and $u(t)$ are assumed to be such that the existence of a unique solution to the initial value problem is guaranteed.

Definition 5.1. *A neighborhood $\mathcal{D}(S)$ of a manifold S is an open set in an embedding space containing the manifold S in its interior.*

Definition 5.2. *An ε -neighborhood of a manifold $S \subset \mathcal{D}(S)$ is defined as $\mathcal{U}_\varepsilon(S) = \{x \in \mathcal{D}(S) | d(x,S) < \varepsilon\}$, where $d(x,S) := \inf_{y \in S} d(x,y)$ is the distance of a point from the manifold S as given by the distance function d of the embedding space.*

Note that in the case of compact manifolds any neighborhood $\mathcal{D}(S)$ contains some ε -neighborhood, but in the case of noncompact manifolds this need not be true. For the needs of defining stability of noncompact manifolds, one uses neighborhoods that contain some ε -neighborhood. We call such neighborhoods *regular*.

Definition 5.3: *A manifold S is said to be (Lyapunov) stable if there exists a regular neighborhood $\mathcal{D}(S)$, $S \subset \mathcal{D}(S)$, such that for every ε -neighborhood $\mathcal{U}_\varepsilon(S)$ contained in it there exists a subneighborhood $\mathcal{V}(S)$ satisfying the property $x(0) \in \mathcal{V}(S) \Rightarrow x(t) \in \mathcal{U}_\varepsilon(S)$ $\forall t \geq 0$. If $\mathcal{D}(S)$ can be taken as the entire state space embedding the manifold S, then the stability is global.*

If a manifold is Lyapunov stable and furthermore there exists a neighborhood $\mathcal{W}(S)$ satisfying the property $x(0) \in \mathcal{W}(S) \Rightarrow d(x(t),S) \to 0$ as $t \to \infty$, then the manifold S is *asymptotically stable*.

If a manifold is Lyapunov stable and for every ε -neighborhood $\mathcal{U}_\varepsilon(S)$, a pertaining subneighborhood $\mathcal{V}(S) \subseteq \mathcal{U}_\varepsilon(S)$ contains a δ-neighborhood $\mathcal{V}_\delta(S)$ then it is *uniformly stable*. And the stability conclusion can be phrased as $\forall \varepsilon > 0$ $\exists \delta > 0$ such that $d(x(0),S) < \delta \Rightarrow d(x(t),S) < \varepsilon$ $\forall t \geq 0$.

If a manifold is uniformly stable and asymptotically stable so that a neighborhood $\mathcal{W}(S)$ contains a δ-neighborhood $\mathcal{W}_\delta(S)$ satisfying the property $x(0) \in \mathcal{W}_\delta(S) \Rightarrow d(x(t), S) \to 0$ as $t \to \infty$, uniformly $\forall\, x(0) \in \mathcal{W}_\delta(S)$, then it is *uniformly asymptotically stable*.

If a manifold is uniformly asymptotically stable and there exist constants $K, \sigma > 0$ such that $d(x(t), S) \leq K d(x(0), S) e^{-\sigma t}$, for $x(0)$ in some δ-neighborhood $\mathcal{W}_\delta(S)$ then the stability is *exponential*. ∎

A sufficient condition for the various types of stability of a manifold, possibly noncompact, is given as the following theorem, motivated by [7], Chap 4. See that reference for the basic definitions used here.

Theorem 5.1. Partial Stability. *Given a manifold S, possibly noncompact, contained in a neighborhood $\mathcal{D} \supset S$, if there exists a C^1 function $V : \mathcal{D} \to \mathbb{R}$ and a class $\mathcal{K}$ function α such that*

$$V(x) = 0 \Leftrightarrow x \in S$$
$$V(x) \geq \alpha(d(x, S)) \ \text{ for } \ x \in \mathcal{D} \setminus S$$
$$\dot{V}(x) \leq 0 \ \text{ for } \ x \in \mathcal{D}$$

then S is Lyapunov stable. If furthermore there is a class $\mathcal{K}$ function β such that

$$\alpha(d(x, S)) \leq V(x) \leq \beta(d(x, S))$$

then the stability is uniform. If in addition there is a class $\mathcal{K}$ function γ such that

$$\dot{V}(x) < -\gamma(d(x, S))$$

then the stability is uniformly asymptotic. If there exist $a, b, c, p > 0$ such that

$$a d(x, S)^p \leq V(x) \leq b d(x, S)^p$$
$$\dot{V}(x) < -c d(x, S)^p$$

then the stability is exponential.

Proof. Take any $\mathcal{U}_\varepsilon(S)$, then on the boundary $\partial \mathcal{U}_\varepsilon(S)$ one has $d(x, S) = \varepsilon$ yielding a bound $V(x) \geq \alpha(\varepsilon)$. Taking $\eta = \alpha(\varepsilon)$ and defining $\mathcal{D}_\eta = \{x \in \mathcal{D}(S) \,|\, V(x) < \eta\}$ one has for $x \in \mathcal{D}_\eta$ that $\alpha(d(x, S)) < \eta$ meaning that $d(x, S) < \varepsilon$ which implies

that $\mathcal{D}_\eta \subseteq \mathcal{U}_\varepsilon(S)$. Taking $x(0) \in \mathcal{D}_\eta$, owing to the nonincreasing property of $V(x)$, one obtains

$$\alpha(d(x(t), S)) \leq V(x(t)) \leq V(x(0)) < \alpha(\varepsilon), \ \ \forall t \geq 0$$

This guarantees $x(t) \in \mathcal{U}_\varepsilon(S) \ \ \forall t \geq 0$, proving the Lyapunov stability of S.

However there is no guarantee that, due to noncompactness, one does not have $\inf_{x \in \partial \mathcal{D}_\eta} d(x, S) = 0$ which means that the η level set of $V(x)$ is not bounded away from the manifold S. This is remedied by the uniformity requirement since then one can choose $\delta > 0$ such that $\beta(\delta) = \alpha(\varepsilon)$, and by the previous argument for $x(0) \in \mathcal{V}_\delta(S)$

$$\alpha(d(x(t), S)) \leq V(x(t)) \leq V(x(0)) < \beta(\delta) = \alpha(\varepsilon),$$

Asymptotic and exponential stability follow from the proof analogous to partial stability results in [7]. ∎

The purpose of class $\mathcal{K}$ functions is to bind the level sets of the Lyapunov function away from the manifold and away from infinity, allowing the interpretation of a value of the Lyapunov function as a measure of the distance from the manifold. For global conclusions one would require $\mathcal{K}_\infty$, as an analog of properness, *i.e.* radial unboundedness.

Remark 5.1. Note that the uniformity property is always found in the case where the manifold S is compact. However since we are dealing here with a noncompact manifolds, the difference needs to be emphasized. Theorem 5.1 will be used together with the inverse optimality approach of the next section, to guarantee asymptotic stabilization to a consensus manifold that is also optimal.

Remark 5.2. In case the partial stability conditions are satisfied one does not necessarily have bounded trajectories. Trajectories are constrained to a noncompact set, and finite escape to infinity, while remaining in the noncompact neighborhood of S, is thereby not precluded. Therefore, in order to avoid such an occurrence one needs to independently impose that solutions can be extended for all time. Since linear systems are globally Lipschitz solutions cannot escape to infinity in finite time. However in more general cases, with nonlinear dynamics, an additional, proper, Lyapunov function could be used to guarantee boundedness of trajectories to a compact set. Note that this does not contradict the stability of a noncom-

pact set, since La Salle invariance principle can be used to guarantee convergence to (a subset of) such a set.

Remark 5.3. The stability properties presented in this coordinate free topological setting are based on properties of neighborhoods of a manifold S. These properties are not preserved by general homeomorphisms or even diffeomorphisms, since it is easy to construct a diffeomorphism that maps a regular neighborhood onto one that is not regular. In fact, the convergence of trajectories can be changed by a proper choice of a mapping. The sufficient condition for a homeomorphism h to preserve the stability properties of a manifold S, *i.e.* to preserve the regularity of neighborhoods $\mathcal{U}(S)$ and convergence of trajectories, is that there exist class $\mathcal{K}$ functions α, β such that

$$\alpha(d(x, S)) \leq d(h(x), h(S)) \leq \beta(d(x, S))$$

on the domain of h. This is fulfilled in particular by homeomorphisms that are uniformly continuous and have uniformly continuous inverse, although this condition is not necessary.

Note that the requirements on the Lyapunov function specialize in case of compact manifolds to familiar conditions [7]. In that case all bounded open neighborhoods are precompact. Specifically when one considers an equilibrium point the conditions specialize to

$$V(x) = 0 \Leftrightarrow x = 0, V(x) \geq \alpha(\|x\|) \text{ for } x \in \mathcal{D} \setminus \{0\}, \dot{V} \leq 0 \text{ for } x \in \mathcal{D}.$$

In case of linear systems and pertaining quadratic partial stability Lyapunov functions $V(x) = x^T P x \geq 0$ the target manifold is the null space of the positive semidefinite matrix P. Also the uniformity condition is automatically satisfied since $\sigma_{\min > 0} \|y\|^2 \leq y^T P y \leq \sigma_{\max} \|y\|^2$, where $y \in \ker P^\perp$, and $\|y\|$ serves as the distance from the null space.

5.4. Inverse Optimal Control

This section presents inverse optimality notions for specific application to the consensus problem, where the consensus manifold may be noncompact. Inverse optimality is a property of a stabilizing control $u = \phi(x)$, *e.g.* guaranteed by a control Lyapunov function $V(x)$, that is optimal with respect to some positive (semi) definite performance index. Let the system be given by the affine-in-control form

$$\dot{x} = f(x,u) = f(x) + g(x)u \,. \tag{5.2}$$

Certain results of optimal control can be explicitly derived for systems in this form, *e.g.* Hamilton-Jacobi-Bellman equation.

5.4.1 Optimality

The following lemma details the conditions for optimality of a stabilizing control under an infinite horizon criterion [7].

Lemma 5.1. Optimality. *Consider the control affine system (5.2). Let S be a target manifold, possibly noncompact. Given the infinite horizon optimality criterion*

$$J(x_0,u) = \int_0^\infty \mathcal{L}(x,u)dt \tag{5.3}$$

If there exist functions $V(x)$, $\phi(x)$, and class $\mathcal{K}$ functions α, γ satisfying the following conditions

$$V(x) = 0 \Leftrightarrow x \in S$$
$$V(x) \geq \alpha(d(x,S))$$
$$\phi(x) = 0 \Leftarrow x \in S$$
$$\nabla V(x) f(x,\phi(x)) \leq -\gamma(d(x,S))$$
$$H(x,\phi(x)) = 0$$
$$H(x,u) \geq 0$$

with $H(x,u) = \mathcal{L}(x,u) + \nabla V(x)^T f(x,u)$. Then the feedback control $u = \phi(x)$ is optimal with respect to the performance index (5.3) and asymptotically stabilizing with respect to the target set S. Furthermore the optimal value of the performance index is

$$J(x_0,\phi(x)) = V(x(0))_{\displaystyle .} \qquad\qquad\qquad \blacksquare$$

Conditions 1, 2 and 4 show that $V(x)$ is a control Lyapunov function for the closed-loop system guaranteeing asymptotic stability of the set S, by Theorem 5.1. If S is taken as a point, *i.e.* a compact manifold of dimension zero, then those conditions become the familiar conditions on the Lyapunov function stabilizing an equilibrium point

$$V(0) = 0$$

$$V(x) \geq \alpha(\|x\|) \quad x \neq 0$$

$$\phi(0) = 0$$

$$\nabla V(x)^T f(x, \phi(x)) < 0$$
.

5.4.2   *Inverse Optimality*

If the feedback control $u = \phi(x)$ is asymptotically partially stabilizing, then there exists a partial stability Lyapunov function $V(x)$ satisfying the conditions 1, 2 and 4 of Lemma 5.1 by the inverse Lyapunov theorems [7],[22].

In inverse optimality settings an asymptotically stabilizing control law $u = \phi(x)$ is given. Then the performance integrand $\mathcal{L}(x,u)$ and Lyapunov function $V(x)$ are to be determined. In this chapter we are concerned with nonnegative performance index integrands, so this is to be contrasted with the more general inverse optimality where the performance integrand need not be nonnegative, as long as it guarantees a stabilizing control. That the performance integrand must be positive (semi) definite imposes constraints on $V(x)$.

The next result formulates inverse optimality in a special form required for the proofs of our main results.

Lemma 5.2a. Inverse optimality. *Consider the control affine system (1). Let* $u = \phi(x)$ *be a stabilizing control, with respect to a manifold* S . *If there exist scalar functions* $V(x)$ *and* $L_1(x)$ *satisfying the following conditions*

$$V(x) = 0 \Leftrightarrow x \in S$$

$$V(x) \geq \alpha(d(x,S))$$

$$L_1(x) \geq \gamma(d(x,S))$$

$$L_1(x) + \nabla V(x)^T f(x) - \frac{1}{4} \nabla V(x)^T g(x) R^{-1} g(x)^T \nabla V(x) = 0 \qquad (5.4)$$

$$\phi(x) = -\frac{1}{2} R^{-1} g(x)^T \nabla V(x)^T$$

then $u = \phi(x)$ *is optimal with respect to the performance index with the integrand* $\mathcal{L}(x,u) = L_1(x) + u^T R u$. *Moreover the optimal value of the performance criterion equals* $J(x_0, \phi(x)) = V(x_0)$.

Proof. Assume one has an optimal control problem (5.2), (5.3) with the optimality performance integrand in (5.3) $\mathcal{L}(x,u) = L_1(x) + u^T R u$. Then Lemma 5.1 states the solution of optimal problem can be obtained by forming the Hamiltonian

$$H(x,u) = L_1(x) + u^T Ru + \nabla V(x)^T (f(x) + g(x)u).$$

This gives the optimal control in form of

$$\frac{\partial}{\partial u} H(x,u) = 2u^T R + \nabla V(x)^T g(x) = 0$$

$$u = \phi(x) = -\frac{1}{2} R^{-1} g(x)^T \nabla V(x)$$

This feedback control vanishes on the set S since the gradient $\nabla V(x)$ equals zero there. The Hamiltonian evaluated at this optimal control equals zero since

$$H(x,\phi(x)) = L_1(x) + \nabla V(x)^T f(x) - \frac{1}{4} \nabla V(x)^T g(x) R^{-1} g(x)^T \nabla V(x) = 0.$$

The Hamiltonian can then be concisely written in quadratic form

$$H(x,u) = L_1 + u^T Ru + \underbrace{\nabla Vf}_{=-L_1 + \frac{1}{4}\nabla VgR^{-1}g^T\nabla V^T} + \nabla Vgu$$

$$= u^T Ru + \underbrace{\frac{1}{4}\nabla VgR^{-1}g^T\nabla V^T}_{=\phi^T R\phi} + \underbrace{\nabla Vgu}_{=-2\phi^T Ru}$$

$$= (u - \phi)^T R(u - \phi) \geq 0.$$

The value of the performance criterion then follows as

$$J = \int_0^\infty \mathcal{L}(x,u)dt = -\int_0^\infty \nabla V(x)^T f(x,u)dt + \int_0^\infty H(x,u)dt = -\int_0^\infty \frac{dV}{dt}dt + \underbrace{\int_0^\infty H(x,u)dt}_{\geq 0}$$

$$= V(x_0) - \underbrace{V(x(t \to \infty))}_{\to 0} + \underbrace{\int_0^\infty H(x,u)dt}_{\geq 0}$$

The optimum is reached precisely at $u = \phi(x)$, for which the integral on the right side vanishes, and by asymptotic stability assumption $V(x(t)) \to 0$, thus the optimal value of the performance criterion equals $J^*(x_0) = J(x_0, \phi(x)) = V(x_0)$. ∎

That this optimal feedback control $u = \phi(x)$ is stabilizing follows also from the Lyapunov equation

$$\dot{V}(x) = \nabla V^T f + \nabla V^T g u = \nabla V^T f - \frac{1}{2}\nabla V^T g R^{-1} g^T \nabla V$$

$$= -L_1 + \frac{1}{4}\nabla V^T g R^{-1} g^T \nabla V - \frac{1}{2}\nabla V^T g R^{-1} g^T \nabla V$$

$$= -L_1(x) - \frac{1}{4}\nabla V^T g R^{-1} g^T \nabla V \leq -L_1(x) \leq -\gamma(d(x, S))$$

By the assumptions of Lemma 5.2a the Lyapunov function $V(x)$ satisfies all the conditions of the Theorem 5.1 for partial stability.

In the linear quadratic case, *i.e.* $\dot{x} = Ax + Bu$, $L_1 = x^T Q x$, $V(x) = x^T P x$, the Hamilton Jacobi Bellman equation (HJB) (5.4) becomes the Algebraic Riccati equation (ARE)

$$Q + A^T P + PA - PBR^{-1}B^T P = 0. \tag{5.5}$$

The next more general result allows state-control cross-weighting terms in the performance integrand $\mathcal{L}$. This result is used in Section 5.8.

Lemma 5.2b. Inverse optimality with cross-weighting terms. *Consider the control affine system* (5.2). *Let* $u = \phi(x)$ *be a stabilizing control, with respect to a manifold* S. *If there exist scalar functions* $V(x)$ *and* $L_1(x)$, $L_2(x)$ *satisfying the following conditions*

$$V(x) = 0 \Leftrightarrow x \in S$$

$$V(x) \geq \alpha(d(x, S))$$

$$L_2(x) = 0 \Leftarrow x \in S$$

$$L_1(x) + \nabla V(x) f(x) - \frac{1}{4}\left[L_2(x) + \nabla V(x)^T g(x)\right] R^{-1} \left[L_2^T(x) + g(x)^T \nabla V(x)\right] = 0$$

$$\phi(x) = -\frac{1}{2}R^{-1}(L_2^T(x) + g(x)^T \nabla V(x))$$

$$L_1(x) + L_2(x)\phi + \phi^T R \phi \geq \gamma(d(x, S))$$

then $u = \phi(x)$ *is optimal with respect to the performance index with the integrand* $\mathcal{L}(x, u) = L_1(x) + L_2(x)u + u^T R u$. *Moreover the optimal value of the performance criterion equals* $J(x_0, \phi(x)) = V(x_0)$.

Proof. Assume one has an optimal control problem (5.2), (5.3) with the optimality performance integrand in (5.3) $\mathcal{L}(x,u) = L_1(x) + L_2(x)u + u^T Ru$. Then Lemma 5.1 states the solution of optimal problem can be obtained by forming the Hamiltonian

$$H(x,u) = L_1(x) + L_2(x)u + u^T Ru + \nabla V(x)^T (f(x) + g(x)u).$$

This gives the optimal control in form of

$$\frac{\partial}{\partial u} H(x,u) = 2u^T R + L_2(x) + \nabla V(x)^T g(x) = 0$$

$$u = \phi(x) = -\frac{1}{2} R^{-1}(L_2^T(x) + g(x)^T \nabla V(x))$$

This feedback control vanishes on the set S since the gradient $\nabla V(x)$ and $L_2(x)$ equal zero there. The Hamiltonian evaluated at this optimal control equals zero since

$$H(x,\phi(x)) = L_1(x) + L_2(x)\left[-\frac{1}{2} R^{-1}(L_2^T(x) + g(x)^T \nabla V(x))\right]$$

$$+ \left[-\frac{1}{2} R^{-1}(L_2^T(x) + g(x)^T \nabla V(x))\right]^T R \left[-\frac{1}{2} R^{-1}(L_2^T(x) + g(x)^T \nabla V(x))\right]$$

$$+ \nabla V(x)^T f(x) + \nabla V(x)^T g(x)\left[-\frac{1}{2} R^{-1}(L_2^T(x) + g(x)^T \nabla V(x))\right]$$

$$= L_1 + \nabla V^T f - \frac{1}{2} L_2 R^{-1} L_2^T - \frac{1}{2} L_2 R^{-1} g^T \nabla V + \frac{1}{4} L_2 R^{-1} L_2^T + \frac{1}{4} \nabla V^T g R^{-1} g^T \nabla V$$

$$+ \frac{1}{2} L_2 R^{-1} g^T \nabla V - \frac{1}{2} \nabla V^T g R^{-1} L_2^T - \frac{1}{2} \nabla V^T g R^{-1} g^T \nabla V$$

$$= L_1 + \nabla V^T f - \frac{1}{4} L_2 R^{-1} L_2^T - \frac{1}{2} L_2 R^{-1} g^T \nabla V - \frac{1}{4} \nabla V^T g R^{-1} g^T \nabla V$$

$$= L_1 + \nabla V^T f - \frac{1}{4}\left[L_2 + \nabla V^T g\right] R^{-1}\left[L_2^T + g^T \nabla V\right] = 0$$

This Hamiltonian can then be concisely written in quadratic form

$$H(x,u) = L_1 + L_2 u + u^T Ru + \underbrace{\nabla Vf}_{=-L_1 + \frac{1}{4} \nabla V g R^{-1} g^T \nabla V^T + \frac{1}{4} L_2 R^{-1} L_2^T + \frac{1}{2} L_2 R^{-1} g^T \nabla V^T} + \nabla V gu$$

$$= u^T Ru + \underbrace{\frac{1}{4} \nabla V g R^{-1} g^T \nabla V^T + \frac{1}{4} L_2 R^{-1} L_2^T + \frac{1}{2} L_2 R^{-1} g^T \nabla V^T}_{=\phi^T R\phi} + \underbrace{L_2 u + \nabla V gu}_{=-2\phi^T Ru}$$

$$= (u - \phi)^T R(u - \phi) \geq 0.$$

The value and the optimal value of the performance criterion then follow as in Lemma 5.2a. ∎

That this optimal feedback control $u = \phi(x)$ is stabilizing follows also from the Lyapunov equation

$$\dot{V}(x) = \nabla V^T f + \nabla V^T g u = \nabla V^T f - \frac{1}{2} \nabla V^T g R^{-1}(L_2^T + g^T \nabla V)$$

$$= -L_1 + \frac{1}{4} L_2 R^{-1} L_2^T + \frac{1}{2} L_2 R^{-1} g^T \nabla V + \frac{1}{4} \nabla V^T g R^{-1} g^T \nabla V$$

$$- \frac{1}{2} \nabla V^T g R^{-1} g^T \nabla V - \frac{1}{2} \nabla V^T g R^{-1} L_2^T$$

$$= -L_1(x) - \frac{1}{4} \nabla V^T g R^{-1} g^T \nabla V + \frac{1}{4} L_2 R^{-1} L_2^T$$

$$= -L_1(x) - L_2(x)\phi - \phi^T R \phi \le -\gamma(d(x, S))$$

By the assumptions of Lemma 5.2b the Lyapunov function $V(x)$ satisfies all the conditions of the Theorem 5.1 for partial stability.

5.5. Optimal Cooperative Control for Quadratic Performance Index and Single Integrator Agent Dynamics

This section considers the global optimal control problem for cooperative single-integrator systems on a given graph topology. Any admissible control protocols for cooperative control must be distributed in the sense that they respect the prescribed graph topology by using information only about the agent and its direct neighbors. Unfortunately, in global optimal control problems, the optimal controls generally require information about all the states [13]. In fact, the communication restrictions imposed by a prescribed graph topology can severely limit the possible global optimal performance of cooperative agents on that graph. It is seen here that the cooperative global optimal control problem only has a solution using distributed control protocols under certain conditions on the graph topology. Moreover, if these conditions hold, then for the existence of global optimality, the global performance index must be selected in a certain fashion that depends on the graph topology. This helps clarify the interactions between stability, optimality, and the graph topology in multi-agent systems on graphs. The approach used here is to consider inverse optimality of consensus protocols for the leaderless consensus [10],[18] and the pinning control cases [26]. We call these respectively the cooperative regulator and cooperative tracker problem.

This section considers the single integrator agent dynamics

$$\dot{x}_i = u_i \in \mathbb{R} , \tag{5.6}$$

or in global form

$$\dot{x} = u , \tag{5.7}$$

where $x = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix}^T$ and $u = \begin{bmatrix} u_1 & \dots & u_N \end{bmatrix}^T$. More general dynamics are considered in the next section.

5.5.1   Optimal Cooperative Regulator

In the leaderless consensus case [10],[18], which we call the cooperative regulator problem, where there are N agents with states $x_i \in \mathbb{R}$, $i = 1..N$, it is desired for all agents to achieve the same state, that is $\|x_i(t) - x_j(t)\| \to 0$ as $t \to \infty$, $\forall (i, j)$. Then the consensus manifold, $S := span(\underline{1})$, where $\underline{1} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T \in \mathbb{R}^N$, is noncompact. Therefore to apply inverse optimality to the cooperative regulator problem, one needs the partial stability results of Theorem 5.1.

Define the local neighborhood tracking error as

$$\eta_i = \sum_{j \in \mathcal{N}_i} e_{ij} (x_i - x_j) . \tag{5.8}$$

The overall local neighborhood tracking error $e = \begin{bmatrix} \eta_1 & \dots & \eta_N \end{bmatrix}^T$ is $e = Lx$. Then a distributed control that guarantees consensus is given as $u_i = -\eta_i$ or in global form

$$u = -Lx , \tag{5.9}$$

which gives the global closed-loop system

$$\dot{x} = -Lx . \tag{5.10}$$

Lemma 5.3. [10], [17]. *If the graph has a spanning tree then 0 is a simple eigenvalue of the graph Laplacian matrix. Furthermore the distributed control law $u = -Lx$ solves the cooperative regulator problem for the system (5.7).* ■

If the condition of Lemma 5.3 is satisfied then the measure of the distance from the consensus manifold in Theorem 5.1 is given by any norm of the local neighborhood disagreement vector $e = Lx$.

The following theorem states sufficient conditions for the global optimality of the linear distributed control law (5.9).

Theorem 5.2. *Let the systems given as (5.7) be connected on a graph topology L. Then for some $R = R^T > 0$ the distributed control $u = -Lx$ is globally optimal with respect to the performance index*

$$J(x_0, u) = \int_0^\infty (x^T L^T RLx + u^T Ru)dt = \int_0^\infty (e^T Re + u^T Ru)dt \qquad (5.11)$$

and is stabilizing to a manifold which is the null space of L if there exists a positive semidefinite matrix $P = P^T \geq 0$ satisfying

$$P = RL. \qquad (5.12)$$

Moreover, if the graph contains a spanning tree then consensus is reached.

Proof: First note that since R is nonsingular the null space of P equals that of L. The Lyapunov function $V(x) = x^T Px$ equals zero on the null space of P and is greater than zero elsewhere since $P \geq 0$. If one introduces the orthogonal complement of the null space of P one has $V(x) = x^T Px \geq \sigma_{>0\min}(P)\|y\|^2$ where $x = x_0 + y$, $y \in \ker P^\perp$. Since $\|y\|$ is a measure of the distance $d(x, \ker P)$ the Lyapunov function is found to satisfy the conditions 1,2 of Lemma 5.2a.

The part of the performance integrand $L_1(x) = x^T Qx = x^T L^T RLx \geq \sigma_{>0\min}(L^T RL)\|y\|^2$ satisfies the condition of Theorem 5.1. The Algebraic Riccati equation (5.5) given as

$$L^T RL - PR^{-1}P = 0$$

is satisfied by $P = RL$, and $u = -Lx = -R^{-1}Px$. Therefore all the conditions of Lemma 5.2 are satisfied by this linear quadratic problem.

If the graph has a spanning tree, by Lemma 5.3, zero is a simple eigenvalue of L and thus the null space of P is the consensus manifold $span(\underline{1})$. Then, consensus is reached. This concludes the proof. ∎

This result was first given in [3] for undirected graphs, where condition (5.12) always holds. See the discussion below.

Note that the time derivative of Lyapunov function $V(x) = x^T Px$ equals

$$\dot{V}(x) = -x^T(PL + L^T P)x = -2x^T L^T RLx \leq -\sigma_{>0\min}(L^T RL)\|y\|^2,$$

which guarantees asymptotic stability of the null space of L or equivalently of P. Also, since $V(x) = x^T P x \leq \sigma_{>0\max}(P)\|y\|^2$, according to Theorem 5.1, the stability is uniform.

It is seen that if condition (5.12) is satisfied for the given graph topology L, then any admissible performance index for global optimality must depend on the graph topology as prescribed by the form (5.11), where the state weighting term is $x^T Q x = x^T L^T R L x$, and R moreover must be selected to satisfy (5.12).

Remark 5.4. Undirected Graphs. The constraint $RL = P$ is a joint constraint on the graph topology and performance criterion. This can easily be met in the case of undirected graphs $L = L^T$ by selecting $R = I$, in which case $P = L$ and $Q = L^T L$. Then, the value of the optimality criterion is given as $J^*(x_0) = V(x_0) = x_0^T L x_0$ which is the graph Laplacian potential [28]. Moreover $\sigma_{>0\min}(L)$ in the proof equals $\lambda_2(L)$, the Fiedler eigenvalue of the graph. More generally, for undirected graphs the constraint (5.12) can be interpreted as the commutativity condition $RL = LR$, which restricts the choice of the symmetric matrix R, but allows choices other than $R = I$. Note that then R generally depends on the graph topology.

The constraint (5.12) can also be met by special bidirectional graphs $L = R^{-1} P$ with P being a symmetric Laplacian matrix and R a positive diagonal matrix. For general digraphs, since P is symmetric, the constraint (5.12) implies the condition

$$RL = L^T R . \tag{5.13}$$

The constraint (5.12) defines a new class of digraphs and is further discussed in Section 5.7.

5.5.2 *Optimal Cooperative Tracker*

In the pinning control case [26],[10] it is desired for all the agents to reach the state x_0 of a leader or control node 0, that is $\|x_i(t) - x_0(t)\| \to 0$ as $t \to \infty$, $\forall i$. This is called a synchronization problem since the leader's state may be time-varying. We call this the cooperative tracker problem. It is assumed here that $\dot{x}_0 = 0$.

Consider the optimal cooperative tracker for systems (5.6) and define the local neighborhood tracking error as

$$\eta_i = \sum_{j \in \mathcal{N}_i} e_{ij}(x_i - x_j) + g_i(x_i - x_0) , \tag{5.14}$$

where the pinning gains $g_i \geq 0$ are nonzero only for a few nodes i directly connected to a leader node. The overall local neighborhood tracking error is equal to $e = (L+G)\delta$, where the global disagreement error is $\delta = x - \underline{1}x_0$. Then a distributed control that guarantees synchronization is given as $u_i = -\eta_i$, or in global form

$$u = -(L+G)\delta \,, \tag{5.15}$$

with $G = diag(g_1...g_N)$ the diagonal matrix of pinning gains. This gives the global dynamics

$$\dot{x} = u = -(L+G)\delta. \tag{5.16}$$

To achieve synchronization, the global disagreement system

$$\dot{\delta} = u = -(L+G)\delta \,, \tag{5.17}$$

must be asymptotically stabilized to the origin. This means that partial stability notions need not be used, and the sought solution P of the global Algebraic Riccati Equation (5.5) must be positive definite.

Lemma 5.4. [10]. *If the graph has a spanning tree, given that there exists at least one non zero pinning gain connecting into a root node, then $L+G$ is nonsingular, and the distributed control $u = -(L+G)\delta$ solves the cooperative tracking problem for the system (5.7).*

If the conditions of Lemma 5.4 are satisfied, then the measure of distance from the target state x_0 is given by any norm of the local neighborhood disagreement vector $e = (L+G)\delta$.

The next result gives sufficient conditions for the global optimality of the distributed linear protocol (5.15).

Theorem 5.3. *Given a graph topology $(L+G)$, let the error dynamics be given as (5.17), and the conditions of Lemma 5.4 be satisfied. Then for some $R = R^T > 0$ the distributed control $u = -(L+G)\delta$ is globally optimal with respect to the performance index*

$$J(\delta_0, u) = \int_0^\infty (\delta^T (L+G)^T R(L+G)\delta + u^T Ru)dt = \int_0^\infty (e^T Re + u^T Ru)dt, \tag{5.18}$$

and guarantees synchronization to the reference state x_0 if there exists a positive definite matrix $P = P^T > 0$ satisfying

$$P = R(L+G).$$ (5.19)

Proof. The Lyapunov function $V(\delta) = \delta^T P \delta > 0$, and performance integrand $L_1(\delta) = \delta^T Q \delta = \delta^T (L+G)^T R(L+G)\delta > 0$ satisfy the conditions of Lemma 5.2. The Algebraic Riccati equation

$$(L+G)^T R(L+G) - PR^{-1}P = 0$$

is satisfied by P, and $u = -(L+G)\delta = -R^{-1}P\delta$ thus proving the theorem. ∎

Remarks similar to Remark 5.4 about the constraint (5.19) between R and $L+G$ apply here. Under conditions of Lemma 5.4 and (5.19) P is nonsingular since both R and $L+G$ are. For the undirected graph case, one choice that satisfies (5.19) is $R = I$. Then $P = L+G$. A more general choice for undirected graphs is any $R > 0$ that commutes with $L+G$.

For general digraphs, since P is symmetric, the constraint (5.19) implies the condition

$$R(L+G) = (L+G)^T R.$$ (5.20)

If R is diagonal this is equivalent to (5.13) and conditions (5.12) and (5.19) are equivalent. The constraint (5.19) is further discussed in Section 5.7.

It is seen that if condition (5.19) is satisfied for the given graph topology $L+G$, then any admissible performance index for global optimality must depend on the graph topology as prescribed by the form (5.18), where the state error weighting term is $\delta^T Q \delta = \delta^T (L+G)^T R(L+G)\delta$, and R moreover must be selected to satisfy (5.19).

Remark 5.5. Both in the optimal cooperative regulator and tracker performance criteria, the expressions Lx and $(L+G)\delta$, respectively, play a prominent role. In either case one can concisely write the performance index as

$$J(e_0, u) = \int_0^\infty (e^T R e + u^T R u) dt$$

.

In this performance index, matrix R generally depends on the graph topology through the constraints (5.12) or (5.19). Moreover, the local neighborhood tracking error $e(t)$ depends on the graph topology L. Thus, when written in terms of the global disagreement error, Q depends on the graph topology in the quadratic term $L_1(\delta) = \delta^T Q \delta = \delta^T (L+G)^T R(L+G)\delta$.

5.6. Optimal Cooperative Control for Quadratic Performance Index and Linear Time-invariant Agent Dynamics

This section considers the global optimal control problem for cooperative systems on a given graph topology where the agents have identical linear time-invariant dynamics. Any admissible control protocols for cooperative control must be distributed in the sense that they respect the prescribed graph topology by using information only about the agent and its direct neighbors. It is seen here that the cooperative global optimal control problem only has a solution using distributed control protocols under certain conditions on the graph topology. Moreover, if these conditions hold, then for the existence of global optimality, the performance index must be selected in a certain fashion that depends on the graph topology. The approach used here is to consider inverse optimality of consensus protocols for the leaderless consensus [10],[18] and the pinning control cases [26]. We call these respectively the cooperative regulator and cooperative tracker problem.

This section considers inverse optimality of consensus protocols for the leaderless and pinning control cases for agents having states $x_i \in \mathbb{R}^n$ with linear time-invariant dynamics

$$\dot{x}_i = Ax_i + Bu_i, \tag{5.21}$$

or in global form

$$\dot{x} = (I_N \otimes A)x + (I_N \otimes B)u \tag{5.22}$$

Here we consider the case of identical agent dynamics. If the agent dynamics are heterogeneous, or not the same, then the Kronecker product cannot be used. The analysis in this chapter relies heavily on the Kronecker product.

5.6.1 Optimal Cooperative Regulator

In the cooperative regulator problem it is desired for all agents to achieve the same state, that is $\left\| x_i(t) - x_j(t) \right\| \to 0$ as $t \to \infty$, $\forall (i, j)$. Define the local neighborhood tracking error as

$$\eta_i = \sum_{j \in \mathcal{N}_i} e_{ij}(x_i - x_j). \tag{5.23}$$

where $\eta_i \in \mathbb{R}^n$. The overall local neighborhood tracking error $e = \begin{bmatrix} \eta_1 & \dots & \eta_N \end{bmatrix}^T$ is equal to $e = (L \otimes I_n)x$. Then a distributed control that guarantees consensus is given as $u_i = -cK_2\eta_i$ for an appropriate coupling gain $c > 0$ and local feedback matrix K_2, as shown in [27]. In global form this is

$$u = -c(L \otimes K_2)x, \tag{5.24}$$

which gives the global closed-loop system

$$\dot{x} = (I_N \otimes A - cL \otimes BK_2)x \tag{5.25}$$

The next result gives sufficient conditions for the global optimality of the distributed linear protocol (5.24).

Theorem 5.4. *Let the systems given as (5.22) be connected on a graph topology L. Suppose there exist a positive semidefinite matrix $P_1 = P_1^T \geq 0$, and a positive definite matrix $P_2 = P_2^T > 0$ satisfying*

$$P_1 = cR_1L, \tag{5.26}$$

$$A^T P_2 + P_2 A + Q_2 - P_2 B R_2^{-1} B^T P_2 = 0, \tag{5.27}$$

for some $Q_2 = Q_2^T > 0, R_1 = R_1^T > 0, R_2 = R_2^T > 0$ and coupling gain $c > 0$. Define the feedback gain matrix K_2 as

$$K_2 = R_2^{-1}B^T P_2. \tag{5.28}$$

Then the distributed control $u = -cL \otimes K_2 x$ is globally optimal with respect to the performance index

$$J(x_0, u) =$$

$$\int_0^\infty x^T \left[c^2 (L \otimes K_2)^T (R_1 \otimes R_2)(L \otimes K_2) - cR_1 L \otimes (A^T P_2 + P_2 A) \right] x + u^T (R_1 \otimes R_2) u \, dt, \tag{5.29}$$

and is stabilizing to the null space of $L \otimes I_n$ for sufficiently high coupling gain c satisfying (5.31). Moreover, if the conditions of Lemma 5.3 hold then consensus is reached.

Proof. Form the matrix $P = P_1 \otimes P_2$. Since P_2 is nonsingular the null space of P equals the null space of $P_1 \otimes I_n$ which by nonsingularity of R_1 is the same as that of $L \otimes I_n$. Therefore by positive semidefiniteness the Lyapunov function $V(x) = x^T P x = x^T (P_1 \otimes P_2) x$ is zero on the null space of $L \otimes I_n$ and positive elsewhere. By the arguments similar to those presented in the proof of Theorem 5.2 this Lyapunov function satisfies the conditions of Theorem 5.1.

Also,

$$\begin{aligned} L_1(x) &= x^T Q x \\ &= x^T (c^2 (L \otimes K_2)^T (R_1 \otimes R_2)(L \otimes K_2) - cR_1 L \otimes (A^T P_2 + P_2 A)) x \end{aligned} \tag{5.30}$$

equals zero on the null space of $L \otimes I_n$, and satisfies the condition of Lemma 5.2 if the value of the coupling gain is taken as

$$c > \frac{\sigma_{\max}(R_1 L \otimes (Q_2 - K_2^T R_2 K_2))}{\sigma_{>0\min}(L^T R_1 L \otimes K_2^T R_2 K_2)}. \tag{5.31}$$

$$\begin{aligned} Q &= c^2 (L \otimes K_2)^T (R_1 \otimes R_2)(L \otimes K_2) - cR_1 L \otimes (A^T P_2 + P_2 A) \\ &= c^2 L^T R_1 L \otimes K_2^T R_2 K_2 + cR_1 L \otimes (Q_2 - K_2^T R_2 K_2) \end{aligned}$$

So one has

$$c^2 \sigma_{>0\min}(L^T R_1 L \otimes K_2^T R_2 K_2) - c\sigma_{\max}(R_1 L \otimes (Q_2 - K_2^T R_2 K_2)) > 0 \Rightarrow Q \geq 0$$

which gives the lower bound on the coupling gain c. The algebraic Riccati equation for system (5.22)

$$(I_N \otimes A)^T P + P(I_N \otimes A) + Q - P(I_N \otimes B) R^{-1} (I_N \otimes B)^T P = 0,$$

written as

$$P_1 \otimes A^T P_2 + P_1 \otimes P_2 A + Q - (P_1 \otimes P_2 B)R_1^{-1} \otimes R_2^{-1}(P_1 \otimes B^T P_2) = 0$$
$$P_1 \otimes (A^T P_2 + P_2 A) + Q - P_1 R_1^{-1} P_1 \otimes (P_2 B R_2^{-1} B^T P_2) = 0$$

is satisfied by the choice of $P = P_1 \otimes P_2$ since

$$
\begin{aligned}
Q &= c^2 (L \otimes K_2)^T (R_1 \otimes R_2)(L \otimes K_2) - c R_1 L \otimes (A^T P_2 + P_2 A) \\
&= c^2 L^T R_1 L \otimes K_2^T R_2 K_2 + c R_1 L \otimes (Q_2 - P_2 B R_2^{-1} B^T P_2) \\
&= P_1 R_1^{-1} P_1 \otimes P_2 B R_2^{-1} B^T P_2 + P_1 \otimes (Q_2 - P_2 B R_2^{-1} B^T P_2) \\
&= Q_1 \otimes P_2 B R_2^{-1} B^T P_2 + P_1 \otimes (Q_2 - P_2 B R_2^{-1} B^T P_2)
\end{aligned}
$$

,

where $Q_1 = c^2 L^T R_1 L = P_1 R_1^{-1} P_1$ is introduced for notational simplicity. It follows by the conditions (5.26), (5.27) of the Theorem that

$$P_1 \otimes (A^T P_2 + P_2 A^T + Q_2 - P_2 B R_2^{-1} B^T P_2) + (Q_1 - P_1 R_1^{-1} P_1) \otimes (P_2 B R_2^{-1} B^T P_2) = 0$$

By conditions (5.26), (5.27) and (5.28) the control satisfies

$$
\begin{aligned}
u &= -c L \otimes K_2 x = -R^{-1}(I_N \otimes B)^T P x = -(R_1^{-1} \otimes R_2^{-1})(I_N \otimes B^T)(P_1 \otimes P_2)x \\
&= -R_1^{-1} P_1 \otimes R_2^{-1} B^T P_2 x
\end{aligned}
$$

.

Therefore all the conditions of the Theorem 5.1 are satisfied.

If the graph has a spanning tree, then by Lemma 5.3, zero is a simple eigenvalue of L and thus the null space of P is the consensus manifold $span(\underline{1})$. Then, consensus is reached. This concludes the proof. ■

Note that the time derivative of the Lyapunov function $V(x) = x^T P x$ equals

$$
\begin{aligned}
\dot{V}(x) &= 2x^T P \dot{x} = 2x^T P(I_N \otimes Ax - cL \otimes BK_2 x) \\
&= x^T (P_1 \otimes (P_2 A + A^T P_2) - 2P(I_N \otimes B)(R_1^{-1} \otimes R_2^{-1})(I_N \otimes B^T)P)x \\
&= -x^T (Q + c^2 (L \otimes K_2)^T (R_1 \otimes R_2)(L \otimes K_2))x \le -x^T Q x
\end{aligned}
$$

implying asymptotic stability to the nullspace of Q, which equals the null space of $L \otimes I_n$ (see (5.30)).

It is emphasized that this theorem provides a global optimal solution given in terms of a local Riccati design equation (5.27) solved at each node. The dimension of this equation is the dimension of the local agent dynamics n and it does not depend on the number of nodes N in the graph. As such, this method is scalable to large graphs.

Note that local optimal design in terms of Riccati equation (5.27) does not guarantee global optimal performance of all the agents unless condition (5.26) holds and the global performance index is selected to depend on the graph topology as in (5.29). According to the results of Section 3.2, however, The locally optimal protocol (5.27), (5.28) does guarantee stable consensus if the conditions of Lemma 5.3 hold.

Remark 5.6. The graph topology constraint (5.26) is similar to the constraint (5.12), and comments similar to those stated in Remark 5.4 apply here as well. This constraint is further discussed in Section 5.7. On the other hand, there always exists a $P_2 = P_2^T > 0$ satisfying the local Riccati equation (5.27) if (A, B) is stabilizable and $(A, \sqrt{Q_2})$ is observable. This local Riccati equation was used for design of protocols that guarantee synchronization in Section 3.2. The value of the coupling gain c must be sufficiently great to overpower the, generally, indefinite terms stemming from the drift dynamics Ax_i as in condition (5.31).

5.6.2   Optimal Cooperative Tracker

We now consider the cooperative tracker problem for linear time-invariant agent dynamics (5.21). In the pinning control or cooperative tracker problem [26],[10] it is desired for all the agents to reach the state $x_0 \in \mathbb{R}^n$ of a leader or control node 0, that is $\|x_i(t) - x_0(t)\| \to 0$ as $t \to \infty$, $\forall i$. It is assumed here that the leader has command generator dynamics

$$\dot{x}_0 = Ax_0 . \tag{5.32}$$

Consider the optimal cooperative tracker problem for the systems (5.21) and define the local neighborhood tracking errors as

$$\eta_i = \sum_{j \in \mathcal{N}_i} e_{ij}(x_i - x_j) + g_i(x_i - x_0), \tag{5.33}$$

where $\eta_i \in \mathbb{R}^n$. The pinning gains $g_i \geq 0$ are nonzero only for a few nodes directly connected to a leader node. The overall local neighborhood tracking error is equal to $e = (L + G) \otimes I_n \delta$, where the global disagreement error is $\delta = x - \underline{1} \otimes x_0$. Then a distributed control that guarantees synchronization is given as $u_i = -cK_2\eta_i$, for an appropriate coupling gain $c > 0$ and local feedback matrix K_2. In global form this is

$$u = -c(L+G) \otimes K_2 \delta ,$$
(5.34)

with $G = diag(g_1 ... g_N)$ the diagonal matrix of pinning gains. This gives the global dynamics

$$\dot{x} = I_N \otimes Ax - c(L+G) \otimes BK_2 \delta .$$
(5.35)

To achieve synchronization the global disagreement system

$$\dot{\delta} = (I_N \otimes A)\delta + (I_N \otimes B)u ,$$
(5.36)

or, with control $u = -c(L+G) \otimes K_2 \delta$,

$$\dot{\delta} = (I_N \otimes A - c(L+G) \otimes BK_2)\delta ,$$
(5.37)

must be asymptotically stabilized to the origin. This means that partial stability notions need not be used, and the sought solution P of the global Algebraic Riccati Equation (5.5) must be positive definite. Stability of (5.37) to the origin is equivalent to asymptotic reference tracking of x_0 and synchronization under the conditions of Lemma 5.4.

The next theorem gives sufficient conditions for the global optimality of the distributed linear protocol (5.34).

Theorem 5.5. *Given a graph topology* $(L+G)$, *let the error dynamics be given as* (5.37) *and the conditions of Lemma 5.4 be satisfied. Suppose there exist a positive definite matrix* $P_1 = P_1^T > 0$, *and a positive definite matrix* $P_2 = P_2^T > 0$ *satisfying*

$$P_1 = cR_1(L+G),$$
(5.38)

$$A^T P_2 + P_2 A + Q_2 - P_2 BR_2^{-1} B^T P_2 = 0 ,$$
(5.39)

for some $Q_2 = Q_2^T > 0, R_1 = R_1^T > 0, R_2 = R_2^T > 0$ *and coupling gain* $c > 0$. *Define the feedback gain matrix* K_2 *as*

$$K_2 = R_2^{-1} B^T P_2 .$$
(5.40)

Then the control $u = -c(L+G) \otimes K_2 \delta$ *is globally optimal with respect to the performance index*

$$J(\delta_0, u)$$
$$= \int_0^\infty \delta^T \left[c^2 ((L+G) \otimes K_2)^T (R_1 \otimes R_2)((L+G) \otimes K_2) - cR_1(L+G) \otimes (A^T P_2 + P_2 A) \right] \delta + u^T (R_1 \otimes R_2) u \, dt$$

(5.41)

and guarantees synchronization for sufficiently high coupling gain c *satisfying (5.42).*

Proof. Form the matrix $P = P_1 \otimes P_2$. Since P_1 and P_2 are nonsingular so is P. The Lyapunov function $V(\delta) = \delta^T P \delta > 0$, by the arguments similar to those presented in the proof of Theorem 5.3. This Lyapunov function satisfies the conditions of Theorem 5.1, specialized to a single equilibrium point. Also,

$$L_1(\delta) = \delta^T Q \delta$$
$$= \delta^T [c^2 ((L+G) \otimes K_2)^T (R_1 \otimes R_2)((L+G) \otimes K_2) + cR_1(L+G) \otimes (A^T P_2 + P_2 A)] \delta$$

is positive definite and satisfies the condition of Theorem 5.1 if the value of the coupling gain is taken as

$$c > \frac{\sigma_{\max}(R_1(L+G) \otimes (Q_2 - K_2^T R_2 K_2))}{\sigma_{\min}((L+G)^T R_1(L+G) \otimes K_2^T R_2 K_2)} .$$

(5.42)

$$Q = c^2 ((L+G) \otimes K_2)^T (R_1 \otimes R_2)((L+G) \otimes K_2) - cR_1(L+G) \otimes (A^T P_2 + P_2 A)$$
$$= c^2 (L+G)^T R_1(L+G) \otimes K_2^T R_2 K_2 + cR_1(L+G) \otimes (Q_2 - K_2^T R_2 K_2)$$

So one has

$$c^2 \sigma_{\min}((L+G)^T R_1(L+G) \otimes K_2^T R_2 K_2) - c\sigma_{\max}(R_1(L+G) \otimes (Q_2 - K_2^T R_2 K_2)) > 0 \Rightarrow Q > 0$$

which gives the lower bound on the coupling gain c. The Algebraic Riccati equation (5.5) for system (5.36)

$$(I_N \otimes A)^T P + P(I_N \otimes A) + Q - P(I_N \otimes B)R^{-1}(I_N \otimes B)^T P = 0 ,$$

written as

$$P_1 \otimes A^T P_2 + P_1 \otimes P_2 A + Q - (P_1 \otimes P_2 B) R_1^{-1} \otimes R_2^{-1} (P_1 \otimes B^T P_2) = 0$$
$$P_1 \otimes (A^T P_2 + P_2 A) + Q - P_1 R_1^{-1} P_1 \otimes (P_2 B R_2^{-1} B^T P_2) = 0$$

is satisfied by the choice of $P = P_1 \otimes P_2$ since

$$
\begin{aligned}
Q &= c^2 ((L+G) \otimes K_2)^T (R_1 \otimes R_2)((L+G) \otimes K_2) - c R_1 (L+G) \otimes (A^T P_2 + P_2 A) \\
&= c^2 (L+G)^T R_1 (L+G) \otimes K_2^T R_2 K_2 + c R_1 (L+G) \otimes (Q_2 - P_2 B R_2^{-1} B^T P_2) \\
&= P_1 R_1^{-1} P_1 \otimes P_2 B R_2^{-1} B^T P_2 + P_1 \otimes (Q_2 - P_2 B R_2^{-1} B^T P_2) \\
&= Q_1 \otimes P_2 B R_2^{-1} B^T P_2 + P_1 \otimes (Q_2 - P_2 B R_2^{-1} B^T P_2)
\end{aligned}
$$

where $Q_1 = c^2 (L+G)^T R_1 (L+G) = P_1 R_1^{-1} P_1$ is introduced for notational simplicity. It follows by the conditions (5.38), (5.39) of the theorem that

$$P_1 \otimes (A^T P_2 + P_2 A^T + Q_2 - P_2 B R_2^{-1} B^T P_2) + (Q_1 - P_1 R_1^{-1} P_1) \otimes (P_2 B R_2^{-1} B^T P_2) = 0$$

By conditions (5.38), (5.39) and (5.40) the control u satisfies

$$
\begin{aligned}
u &= -c(L+G) \otimes K_2 \delta = -R^{-1} (I_N \otimes B)^T P \delta \\
&= -(R_1^{-1} \otimes R_2^{-1})(I_N \otimes B^T)(P_1 \otimes P_2) \delta = -R_1^{-1} P_1 \otimes R_2^{-1} B^T P_2 \delta
\end{aligned}
$$

Therefore all the conditions of the Theorem 5.1 are satisfied, and the control is stabilizing to the origin for (5.36). This guarantees synchronization and completing the proof. ∎

Note that the time derivative of the Lyapunov function $V(\delta) = \delta^T P \delta$ equals

$$
\begin{aligned}
\dot{V}(\delta) &= 2\delta^T P \dot{\delta} = 2\delta^T P(I_N \otimes A - c(L+G) \otimes B K_2) \delta \\
&= \delta^T (P_1 \otimes (P_2 A + A^T P_2) - 2P(I_N \otimes B)(R_1^{-1} \otimes R_2^{-1})(I_N \otimes B^T) P) \delta \\
&= -\delta^T (Q + c^2 (L \otimes K_2)^T (R_1 \otimes R_2)(L \otimes K_2)) \delta \le -\delta^T Q \delta < 0,
\end{aligned}
$$

implying asymptotic stability of (5.37) to the origin.

This theorem provides a global optimal solution given in terms of a local Riccati design equation (5.39) solved at each node. The dimension of this equation is the dimension of the local agent dynamics n and it does not depend on the number of nodes N in the graph. As such, this method is scalable to large graphs.

Note that local optimal design in terms of Riccati equation (5.39) does not guarantee global optimal performance of all the agents unless condition (5.38) holds and the global performance index is selected to depend on the graph topology as in (5.41). According to the results of Section 3.2, however, the locally optimal protocol (5.39), (5.40) does guarantee synchronization if the conditions of Lemma 5.4 hold.

Remark 5.7. The graph topology constraint (5.38) is similar to the constraint (5.19), and comments similar to those stated there apply here as well. This constraint is further discussed in Section 5.7. On the other hand, there always exists a $P_2 = P_2^T > 0$ satisfying (5.39) if (A, B) is stabilizable and $(A, \sqrt{Q_2})$ observable. This local Riccati equation was used for design of protocols that guarantee synchronization in Section 3.2. The value of the coupling gain c must be sufficiently great to overpower the, generally, indefinite terms stemming from the drift dynamics Ax_i, as in condition (5.42).

5.7.   Constraints on Graph Topology

In order to admit a distributed control solution to an appropriately defined global optimal control problem, the graph topology must satisfy the appropriate constraints (5.12), (5.19) or (5.26), (5.38) in terms of the Laplacian matrix L or the pinned Laplacian matrix $L + G$. This section investigates classes of graphs that satisfy those conditions. A new class of digraphs is presented for which the condition always holds, namely, the digraphs whose Laplacian matrix is simple, that is, having diagonal Jordan form.

Generally one can express these conditions as

$$RL = P , \tag{5.43}$$

where $R = R^T > 0$, L is a singular (*i.e.* the graph Laplacian) or nonsingular (*i.e.* the pinned graph Laplacian) M-matrix, and $P = P^T \geq 0$, a singular or nonsingular positive definite matrix, respectively. Equivalently

$$RL = L^T R \tag{5.44}$$

For the following classes of graph topologies one can satisfy this condition.

5.7.1. *Undirected Graphs*

Given that the graph is undirected, then L is symmetric, *i.e.* $L = L^T$ so the condition (5.43) becomes a commutativity requirement

$$RL = L^T R = LR . \tag{5.45}$$

See Remark 5.4 where it is shown that the choice $R = I$ always satisfies this condition for undirected graphs. Then $P = L$.

More generally, condition (5.45) is satisfied by symmetric matrices R and L if and only if R and L have all eigenvectors in common. Since L is symmetric it has a basis of orthogonal eigenvectors, and one can construct R satisfying (5.45) as follows. Let T be an orthogonal matrix whose columns are eigenvectors of L, then $L = T\Lambda T^T$ with Λ a diagonal matrix of real eigenvalues. Then for any positive definite diagonal matrix Θ one has that $R = T\Theta T^T > 0$ commutes with L and satisfies the commutativity requirement (5.45).

Note that the R so constructed for global optimal performance depends on all the eigenvectors of the Laplacian L in (5.43) (*i.e.* the graph Laplacian or the pinned Laplacian as appropriate). By contrast, it is known that consensus properties depend only on the left eigenvector for the zero eigenvalue of L [10],[18]. That is, for global optimal performance more information about the eigenstructure of the Laplacian is required than in simply achieving consensus.

5.7.2. Detail Balanced Graphs

Given a detail balanced graph (defined in Section 5.2), its Laplacian matrix is symmetrizble, that is there exists a positive diagonal matrix $\Lambda > 0$ such that $L = \Lambda P$ where P is a symmetric graph Laplacian matrix. Then, condition (5.43) holds with $R = \Lambda^{-1}$. Recall that for detail balanced graphs, the diagonal elements of R in the unpinned case are then the elements of the left eigenvector of the Laplacian for the eigenvalue of zero.

5.7.3. Directed Graphs with Simple Laplacian

In this subsection we introduce a new class of digraphs which, to our knowledge, has not yet appeared in the cooperative control literature. This class of digraphs admits a distributed control solution to an appropriately defined global optimal control problem.

Given a directed graph, let it be such that Laplacian matrix L (either the graph Laplacian or the pinned graph Laplacian) in (5.43) is simple, that is its Jordan form is diagonal (*i.e.* there exists a basis of rank 1 eigenvectors of L). Then there exists a matrix $R > 0$ depending on all the eigenvectors of the Laplacian L that satisfies the condition (5.43). This result is given in the following theorem. Let the Jordan form decomposition of L be given by

$$L = T^{-1}\Lambda T \tag{5.46}$$

with T the matrix of left eigenvectors and Λ diagonal and real.

Theorem 5.6. *Let L be a Laplacian matrix (generally not symmetric). Then there exists a positive definite symmetric matrix $R = R^T > 0$ such that $RL = P$ is a symmetric positive semidefinite matrix if and only if L is simple. Then a suitable choice for R is $R = T^T T$.*

Proof. (*i*) Let L be simple. Then it is diagonalizable, *i.e.* there exists a transformation matrix T such that $TLT^{-1} = \Lambda$, where Λ is a diagonal matrix of eigenvalues of L. Then

$$TLT^{-1} = \Lambda = \Lambda^T = T^{-T} L^T T^T,$$

implying $T^T TLT^{-1} T^{-T} = L^T$, which implies that $(T^T T)L = L^T (T^T T)$. Let $R = T^T T$. Obviously, $R = R^T > 0$ and $P = RL = T^T TL = T^T \Lambda T \geq 0$ since $\Lambda \geq 0$ ($\forall x$ $0 \leq x^T Px = x^T T^T \Lambda Tx = y^T \Lambda y \ \forall y$), [1].

(*ii*) Let L be positive semidefinite matrix. Suppose there exists $R = R^T > 0$ satisfying the condition $RL = P$ is a symmetric positive semidefinite matrix. Then one needs to show that L is simple. To show this, we will prove the contrapositive of this by contradiction. So we suppose the negation of the contrapositive. That is, suppose L is not simple but that there exists $R = R^T > 0$ satisfying the condition $RL = P$ is a symmetric positive semidefinite matrix.

Since L is not simple, there exists a coordinate transformation bringing L to a Jordan canonical form $T^{-1} LT = J$, with nonzero superdiagonal (otherwise L would be simple). Then one has $RL = RTJT^{-1} = P = P^T = T^{-T} J^T T^T R$. But then $(T^T RT)J = J^T (T^T RT)$. Therefore there exists $R_2 = T^T RT = R_2^T > 0$ such that $R_2 J = J^T R_2$. Without loss of generality let us assume that the first Jordan block is not simple, and with a slight abuse of notation $R_{2,11}$ will refer to the corresponding block in R_2. Then one has that $R_{2,11}(\lambda I + E) = (\lambda I + E^T)R_{2,11} \Rightarrow R_{2,11} E = E^T R_{2,11}$, where E is a nilpotent matrix having ones on the superdiagonal. This identity means that the first row and first column of $R_{2,11}$ are zero, except for the last entry. However then R_2 cannot be positive definite, since there are vanishing principal minors (Sylvester's test). ∎

Since the simplicity of the Laplacian matrix is a necessary and sufficient condition for the graph topology constraint (5.43) to be satisfied, one can state the following theorem relating the graph Laplacian matrix (unpinned or pinned) to the global optimality of distributed cooperative control as given in Sections 5.5 and 5.6. The next result if for Section 5.6.

Theorem 5.7. *The distributed control in cooperative regulator problem (5.25) (respectively cooperative tracker problem (5.37)) is optimal under optimality criterion (5.29) (resp. (5.41)) for some $R = R_1 \otimes R_2 > 0$, if and only if the graph Laplacian matrix in (5.26) (resp. (5.38)) is simple.*

Proof. If the control law in the respective instances is optimal then the constraint (5.43) on P, R, L is satisfied, hence, by Theorem 5.6, L is simple. If, conversely L is simple, then there exists a quadratic performance criterion with respect to which the cooperative distributed control given by L is inversely optimal.   ∎

Remark 5.8. These results place constraints on the graph topologies on which the linear distributed protocols can guarantee global optimality with respect to a standard linear quadratic performance criterion. Moreover, if the constraints hold, then the global performance index must be selected to depend on the graph topology. This in turn guarantees asymptotic consensus or synchronization. The following section will allow for a more general form of performance criterion which yields optimal control for general directed graphs.

5.8. Optimal Cooperative Control for General Digraphs: Performance Index with Cross-Weighting Terms

Heretofore, we have seen that, when using the standard quadratic performance index that weights norms of the state and the control input, global optimal performance can only be obtained by linear distributed control protocols only if the graph satisfies a condition. This condition was seen to hold for the class of digraphs with Laplacian matrices having simple Jordan form. In this section we show that if state-control cross-weighting terms are allowed in the performance index, then global optimal performance can be obtained by distributed protocols on any digraph. This indicates that global performance indices with cross-weighting terms are in some sense more natural for cooperative multi-agent systems on communication graph topologies.

When dealing with general directed graphs the constraint (5.12) on the graph Laplacian *L*, or (5.19) on the pinned graph Laplacian $L + G$, is too restrictive. These constraints only hold for a special class of digraphs whose Laplacian matrix (or pinned Laplacian $L + G$) has diagonal Jordan form. This constraint can be relaxed, and optimal cooperative controllers developed for arbitrary digraphs, by allowing state-control cross-weighting terms in the performance criterion [13]. Then, requiring that the performance criterion be positive (semi) definite leads to conditions on kernel matrix P in $V(x) = x^T P x$, or equivalently on the control Lyapunov function, which should be satisfied for the existence of distributed

globally optimal controls. This condition is milder than the conditions (5.12), (5.19) where no cross-weighting term is allowed in the performance index.

In Subsections 5.8.1 and 5.8.2 we treat the optimal cooperative regulator and tracker for single-integrator dynamics, including a state-control cross weighting term in the performance index. In Subsection 5.8.3 we discuss the resulting constraint conditions on the graph topology that must be satisfied by the graph Laplacian for existence of an optimal controller of distributed form. It is shown that, unlike conditions (5.12) and (5.19), the new conditions can be satisfied for arbitrary directed graphs by proper selection of the performance index weighting matrices.

In Subsections 5.8.4 and 5.8.5 we treat the optimal cooperative regulator and tracker for linear time-invariant agent dynamics. Again, it is shown that, if cross-weighting terms are allowed in the performance index, then the conditions required for existence of a distributed optimal controller can be satisfied on arbitrary digraphs.

5.8.1. *Optimal Cooperative Regulator- Single-Integrator Agent Dynamics*

This section considers the cooperative regulator problem for single-integrator agent dynamics $\dot{x}_i = u_i$ or in global form (5.7). Since the consensus manifold is noncompact, it is necessary to use the partial stability results in Theorem 5.1. Using the distributed control protocol $u = -Lx$ gives the closed-loop system $\dot{x} = -Lx$.

The next result extends Theorem 5.2 to the case of state-control weighting terms in the global performance index.

Theorem 5.8. *Let the graph topology be given by L and multi-agent systems be given as (5.7). Then for some $R = R^T > 0$ the control $u = \phi(x) = -Lx$ is optimal with respect to the performance index $J(x_0, u) = \int_0^\infty \mathcal{L}(x, u)dt$ with the performance integrand*

$$\mathcal{L}(x, u) = x^T L^T RLx + u^T Ru + 2x^T (L^T R - P)u \geq 0 \tag{5.47}$$

and is stabilizing to a manifold which is the null space of L if there exists a positive semidefinite matrix $P = P^T \geq 0$, having the same kernel as L, satisfying the inequality condition

$$L^T P + PL - PR^{-1}P \geq 0 \qquad (5.48)$$

Moreover, if the graph contains a spanning tree then consensus is reached.

Proof. Let $V(x) = x^T Px$ be a partial stability Lyapunov function. Then

$$\dot{V}(x) = -2x^T PLx = -x^T(PL + L^T P)x \leq -x^T PR^{-1}Px \leq \sigma_{>0\min}(PR^{-1}P)\|y\|^2$$

where $x = x_0 + y$, $y \in \ker P^{\perp}$. Thus the control is stabilizing to the null space of P. Also the performance integrand equals

$$\mathcal{L}(x,u) = \begin{bmatrix} x & u \end{bmatrix}^T \begin{bmatrix} L^T RL & L^T R - P \\ RL - P & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix},$$

One has $\mathcal{L}(x,u) \geq 0$ if $R > 0$ and the Schur complement inequality (5.48) holds. The control is optimal since

$$u = \phi(x) = -\frac{1}{2}R^{-1}(L_2^T + g^T \nabla V)$$

$$= -\frac{1}{2}R^{-1}(2(RL - P)x + 2Px) = -\frac{1}{2}R^{-1}(2RL)x$$

$$= -Lx$$

The performance integrand evaluated at the optimal control satisfies

$$\mathcal{L}(x,\phi(x)) = x^T L^T RLx + x^T LRLx - 2x^T(L^T R - P)Lx$$

$$= 2x^T L^T RLx - 2x^T L^T RLx + 2x^T PLx$$

$$= x^T(PL + L^T P)x \geq x^T PR^{-1}Px$$

$$\geq \sigma_{\min}(PR^{-1}P)\|y\|^2$$

Hence all the conditions of Lemma 5.2b are satisfied. If the graph has a spanning tree, by Lemma 5.3, zero is a simple eigenvalue of L and thus the null space of P is the consensus manifold $span(\underline{1})$. Then, consensus is reached. This concludes the proof. ∎

Note that if the constraint $P = RL$ in (5.12) is met as in previous sections, the cross-weighting term in $\mathcal{L}(x,u)$ in the proof vanishes, so the performance criterion is quadratic, and therefore $\mathcal{L}(x,u) \geq 0$ That is, substituting $P = RL$ into (5.48) shows that (5.48) is satisfied.

Remark 5.9. The kernels of P and L in (5.48) need not be the same, but under condition (5.48) their relation is given by the following result.

Proposition. *Let there exist a symmetric positive semidefinite P satisfying (5.48) for some L, and $R > 0$ then* $\ker L \subseteq \ker P$.

The proof of this proposition uses a result on symmetric positive semidefinite matrices summarized in the following lemma, which is not difficult to prove.

Lemma 5.5. *Given a positive semidefinite symmetric matrix $Q \geq 0$ the kernel of Q equals the set where the pertaining quadratic form satisfies $x^T Q x = 0$.*

Note that this result, albeit straightforward, is not trivial since a nonsymmetric matrix Q generally does not share this property.

Now, to prove the proposition notice that under (5.48) $L^T P + PL - PR^{-1}P$ is a symmetric positive semidefinite matrix. Consider the quadratic form

$$x^T (L^T P + PL - PR^{-1}P)x \geq 0 .$$

Assume $x \in \ker L$ but $x \notin \ker P$, then for such an x

$$\underbrace{x^T L^T Px + x^T PLx}_{=0} - x^T PR^{-1}Px \geq 0 \Rightarrow -x^T PR^{-1}Px = (Px)^T R^{-1} (Px) \geq 0 .$$

However, since $R > 0$ this forces $Px = 0$, whence it follows $x \in \ker P$ which is a contradiction. Therefore one has $x \in \ker L \Rightarrow x \in \ker P$ meaning $\ker L \subseteq \ker P$, which proves the proposition. More discussion about the condition (5.48) is provided in Subsection 5.8.3.

5.8.2. *Optimal Cooperative Tracker- Single-Integrator Agent Dynamics*

In case of the optimal cooperative tracker with pinning from a leader node for the single-integrator agent dynamics system (5.7), one has the global disagreement error dynamics (5.17), which must be stabilized to the origin. This means that partial stability notions need not be used, and the sought matrix P must be positive definite. Therefore, for the optimal cooperative tracker problem the applied logic is the same, the only difference being that positive semidefiniteness is replaced by positive definiteness in the performance index and Lyapunov function.

Theorem 5.9. *Let the graph topology be $(L+G)$, the error dynamics be given as (5.17), and the conditions of Lemma 5.4 be satisfied. Then for some $R = R^T > 0$ the control $u = \phi(\delta) = -(L+G)\delta$ is optimal with respect to the performance index*

$$J(\delta_0, u) = \int_0^\infty \mathcal{L}(\delta, u)dt \text{ with the performance integrand}$$

$$\mathcal{L}(\delta, u) = \delta^T (L+G)^T R(L+G)\delta + u^T Ru + 2\delta^T ((L+G)^T R - P)u > 0 \quad (5.49)$$

and synchronization is reached if there exists a positive definite matrix $P = P^T \geq 0$, satisfying the inequality

$$(L+G)^T P + P(L+G) - PR^{-1}P > 0 \quad (5.50)$$

Proof. : Let $V(\delta) = \delta^T P\delta$ be a Lyapunov function. Then

$$\dot{V}(\delta) = -2\delta^T P(L+G)\delta = -\delta^T (P(L+G) + (L+G)^T P)\delta < -\delta^T PR^{-1}P\delta < 0.$$

Thus the control is stabilizing to the origin for (5.17) and synchronization is reached. Also the performance integrand equals

$$\mathcal{L}(\delta, u) = \begin{bmatrix} \delta & u \end{bmatrix}^T \begin{bmatrix} (L+G)^T R(L+G) & (L+G)^T R - P \\ R(L+G) - P & R \end{bmatrix} \begin{bmatrix} \delta \\ u \end{bmatrix} > 0$$

and $\mathcal{L}(x, u) > 0$ if $R > 0$ and inequality (5.50) holds (by Schur complement). The control is optimal since

$$u = \phi(\delta) = -\frac{1}{2}R^{-1}(L_2^T + g^T \nabla V)$$

$$= -\frac{1}{2}R^{-1}(2(R(L+G) - P)\delta + 2P\delta) = -\frac{1}{2}R^{-1}(2R(L+G))\delta$$

$$= -(L+G)\delta$$

The performance integrand evaluated at the optimal control satisfies

$$\mathcal{L}(\delta, \phi(\delta)) = \delta^T (L+G)^T R(L+G)\delta + \delta^T (L+G)^T R(L+G)\delta$$

$$- 2\delta^T ((L+G)^T R - P)(L+G)\delta$$

$$= 2x\delta^T (L+G)^T R(L+G)\delta - 2\delta^T (L+G)^T R(L+G)\delta + 2\delta^T P(L+G)\delta$$

$$= \delta^T (P(L+G) + (L+G)^T P)\delta > \delta^T PR^{-1}P\delta > 0$$

Hence all the conditions of Lemma 5.2b are satisfied, which concludes the proof. ∎

Note that if the constraint $P = R(L + G)$ in (5.19) is met as in previous sections, the cross-weighting term in $\mathcal{L}(\delta, u)$ in the proof vanishes, so that the performance criterion is quadratic and therefore $\mathcal{L}(x, u) > 0$. That is, $P = R(L + G)$ guarantees that (5.50) is satisfied. More discussion about the condition (5.50) is provided in Subsection 5.8.3.

5.8.3. Condition for Existence of Global Optimal Control with Cross-weighting Terms in the Performance Index

In this section we show that conditions (5.48) and (5.50) can be met for arbitrary digraphs, in contrast to (5.12) or (5.19) which only hold for digraphs with simple Laplacian matrices. This indicates that global performance indices with cross-weighting terms are in some sense more natural for cooperative multi-agent systems on communication graph topologies.

Conditions (5.48) and (5.50) must be met for the existence of a distributed linear control protocol that solves a global optimality problem on the prescribed graph. These conditions allow state-control cross-weighting terms in the performance index, and express joint constraints on the graph matrices L, or $L + G$, and the choice of control weighting matrix R. These conditions are less strict than the conditions (5.12) and (5.19), respectively, that guarantee the existence of a global optimal controller of distributed form when no cross-weighting term is allowed. In particular, note that if condition (5.12) or (5.19) hold then, respectively, (5.48) and (5.50) hold as well, and the cross-weighting terms equal zero. The additional freedom stemming from allowing the presence of cross-weighting terms in the performance integrand allows for more general graph topologies excluded under the Riccati conditions (5.12) or (5.19) to support inverse optimality of linear distributed control protocols.

Conditions (5.12) and (5.19) are only satisfied by digraphs that have a simple Jordan form for L or $L+G$, respectively. This includes undirected graphs and the detail balanced digraphs. It is now shown that conditions (5.48) and (5.50) can be satisfied for arbitrary digraphs.

For arbitrary digraphs, a matrix P having the same kernel as L and satisfying the optimal cooperative regulator condition (5.48) can be constructed as follows. To find $P = P^T \geq 0$ such that $L^T P + PL - PR^{-1}P \geq 0$ and $\ker P = \ker L$ one can solve the equivalent inequality

$$-L^T P - PL \leq -PR^{-1}P \,,$$

by solving an intermediate Lyapunov equation

$$-L^T P - PL = -Q \le -PR^{-1}P \,, \tag{5.51}$$

with $Q = Q^T \ge 0$, $\ker Q = \ker L$. The solution P to this Lyapunov equation exists since $-L$ is stabilizing to its nullspace, and equals

$$P = \int_0^\infty e^{-L^T \tau} Q e^{-L\tau} d\tau \ge 0$$

.

That $P = P^T$ follows from the construction, and if L has a simple zero eigenvalue, which is necessary for consensus, $\ker P = \ker Q = \ker L$. Since P is bounded for any fixed Q, and the solution P does not depend on R, choosing

R big enough, in the sense of any matrix norm, will make $PR^{-1}P \le Q$. Therefore, such a constructed P satisfies condition (5.48).

A similar method can be used to construct, for any digraph, the required matrix P for the optimal tracker condition (5.50).

5.8.4. *General Linear Time-Invariant Systems- Cooperative Regulator*

This section deals with agents having states $x_i \in \mathbb{R}^n$ with drift dynamics $\dot{x}_i = Ax_i + Bu_i \in \mathbb{R}^n$, or in global form (5.22). Since the synchronization manifold, null space of $L \otimes I_n$, is noncompact, the partial stability Theorem 5.1 must be used. The next result extends Theorem 5.4 to the case of state-control cross-weighting terms.

Theorem 5.10. *Let the graph topology be given by L and multi-agent systems be given as (5.22). Define the local feedback matrix to be K_2 such that the cooperative feedback control $u = -cL \otimes K_2 x$, with a scalar coupling gain $c > 0$, makes (5.22) asymptotically converge to the consensus manifold, i.e. achieve consensus. Then there exists a positive semidefinite matrix $P = P^T \ge 0$, satisfying*

$$c^2 (L \otimes K_2)^T R(L \otimes K_2) - P(I_N \otimes A) - (I_N \otimes A^T)P \ge 0 \tag{5.52}$$

$$P(cL \otimes BK_2 - (I_N \otimes A)) + (cL \otimes BK_2 - (I_N \otimes A))^T P$$
$$-P(I_N \otimes B)R^{-1}(I_N \otimes B^T)P \geq 0 \tag{5.53}$$

for some $R = R^T > 0$. Moreover, the control $u = -cL \otimes K_2 x$ is optimal with respect to the performance index $J(x_0, u) = \int_0^\infty \mathcal{L}(x, u)dt$ with the performance integrand

$$\mathcal{L}(x,u) = x^T \left[c^2 (L \otimes K_2)^T R(L \otimes K_2) - P(I_N \otimes A) - (I_N \otimes A^T)P \right] x + u^T Ru$$
$$+ 2x^T \left[c(L \otimes K_2)^T R - P(I_N \otimes B) \right] u \geq 0$$

and is stabilizing to the null space of P for sufficiently high coupling gain c. Finally, if the graph has a spanning tree, then consensus is reached.

Proof. Since the cooperative feedback is assumed to make (5.22) synchronize the closed-loop system matrix $A_{cl} = I_N \otimes A - cL \otimes BK_2$ defines a partially stable system with respect to the consensus manifold. This manifold equals the kernel of $L \otimes I_n$ if L contains a spanning tree. Then for any positive semidefinite matrix $Q = Q^T \geq 0$, having the kernel equal to the consensus manifold, there exists a solution $P = P^T \geq 0$ to the Lyapunov equation

$$PA_{cl} + A_{cl}^T P = -Q$$

$$P = \int_0^\infty e^{A_{cl}^T \tau} Q e^{A_{cl}\tau} d\tau \geq 0$$

with kernel equal to the consensus manifold.

The inequality (5.52) is satisfied for sufficiently high values of the coupling gain $c > 0$, since for sufficiently large c the first term dominates the second one everywhere, except on the nullspace of $L \otimes K_2$. On this nullspace the control signal vanishes. According to the assumption on partial asymptotic stability of the consensus manifold under the chosen feedback control, and the Lyapunov equation, the second term in (5.52) is positive semidefinite when evaluated on the nullspace of $L \otimes K_2$. Hence the inequality (5.52) is satisfied everywhere. The inequality (5.53) is satisfied via Lyapunov equation by choosing $R = R^T > 0$ sufficiently big, in the sense of matrix norms, such that $0 \leq P(I_N \otimes B)R^{-1}(I_N \otimes B^T)P \leq Q$.

Let $V(x) = x^T P x$ be a partial stability Lyapunov function. Then

$$\dot{V}(x) = 2x^T P\dot{x} = 2x^T P((I_N \otimes A)x - c(L \otimes BK_2)x)$$

$$= x^T (P(I_N \otimes A) + (I_N \otimes A^T)P)x - cx^T P(L \otimes BK_2)x - cx^T (L \otimes BK_2)^T Px$$

$$= x^T \left[P(I_N \otimes A - cL \otimes BK_2) + (I_N \otimes A - cL \otimes BK_2)^T P \right] x$$

$$= x^T \left[PA_{cl} + A_{cl}^T P \right] x = -x^T Qx \le -x^T P(I_N \otimes B)R^{-1}(I_N \otimes B^T)Px \le 0$$

The quadratic form $x^T Qx \ge 0$ serves as a measure of distance from the consensus manifold. Thus the control is stabilizing to the consensus manifold, as assumed. Also the performance integrand equals

$$\mathcal{L}(x,u)$$

$$= \begin{bmatrix} x & u \end{bmatrix}^T \begin{bmatrix} c^2(L \otimes K_2)^T R(L \otimes K_2) - P(I_N \otimes A) - (I_N \otimes A^T)P & c(L \otimes K_2)^T R - P(I_N \otimes B) \\ cR(L \otimes K_2) - (I_N \otimes B)P & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

$$\ge 0$$

and $\mathcal{L}(x,u) \ge 0$ by Schur complement if inequalities (5.52) and (5.53) hold. The feedback control is optimal since

$$u = \phi(x) = -\frac{1}{2}R^{-1}(L_2^T + g^T \nabla V)$$

$$= -\frac{1}{2}R^{-1}(2(cR(L \otimes K_2) - (I_N \otimes B^T)P)x + 2(I_N \otimes B^T)Px)$$

$$= -\frac{1}{2}R^{-1}(2cR(L \otimes K_2))x = -c(L \otimes K_2)x$$

The performance integrand evaluated at the optimal control satisfies

$$\mathcal{L}(x,\phi(x)) = x^T \left[c^2(L \otimes K_2)^T R(L \otimes K_2) - P(I_N \otimes A) - (I_N \otimes A^T)P \right] x$$

$$+ x^T c^2(L \otimes K_2)^T R(L \otimes K_2)x - 2cx^T \left[c(L \otimes K_2)^T R - P(I_N \otimes B) \right](L \otimes K_2)x$$

$$= -x^T \left[P(I_N \otimes A) + (I_N \otimes A^T)P \right] x + 2cx^T P(I_N \otimes B)(L \otimes K_2)x$$

$$= -x^T \left[P(I_N \otimes A) + (I_N \otimes A^T)P \right] x + 2cx^T P(L \otimes BK_2)x$$

$$= x^T \left[(cL \otimes BK_2 - I_N \otimes A)^T P + P(cL \otimes BK_2 - I_N \otimes A) \right] x$$

$$= x^T \left[-A_{cl}^T P - PA_{cl} \right] x = x^T Qx \geq x^T P(I_N \otimes B)R^{-1}(I_N \otimes B^T)Px \geq 0$$

Hence all the conditions of Lemma 5.2b are satisfied.

If the communication graph has a spanning tree, then by Lemma 5.3 the null space of $L \otimes I_n$ equals the synchronization manifold $S := span(\underline{1} \otimes \alpha)$, $\alpha \in \mathbb{R}^n$, therefore synchronization is asymptotically achieved in the optimal way.

This concludes the proof. ∎

Specifying the form of P and R as $P = P_1 \otimes P_2$, $R = R_1 \otimes R_2$ and assuming further that $K_2 = R_2^{-1}B^T P_2$, where P_2 is a solution of local Riccati equation $P_2 A + A^T P_2 + Q_2 - P_2 B R_2^{-1} B^T P_2 = 0$ as used in Theorems 5.4 and 5.5, the inequality (5.53) becomes

$$cP_1 L \otimes P_2 BK_2 - P_1 \otimes (P_2 A + A^T P_2) + cL^T P_1 \otimes K_2^T B^T P_2$$

$$- P_1 R_1^{-1} P_1 \otimes P_2 B R_2^{-1} B^T P_2 \geq 0 \qquad .$$

Equivalently, since

$$P_2 BK_2 = K_2^T B^T P_2 = P_2 B R_2^{-1} B^T P_2 = K_2^T R_2 K_2 \geq 0 \quad ,$$

one has

$$(cP_1 L + cL^T P_1 - P_1 R_1^{-1} P_1) \otimes K^T R_2 K - P_1 \otimes (K^T R_2 K - Q_2) \geq 0 \quad .$$

Note the similarity to condition (5.48) introduced in single integrator consensus problem Subsection 5.8.1. If condition (5.48) is satisfied then for sufficiently high value of the coupling gain c this constraint can be met.

Clearly if Riccati conditions (5.26), (5.27) are satisfied then the cross-weighting term vanishes

$$cR(L \otimes K_2) = (I_N \otimes B^T)P \Leftrightarrow c(L \otimes K_2) = R^{-1}(I_N \otimes B^T)P \quad .$$

One should also note that choosing $K_2 = R_2^{-1} B^T P_2$ affords an infinite interval of positive values for coupling gain c that achieve synchronization [28], which allows one to find a sufficiently high value of c to satisfy inequality (5.52) without worrying about losing synchronization. The same applies whenever one needs to choose c to be, for any purpose, sufficiently large.

5.8.5. General Linear Time-Invariant Systems- Cooperative Tracker

If the goal is to synchronize dynamics (5.22) to the leader's trajectory $\dot{x}_0 = A x_0$, *i.e.* solve the cooperative tracking problem, then one should use the global error system $\delta = x - \underline{1} \otimes x_0$, with the same global error dynamics

$$\dot{\delta} = (I_N \otimes A)\delta + (I_N \otimes B)u . \tag{5.54}$$

Theorem 5.11. *Let the graph topology be given by $(L+G)$, the conditions of Lemma 5.4 be satisfied. and multi-agent systems be given as (5.54). Define the local feedback matrix to be K_2 such that the cooperative feedback control $u = -c(L+G) \otimes K_2 \delta$, with a scalar coupling gain $c > 0$, makes (5.54) asymptotically converge to the origin. Then there exists a positive definite matrix $P = P^T > 0$, satisfying*

$$c^2((L+G) \otimes K_2)^T R((L+G) \otimes K_2) - P(I_N \otimes A) - (I_N \otimes A^T)P > 0 \tag{5.55}$$

$$P(c(L+G) \otimes BK_2 - (I_N \otimes A)) + (c(L+G) \otimes BK_2 - (I_N \otimes A))^T P$$
$$-P(I_N \otimes B)R^{-1}(I_N \otimes B^T)P > 0 \tag{5.56}$$

for some $R = R^T > 0$. Moreover, the control $u = -c(L+G) \otimes K_2 \delta$ is optimal with respect to the performance index $J(\delta_0, u) = \int_0^\infty \mathcal{L}(\delta, u)dt$ with the performance integrand

$$\mathcal{L}(\delta, u) = \delta^T \left[c^2 ((L+G) \otimes K_2)^T R((L+G) \otimes K_2) - P(I_N \otimes A) - (I_N \otimes A^T)P \right] \delta + u^T R u$$

$$+2\delta^T \left[c((L+G) \otimes K_2)^T R - P(I_N \otimes B) \right] u > 0$$

.

Proof. Since the cooperative feedback is assumed to stabilize (5.54) to the origin the closed-loop system matrix $A_{cl} = I_N \otimes A - c(L+G) \otimes BK_2$ defines a stable system. Then for any positive definite matrix $Q = Q^T > 0$, there exists a solution $P = P^T > 0$ to the Lyapunov equation

$$PA_{cl} + A_{cl}^T P = -Q$$

that is given by

$$P = \int_0^\infty e^{A_{cl}^T \tau} Q e^{A_{cl} \tau} d\tau > 0$$

The inequality (5.55) is satisfied for sufficiently high values of the coupling gain $c > 0$, since for sufficiently large c the first term in (5.55) dominates the second one everywhere. The inequality (5.56) is satisfied via Lyapunov equation by choosing $R = R^T > 0$ sufficiently big, in the sense of matrix norms, such that $0 \leq P(I_N \otimes B)R^{-1}(I_N \otimes B^T)P < Q$.

Let $V(\delta) = \delta^T P \delta > 0$ be a Lyapunov function. Then

$$\dot{V}(\delta) = 2\delta^T P \dot{\delta} = 2\delta^T P((I_N \otimes A)\delta - c((L+G) \otimes BK_2)\delta)$$

$$= \delta^T (P(I_N \otimes A) + (I_N \otimes A^T)P)\delta - c\delta^T P((L+G) \otimes BK_2)\delta - c\delta^T ((L+G) \otimes BK_2)^T P\delta$$

$$= \delta^T (P(I_N \otimes A - c(L+G) \otimes BK_2) + (I_N \otimes A - c(L+G) \otimes BK_2)^T P)\delta$$

$$= \delta^T (PA_{cl} + A_{cl}^T P)\delta = -\delta^T Q\delta < -\delta^T P(I_N \otimes B)R^{-1}(I_N \otimes B^T)P\delta \leq 0.$$

Thus the control is stabilizing to the origin, as assumed. Also the performance integrand equals

$$\mathcal{L}(\delta, u) =$$

$$\begin{bmatrix} \delta & u \end{bmatrix}^T \begin{bmatrix} c^2((L+G)\otimes K_2)^T R((L+G)\otimes K_2) - P(I_N\otimes A) - (I_N\otimes A^T)P & c((L+G)\otimes K_2)^T R - P(I_N\otimes B) \\ cR((L+G)\otimes K_2) - (I_N\otimes B^T)P & R \end{bmatrix} \begin{bmatrix} \delta \\ u \end{bmatrix}$$

$$> 0$$

and $\mathcal{L}(\delta, u) > 0$ by Schur complement if inequalities (5.55), (5.56) hold. The control is optimal since

$$u = \phi(\delta) = -\frac{1}{2}R^{-1}(L_2^T + g^T\nabla V)$$

$$= -\frac{1}{2}R^{-1}(2(cR((L+G)\otimes K_2) - (I_N\otimes B^T)P)\delta + 2(I_N\otimes B^T)P\delta)$$

$$= -\frac{1}{2}R^{-1}(2cR((L+G)\otimes K_2))\delta$$

$$= -c(L+G)\otimes K_2\delta$$

The performance integrand evaluated at the optimal control satisfies

$$\mathcal{L}(\delta, \phi(\delta)) = \delta^T\left[c^2((L+G)\otimes K_2)^T R((L+G)\otimes K_2) - P(I_N\otimes A) - (I_N\otimes A^T)P \right]\delta$$

$$+ \delta^T c^2((L+G)\otimes K_2)^T R((L+G)\otimes K_2)\delta$$

$$- 2c\delta^T\left[c((L+G)\otimes K_2)^T R - P(I_N\otimes B) \right]((L+G)\otimes K_2)\delta$$

$$= -\delta^T\left[P(I_N\otimes A) + (I_N\otimes A^T)P \right]\delta + 2c\delta^T P(I_N\otimes B)((L+G)\otimes K_2)\delta$$

$$= -\delta^T\left[P(I_N\otimes A) + (I_N\otimes A^T)P \right]\delta + 2c\delta^T P((L+G)\otimes BK_2)\delta$$

$$= \delta^T\left[(c(L+G)\otimes BK_2 - I_N\otimes A)^T P + P(c(L+G)\otimes BK_2 - I_N\otimes A) \right]\delta$$

$$= \delta^T\left[-A_d^T P - PA_d \right]\delta = \delta^T Q\delta > \delta^T P(I_N\otimes B)R^{-1}(I_N\otimes B^T)P\delta \geq 0$$

Hence all the conditions of Lemma 5.2b are satisfied, which concludes the proof. ∎

Specifying the form of P and R as $P = P_1 \otimes P_2$, $R = R_1 \otimes R_2$ and assuming further that $K_2 = R_2^{-1}B^T P_2$, where P_2 is a solution of local Riccati equation $P_2 A + A^T P_2 + Q_2 - P_2 B R_2^{-1} B^T P_2 = 0$, as used in Theorems 5.4 and 5.5, the inequality (5.56) becomes

$$cP_1(L+G) \otimes P_2 B K_2 - P_1 \otimes (P_2 A + A^T P_2) + c(L+G)^T P_1 \otimes K_2^T B^T P_2$$
$$- P_1 R_1^{-1} P_1 \otimes P_2 B R_2^{-1} B^T P_2 > 0$$

.

Equivalently, since

$$P_2 B K_2 = K_2^T B^T P_2 = P_2 B R_2^{-1} B^T P_2 = K_2^T R_2 K_2 \geq 0,$$

one has

$$(cP_1(L+G) + c(L+G)^T P_1 - P_1 R_1^{-1} P_1) \otimes K_2^T R_2 K_2$$
$$- P_1 \otimes (K_2^T R_2 K_2 - Q_2) > 0$$

.

Note the similarity to condition (5.50) introduced in single integrator consensus problem Subsection 5.8.2. If condition (5.50) is satisfied then for sufficiently high value of the coupling gain c this constraint can be met.

Remark 5.10. Due to the connection of the inequality here and the one in the preceding section with inequalities for single integrator systems (5.48) and (5.50), respectively, the comments in Section 5.8.3 on constructing the suitable P_1 matrix apply here as well.

Clearly if Riccati conditions (5.38), (5.39) are satisfied then the cross-weighting term vanishes, for

$$cR((L+G) \otimes K_2) = (I_N \otimes B^T)P \Leftrightarrow c((L+G) \otimes K_2) = R^{-1}(I_N \otimes B^T)P.$$

One should also note that choosing $K_2 = R_2^{-1}B^T P_2$ affords an infinite interval of positive values for coupling gain c that achieve stabilization, [28], which allows one to find a sufficiently high value of c to satisfy inequality (5.55) without worrying about losing stability of the closed-loop system.

5.9.   Conclusion

In this chapter we consider the global optimality of local distributed control protocols in terms of the overall performance of the team of all agents. It is seen that globally optimal performance using standard LQR performance measures cannot in general be guaranteed by using distributed control protocols for arbitrary digraphs. For linear distributed agent control protocols to yield global optimal team behaviors, the graph topology must satisfy certain conditions. These conditions are shown to define a new class of digraphs that admit global optimal control design using distributed control protocols- namely, the digraphs whose Laplacian matrix has a diagonal Jordan form. If this condition is satisfied, moreover, any admissible global performance indices must be selected in a certain manner that depends on the graph topology. This clarifies the interactions between stability, optimal performance, and the communication graph topology in multi-agent systems. In Section 5.8 a more general LQR performance index is considered which has state-control cross-weighting. It is shown that, with proper choice of this performance index, optimal cooperative control in distributed form can be achieved for any directed graph. This shows that performance indices with state-control cross-weighting terms are in some sense more natural for cooperative control on communication graphs.

References

[1]   Bernstein, D.S (2009). *Matrix Mathematics, Theory Facts, and Formulas.* (Corollary 5.3.8.) Princeton University Press, Princeton and Oxford.

[2]   Borelli, F. & Keviczky, T. (2008). Distributed LQR Design for Identical Dynamically De-coupled Systems. *IEEE Transactions on Automatic Control*, 53(8), pp. 1901-1912.

[3]   Cao, Y. & Ren, W. (2010). Optimal Linear-Consensus Algorithms: An LQR Perspective. *IEEE Transactions on Systems Man, and Cybernetics*, 40(3), pp. 819-830.

[4]   Dong, W. (2010). Distributed optimal control of multiple systems. *International Journal of Control*, 83(10), pp. 2067-2079.

[5]   Dunbar, W.B., & Murray, R.M. (2006). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4), pp. 549-558.

[6]   Fax, J.A., & Murray, R. M. (2004), Information Flow and Cooperative Control of Vehicle Formations. *IEEE Transactions on Automatic Control*, 49(9), pp. 1465-1476.

[7]   Haddad, W.M., & Chellaboina, V. (2008). *Nonlinear Dynamical Systems and Control*, *A Lyapunov-Based Approach,* Princeton University Press, Princeton and Oxford, ch. 4.

[8]   K. Hengster-Movric and F.L. Lewis, "Cooperative Optimal Control for Multi-agent Systems on Directed Graph Topologies", Int. J. Robust and Nonlinear Control, 2013.

[9]   Hong, Y., Wang, X., & Z. Jiang. (2010). Multi-Agent Coordination with General Linear Models: A Distributed Output Regulation Approach. 8th *IEEE International conference on Control and Automation (ICCA)*, pp. 137-142.

[10] Jadbabaie, A., Lin, J., & Morse, A. (2003). Coordination of groups of mobile autonomous agents using nearest neighbour rules. *IEEE Transactions on Automatic Control*, 48(6), pp. 988–1001.

[11] Jovanovic, M.R. On the optimality of localized distributed controllers. (2005) *American Control Conference, June 8-10, 2005, Portland, OR,USA*, pp. 4583-4588.

[12] Khoo, S., Xie, L., & Man, Z. (2009). Robust finite-time consensus tracking algorithm for multirobot systems. *IEEE/ASME Transactions on Mechatronics*, 14(2), pp. 219-228.

[13] Lewis, F.L, Vrabie, D., Syrmos, V.L. (2012). *Optimal Control,* Third edition, John Wiley & Sons.

[14] Li, Z., Duan, Z., Chen, G., & Huang, L. (2010). Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint. *IEEE Transactions on Circuits and Systems I, Reg. Papers,* 57(1), pp. 213–224.

[15] Ma, C.Q., & Zhang, J.F.(2010). Necessary and Sufficient Conditions for Consensusability of Linear Multi-Agent Systems. *IEEE Transactions on Automatic Control*, 55, pp. 1263-1268.

[16] Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and Cooperation in Networked Multi-Agent Systems (invited paper). *Proceedings of the IEEE*, 95(1), pp. 215-233.

[17] Olfati-Saber, R., & Murray, R. M. (2003). Consensus protocols for networks of dynamic agents. *Proceedings of American Control Conference*, pp. 951-956.

[18] Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9), pp. 1520-1533.

[19] Qu, Z., Simaan, M. & Doug, J. (2009). Inverse Optimality of Cooperative Control for Networked Systems. *Joint 48th IEEE Conference on Decision and Control and28th Chinese Control Conference*, *Shanghai, P.R. China*, December 16-18, 2009.

[20] Ren, W., & Beard, R.W. (2005). Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50, pp. 655–661.

[21] Semsar-Kazerooni, E., & Khorasani, K. (2009). Multi-agent team cooperation: A game theory approach. *Automatica,* 45(10), pp. 2205-2213.

[22] Slotine, J.J. & Lee, W. *Applied Nonlinear Control*, New Jersey, Prentice Hall, ch. 4.

[23] Sontag, E.D. (1998). *Mathematical control theory*, New York: Springer.

[24] Tuna, S. E. (2008). LQR-based coupling gain for synchronization of linear systems. *arXiv:0801.3390v1 [math.OC]*.

[25] Vamvoudakis, K.G. & Lewis, F.L. (2011). Policy Iteration Algorithm for Distributed Networks and Graphical Games. *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, USA, December 12-15, 2011*.

[26] Wang, X. F., & Chen, G. (2002). Pinning control of scale free dynamical networks. *Physica A*, 310, pp. 521-531.

[27] Zhang, H., & Lewis, F. L. (2011). "Optimal Design for Synchronization of Cooperative Systems: State Feedback, Observer and Output Feedback", *IEEE Transactions on Automatic Control*, 56(8), pp. 1948-1953.

[28] Zhang, H., & Lewis, F. L. (2011). Lyapunov, Adaptive, and Optimal Design Techniques for Cooperative Systems on Directed Communication Graphs. *IEEE Transaction on Industrial Electronics*, 59(7), pp. 3026-3041.

6
Graphical Games: Distributed Multi-player Games on Graphs

In Chapters 3 and 4 we showed that locally optimal Riccati design of cooperative control protocols can guarantee synchronization of multi-agent systems on communication graphs. The control protocols turned out to be distributed in the sense that the control of each agent only depends on the allowable information available in the graph, namely, information about the agent and its neighbors. By contrast, in Chapter 5 we saw that for a linear distributed protocol to be globally optimal for multi-agent teams on graphs, the global performance index must be selected to depend on the graph topology. Specifically, global performance meaures should depend on the graph Laplacian matrix. Moreover, the standard linear quadratic regulator (LQR) problem does not have a linear distributed protocol solution unless the graph is of a certain class, namely, the graph Laplacian matrix must be simple- it must have a diagonal Jordan form.

In this chapter it is seen that distributed control protocols that both guarantee synchronization and are globally optimal for the multi-agent team always exist on any sufficiently communication graph if a different definition of optimality is used. To this end, we study the notion of Nash equilibrium for multi-player games on graphs. This leads us to the idea of a new sort of differential game- graphical games. In graphical games, each agent has its own dynamics as well as its own local performance index. The dynamics and local performance indices of each agent are distributed; they depend on the state of the agent, thecontrol of the agent, and the controls of the agent's neighbors. We show how tocompute distributed control protocols that guarantee global Nash equilibrium for multi-agent teams on any graph that has a spanning tree.

Teams of multi-agent systems arise in several domains of engineering and can be used to solve problems which are difficult for an individual agent to solve. Strategies for team decision problems, including optimal control, N-player games (H-infinity control, non-zero sum), and so on are normally solved for off-line by solving associated matrix equations such as the coupled Riccati equations or coupled Hamilton-Jacobi equations. However, using that approach players cannot change their objectives online in real time without calling for a completely new off-line solution for the new strategies. Therefore, in this chapter we bring together cooperative control, game theory, and reinforcement learning to present a multi-agent formulation for online solution of team games.

The notion of graphical games is developed for dynamical multi-agent systems, where both the dynamics and the performance indices for each node depend only on local neighbor information. It is shown that standard definitions for Nash equilibrium are not sufficient for graphical games and a new definition of "Interactive Nash Equilibrium" is given. Based on reinforcement learning orinciples, we give a cooperative policy iteration algorithm for graphical games that converges to the best response when the neighbors of each agent do not update their policies, and to the cooperative Nash equilibrium when all agents update their policies simultaneously. This is used to develop reinforcement learning methods for online adaptive learning solutions of graphical games in real time along with proofs of stability and convergence.

6.1   Introduction: Games, Reinforcement Learning, and Policy Iteration

Game theory provides an ideal environment in which to study multi-player decision and control problems, and offers a wide range of challenging and engaging problems. Game theory[31],[5],[29] has been successful in modeling strategic behavior, where the outcome for each player depends on the actions of himself and all the other players. Every player chooses a control to minimize independently from the others his own performance objective. Multi-player cooperative games rely on solving coupled Hamilton-Jacobi (HJ) equations, which in the linear quadratic case reduce to the coupled algebraic Riccati equations [1],[5],[11]. These coupled equations are difficult to solve. Solution methods are generally offline and generate fixed control policies that are then implemented in online controllers in real time.

Reinforcement learning (RL) is a sub-area of machine learning concerned with how to methodically modify the actions of an agent (player) based on observed responses from its environment [30]. RL methods have allowed control systems researchers to develop algorithms to learn online in real time the solutions to optimal control problems for dynamic systems that are described by difference or ordinary differential equations. These involve computational intelligence techniques known as Policy Iteration (PI) and Vakue Iteration [6],[30], which refer to a class of algorithms with two steps, *policy evaluation* and *policy improvement*. These methods have primarily been developed for discrete-time systems, and implementation for control systems design has been developed through approximation of the value function [6],[40],[41]. Neurodynamic Programming [6], finds optimal control solutions by offline computation using value function appxorimation (VFA) and policy iteration. The approximate dynamic programming (ADP) methods of Werbos [40],[41] show how to use VFA and value iteration to compute optimal control solutions online in real time using data measurements available along the system trajectories. Thus, RL provides effective means of learning solutions to HJ equations online. Applications of ADP are further developed in [3],[4],[9] and surveyed in [38]. In control theoretic terms, the PI algorithm amounts to learning the solution to a Lyapunov equation, and then updating the policy through minimizing a Hamiltonian function. Practical methods for applying PI and value iteration to continuous-time systems have been developed in [2] and using the integral reinforcement learning methods in [35],[36].

RL methods have been used to solve multiplayer games for finite-state systems in [8], [25]. RL has been applied to learn online in real-time the solutions for optimal control problems for dynamic systems and differential games in [9],[14],[32],[33].

This chapter brings together cooperative control, reinforcement learning, and game theory to study multi-player differential games on communication graph topologies. A new class of differential games is introduced- the graphical games. In graphical games, each agent has its own dynamics as well as its own local performance index. The dynamics and local performance indices of each agent are distributed; they depend on the state of the agent, thecontrol of the agent, and the controls of the agent's neighbors. This graphical game allows for synchronization as well as Nash equilibrium solutions on communication graphs. It is shown that standard definitions for Nash equilibrium are not sufficient for graphical games and a new definition of "Interactive Nash Equilibrium" is given. We develop coupled Riccati equations for solution of graphical games. A Policy Iteration algorithm for solution of graphical games is given that relies only on local information from neighbor nodes. It is shown that this algorithm converges to the best response policy of a node if its neighbors have fixed policies, and to the global Nash solution if all nodes update their policies. Finally, an online adaptive learning algorithm is presente for computing the Nash equilibrium solutions of graphical games. The approach here is based on VFA using neural network adaptive control as given in [20].

The material in this chapter is from [34].

6.2   Synchronization and Node Error Dynamics

In this section we recall some graph theory basics. We introduce the synchronization problem for multi-agent systems connected by a communication graph. The local neighborhood synchronization error dynamics are derived. These error dynamics are the basis for the definition of graphical games in the next section.

6.2.1 Graphs

Consider a graph $Gr = (V, E)$ with a nonempty finite set of N nodes $V = \{v_1, \cdots, v_N\}$ and a set of edges or arcs $E \subseteq V \times V$. We assume the graph is simple, e.g. no repeated edges and $(v_i, v_i) \notin E, \forall i$ no self loops. Denote the connectivity matrix as $E = [e_{ij}]$ with $e_{ij} > 0$ *if* $(v_j, v_i) \in E$ and $e_{ij} = 0$ otherwise. Note $e_{ii} = 0$. The set of neighbors of a node v_i is $N_i = \{v_j : (v_j, v_i) \in E\}$, i.e. the set of nodes with arcs incoming to v_i. Define the in-degree matrix as a diagonal matrix $D = diag(d_i)$ with $d_i = \sum_{j \in N_i} e_{ij}$ the weighted in-degree of node i (i.e. i-th row sum of E). Define the graph Laplacian matrix as $L = D - E$, which has all row sums equal to zero.

A directed path is a sequence of nodes $v_0, v_1, \cdots, v_r$ such that $(v_i, v_{i+1}) \in E, i \in \{0, 1, \cdots, r-1\}$. A directed graph is strongly connected if there is a directed path from v_i to v_j for all distinct nodes $v_i, v_j \in V$. A (directed) tree is a connected digraph where every node except one, called the root node, has in-degree equal to one. A graph is said to have a spanning tree if a subset of the edges forms a directed tree. A strongly connected digraph contains a spanning tree. In a strongly connected graph, all nodes are root nodes.

General directed graphs with fixed topology are considered in this chapter.

6.2.2 Synchronization and Node Error Dynamics

Consider the *N* systems or agents distributed on communication graph *Gr* with node dynamics

$$\dot{x}_i = Ax_i + B_i u_i.$$

(6.1)

where $x_i(t) \in \mathbb{R}^n$ is the state of node i, $u_i(t) \in \mathbb{R}^{m_i}$ its control input. Cooperative team objectives may be prescribed in terms of the *local neighborhood tracking error* $\delta_i \in \mathbb{R}^n$ (Khoo, Xie, and Man, 2009) as

$$\delta_i = \sum_{j \in N_i} e_{ij}(x_i - x_j) + g_i(x_i - x_0)$$

(6.2)

The pinning gain $g_i \geq 0$ is nonzero for a small number of nodes i that are coupled directly to the leader or control node x_0, and $g_i > 0$ for at least one i [24],[39]. We refer to the nodes i for which $g_i \neq 0$ as the pinned or controlled nodes. Note that δ_i represents the information available to node i for state feedback purposes as dictated by the graph structure, depending as it does only on immediate neighbors in the graph.

The state of the control or target node is $x_0(t) \in \mathbb{R}^n$ which satisfies the dynamics

$$\dot{x}_0 = Ax_0$$

(6.3)

Note that this is in fact a *command generator* [19] and we seek to design a cooperative control command generator tracker. Note that the trajectory generator *A* may not be stable. As such, this command generator captures a wide range of motion trajectories, including position unit step commands, velocity unit ramp commands, oscillatory commands (which have applications in hard disk drive tracking control), and more.

The **Synchronization control design problem** is to design local control protocols $u_i(t) \in \mathbb{R}^{m_i}$ so that all the nodes in *Gr* to synchronize to the state of the control node, i.e. one requires $x_i(t) \to x_0(t), \forall i$. The control protocols must be distributed in the sense that they depend for each agent on information locally available in the graph, that is, on the states of the agent's immediate neighbors.

From (6.2), the overall error vector for network *Gr* is given by

$$\delta = ((L+G) \otimes I_n)(x - \underline{x}_0) = ((L+G) \otimes I_n)\zeta$$

(6.4)

3

where the global vectors are $x = \begin{bmatrix} x_1^T & x_2^T & \cdots & x_N^T \end{bmatrix}^T \in \mathbb{R}^{nN}$, $\delta = \begin{bmatrix} \delta_1^T & \delta_2^T & \cdots & \delta_N^T \end{bmatrix}^T \in \mathbb{R}^{nN}$, and $\underline{x}_0 = \underline{I}x_0 \in \mathbb{R}^{nN}$, with $\underline{I} = \underline{1} \otimes I_n \in \mathbb{R}^{nN \times n}$ and $\underline{1}$ the N-vector of ones. The Kronecker product is $\otimes$ (Brewer, 1978). $G \in \mathbb{R}^{N \times N}$ is a diagonal matrix with diagonal entries equal to the pinning gains g_i. The (global) consensus or synchronization error (e.g. the disagreement vector in (Olfati-Saber, and Murray, 2004)) is

$$\zeta = (x - \underline{x}_0) \in \mathbb{R}^{nN} \tag{6.5}$$

The communication digraph is assumed to be strongly connected. Then, if $g_i \neq 0$ for at least one i, $(L+G)$ is nonsingular with all eigenvalues having positive real parts [17]. The next result therefore follows from (6.4), the Cauchy Schwartz inequality, and the properties of the Kronecker product [7].

Lemma 6.1. *Let the graph have a spanning tree and $g_i > 0$ for at least one root node i. Then $L+G$ is nonsingular and the global consensus synchronization error is bounded by*

$$\|\zeta\| \leq \|\delta\| / \underline{\sigma}(L+G) \tag{6.6}$$

with $\underline{\sigma}(L+G)$ the minimum singular value of $(L+G)$, and $\delta(t) \equiv 0$ if and only if the nodes synchronize, that is

$$x(t) = \underline{I}x_0(t) \tag{6.7}$$

∎

According to this lemma, if the local neighborhood errors (6.2) are small and the graph has a spanning tree with pinning into at least one root node, then the global disagreement errors (6.5) are small and all nodes synchronize cloes to the leader's state. We specify the next asumption.

Assumption 6.1. *The graph has a spanning tree with pinning gain $g_i > 0$ for at least one root node i.*

To find the dynamics of the local neighborhood tracking error, write

$$\dot{\delta}_i = \sum_{j \in N_i} e_{ij}(\dot{x}_i - \dot{x}_j) + g_i(\dot{x}_i - \dot{x}_0) \tag{6.8}$$

which by using equations (6.1) and (6.3) becomes

$$\dot{\delta}_i = A\delta_i + (d_i + g_i)B_i u_i - \sum_{j \in N_i} e_{ij} B_j u_j \tag{6.9}$$

with $\delta_i \in \mathbb{R}^n$, $u_i \in \mathbb{R}^{m_i}$, $\forall i$.

These error dynamics are dynamical systems where the error of agent i is driven by the control agent of agent i and also by the control inputs of its neighbors in the graph. Note that if the control inputs are selected to make all the local neighborhood error dynamics asymptotically stable and the graph has a spanning tree with pinning into at least one root node, then Lemma 1 shows that the states of all agents synchronize to the leader's state.

6.3   Cooperative Multi-player Games on Graphs

In this section we define the core idea of this chapter- Graphical Games. This idea captures notions of differential game theory for multiple dynamical agents connected by a communication graph topology. In graphical games, not only are the dynamics and control policies of the agents important, but also of vital concern is the topology of the communication graph. We wish to achieve synchronization while simultaneously optimizing some local performance specifications on the agents. To capture this, we intend to use the machinery of multi-player games [5].

A key idea in game theory is the relation between the control policy of a single agent and the control policies of other agents in the game. In graphical games, there are two main sets of agents with which the policy of a single agent must be compared: the set of all its neighbors and the set of all other agents in the graph. Therefore, define

$$u_{Gr-i} = \{u_j : j \in N, j \neq i\} \tag{6.10}$$

as the set of policies of all the other nodes in the graph other than node i. Define

$$u_{-i} = \{u_j : j \in N_i\} \tag{6.11}$$

as the set of policies $\{u_j : j \in N_i\}$ of the neighbors of node i.

6.3.1   Cooperative Performance Index

In the error dynamics (6.9), the state of each agent is affected by its own control policies and also by the control policies of its immediate neighbors in the graph. This structure comes from the desired objective of synchronizing the states of all the agents. Therefore, let us define a multi-player game where likewise the performance index for each agent depends on its own control policies and also the control policies of its immediate neighbors in the graph. Define the local performance indices

$$J_i(\delta_i(0), u_i, u_{-i}) = \tfrac{1}{2} \int_0^\infty (\delta_i^T Q_{ii} \delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j) \, dt \quad \equiv \tfrac{1}{2} \int_0^\infty L_i(\delta_i(t), u_i(t), u_{-i}(t)) \, dt \tag{6.12}$$

where all weighting matrices are constant and symmetric with $Q_{ii} > 0, R_{ii} > 0, R_{ij} \geq 0$. Note that the i-th performance index includes only information about the inputs of node i and its neighbors.

It is desired to select control policies for all the agents such that the performance indices of each agent are minimized in the sense that

$$J_i^*(\delta_i(0)) = \min_{u_i} \tfrac{1}{2} \int_0^\infty (\delta_i^T Q_{ii} \delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j) \, dt \tag{6.13}$$

for all agents i. This optimization problem only takes into account directly the control policy u_i of agent i and the value of its own performance index. The policies of other agents in the game come into play only indirectly through their influence on $J_i(\delta_i(0), u_i, u_{-i})$. This has been called a *noncooperative game* in the literature in the sense that each player only takes into accountdirectly its own policy and not those of other agents in the graph.

We call minimizing (6.13) subject to the dynamics (6.9) a *graphical game*. It depends on the topology of the communication graph $Gr = (V, E)$.

Though the optimization objective (6.13) is myopic and does not take into account the neighbors policies, it is seen that the neighbors' policies are weighted in the performance index (6.12). Therefore, in some sense each agent tries to minimize its own performance index without causing its neighbors to use undue control energy.

If optimization problem (6.13) is solved, then boundedness of $J_i(\delta_i(0), u_i, u_{-i})$ for all i implies that the integrand vanishes with t. Then, positive definiteness of Q_{ii} guarantees that the states $\delta_i(t)$ go to zero, so that local error dynamics (6.9) are stable. Hence, if the graph is strongly connected, the states of equations (6.1) synchronize to the leader (6.3) in view of Lemma 6.1.

For dynamics (6.9) with performance objectives (6.12), introduce the associated Hamiltonians

$$H_i(\delta_i, p_i, u_i, u_{-i}) \equiv p_i^T \left(A\delta_i + (d_i + g_i)B_i u_i - \sum_{j \in N_i} e_{ij} B_j u_j \right) + \frac{1}{2}\left(\delta_i^T Q_{ii}\delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j \right) \tag{6.14}$$

where p_i is the costate variable. Necessary conditions [21] for a minimum of (6.12) are (6.9) and the costate equations

$$-\dot{p}_i = \frac{\partial H_i}{\partial \delta_i} \equiv A^T p_i + Q_{ii}\delta_i \tag{6.15}$$

and the stationarity conditions

$$0 = \frac{\partial H_i}{\partial u_i} \Rightarrow u_i = -(d_i + g_i)R_{ii}^{-1}B_i^T p_i \tag{6.16}$$

6.3.2 Global and Local Performance Objectives: Cooperation and Competition

The overall objective in graphical games is to ensure synchronization of all the states $x_i(t)$ to $x_0(t)$. The multi-player game formulation just presented allows for considerable freedom of each agent while achieving this objective. The minimization objective in (6.13) is myopic or 'noncooperative' in that it only takes into account directly the control policy of agent i. This has been called a noncooperative game in the literature. Nevertheless, the agents can have conflicting as well as collaborative team objecties as seen in the following development.

The performance objective of each node can be written as

$$J_i = \frac{1}{N_i}\sum_{j \in N_i} J_j + \frac{1}{N_i}\sum_{j \in N_i}(J_i - J_j) \equiv J_{team} + J_i^{conflict} \tag{6.17}$$

where J_{team} is the overall ('center of gravity') performance objective of the networked team and $J_i^{conflict}$ is the conflict of interest or competitive objective. J_{team} measures how much the players are vested in common goals, and $J_i^{conflict}$ expresses to what extent their objectives differ. The objective functions can be chosen by the individual players, or they may be assigned to yield some desired team behavior.

In the case of N-player zero-sum games, one has $J_{team} = 0$ and there are no common objectives. One case is the 2-player zero-sum game, where $J_2 = -J_1$, and one has $J_{team} = 0$ and $J_i^{conflict} = J_i$.

6.3.3 Graphical Games

Interpreting the control inputs u_i, u_j as state-dependent policies or strategies, the value function for node i corresponding to those policies is

$$V_i(\delta_i(t)) = \frac{1}{2}\int_t^\infty (\delta_i^T Q_{ii}\delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j)\, dt \tag{6.18}$$

6

Given fixed policies of its neighbor nodes, the control objective of agent i in the graphical game is to determine

$$V_i^*(\delta_i(t)) = \min_{u_i} \frac{1}{2} \int_t^\infty (\delta_i^T Q_{ii} \delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j)\, dt \tag{6.19}$$

This is known as the best response policy of agent i to the fixed policies of its neighbors.

Definition 6.1. Admissible Policies. *Control policies u_i, $\forall i$ are defined as admissible if u_i are continuous, $u_i(0) = 0$, u_i stabilize systems (6.9) locally, and the values (6.18) are finite.*

When V_i is finite, using Leibniz' formula, a differential equivalent to (6.18) is given in terms of the Hamiltonian functions (6.14) by the graphical games Bellman equations

$$H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}, u_i, u_{-i}) \equiv \frac{\partial V_i}{\partial \delta_i}^T \left(A\delta_i + (d_i + g_i)B_i u_i - \sum_{j \in N_i} e_{ij} B_j u_j \right) + \frac{1}{2}\left(\delta_i^T Q_{ii}\delta_i + u_i^T R_{ii}u_i + \sum_{j \in N_i} u_j^T R_{ij}u_j \right) = 0 \tag{6.20}$$

with boundary condition $V_i(0) = 0$. (The gradient is disabused here as a column vector.) That is, solution of equation (6.20) serves as an alternative to evaluating the infinite integral (6.18) for finding the value associated to the current feedback policies. It is shown in the proof of Theorem 6.2 that (6.20) is a Lyapunov equation. According to (6.20) and (6.14) one equates the costate $p_i = \partial V_i / \partial \delta_i$.
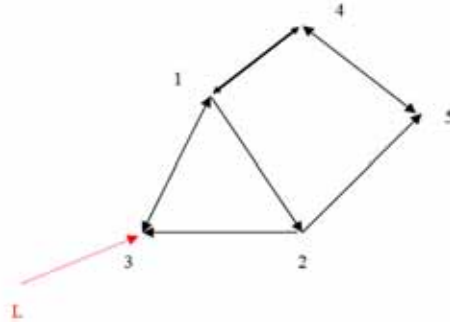
Fig. 6.1 Communication Graph.

Figure 6.1 shows an example communication graph that will be used later for simulation purposes. The incoming edges show the flow of the local information that is exchanged between the agents, e.g. node 2 receives information only from agent 1 (incoming edge). This is the sort of structure captured in the graphical game. Such topologies appear in practical situations such as coordination of heterogeneous teams of unmanned vehicles.

We assume throughout the chapter that the game is well-formed in the following sense.

Definition 6.2. Well-Formed Graphical Game. *The graphical game with local dynamics (6.9) and performance indices (6.12) is well-formed if $B_j \neq 0 \rightleftharpoons e_{ij} \in E$, $R_{ij} \neq 0 \rightleftharpoons e_{ij} \in E$. This means that whenever we have an edge in the communication graph, the input matrix B_j and the user defined matrix R_{ij} in the performance indices should be different than zero.*

Employing the stationarity condition (6.16) one obtains the control policies

$$u_i = u_i(V_i) \equiv -(d_i + g_i)R_{ii}^{-1}B_i^T \frac{\partial V_i}{\partial \delta_i} \equiv -h_i(p_i) \tag{6.21}$$

In graphical games we are interested in Nash equilibrium.

7

Definition 6.3. Nash Equilibrium [5]. *(Global Nash equilibrium) An N-tuple of policies* $\left\{u_1^*, u_2^*, ..., u_N^*\right\}$ *is said to constitute a global Nash equilibrium solution for an N player game if for all $i \in N$ and $\forall u_i, u_{Gr-i}$*

$$J_i^* \triangleq J_i(u_i^*, u_{Gr-i}^*) \le J_i(u_i, u_{Gr-i}^*) \tag{6.22}$$

The N-tuple of game values $\left\{J_1^*, J_2^*, ..., J_N^*\right\}$ is known as a Nash equilibrium outcome of the N-player game.

In Nash equilibrium, each player is in best response to all other players in the game. Then, it does not benefit any player i to change his policy u_i unilaterally when others do not, because any change in policy u_i will increase his cost J_i or at best leave it the same.

The distributed multi-player graphical game with local dynamics (6.9) and local performance indices (6.12) should be contrasted with standard multi-player games [5] which have centralized dynamics

$$\dot{z} = Az + \sum_{i=1}^{N} B_i u_i \tag{6.23}$$

where $z \in \mathbb{R}^n$ is the state, $u_i(t) \in \mathbb{R}^{m_i}$ is the control input for every player, and where the performance index of each player depends on the control inputs of all other players. In the graphical games, by contrast, each node's dynamics and performance index only depend on its own state, its control, and the controls of its immediate neighbors in the graph. See Figure 6.1, where only local information is exchanged. On the contrary in the standard games (Basar, and Olsder 1999) the communication graph is complete (e.g. fully connected). It can be shown tat graphical games are a special case of standard multi-player games where the graph topolocy is explicitly revealed.

Static (e.g. nondynamic) graphical games have been studied in the computational intelligence community [15],[16],[29]. A (nondynamic) graphical game has been defined there as a tuple (Gr, U, v) with $Gr = (V, E)$ a graph with N nodes, action set $U = U_1 \times \cdots \times U_N$ with U_i the set of actions available to node i, and $v = \begin{bmatrix} v_1 & \cdots & v_N \end{bmatrix}^T$ a payoff vector, with $v_i(U_i, \{U_j : j \in N_i\}) \in \mathbb{R}$ the payoff function of node i. It is important to note that *the payoff of node i only depends on its own action and those of its immediate neighbors*. The work on static graphical games has focused on developing algorithms to find standard Nash equilibria for payoffs generally given in terms of matrices at each node. Multiple Nash equilibria may occur. Such algorithms are simplified in that they only have complexity on the order of the maximum node degree in the graph, not on the order of the number of players *N*. Undirected graphs are studied, and it is assumed that the graph is connected.

The intention in this chapter is to provide online real-time adaptive methods for solving differential graphical games that are distributed in nature. That is, the control protocols and adaptive algorithms of each node are allowed to depend only information about itself and its neighbors. Moreover, as the game solution is being learned, all node dynamics are required to be stable, until finally all the nodes synchronize to the state of the control node. These online methods are discussed in Section 6.7.

The following notions are needed in the study of differential graphical games.

Definition 6.4. Best Response [29]. *Agent i's best response to fixed policies u_{-i} of his neighbors is the policy u_i^* such that*

$$J_i(u_i^*, u_{-i}) \le J_i(u_i, u_{-i}) \tag{6.24}$$

for all policies u_i of agent i.

For centralized multi-agent games, where there is one single dynamics given by (6.23) and the performance of each agent depends on the actions of all other agents, an equivalent definition of Nash equilibrium is that each agent is in best response to all other agents. In graphical games, if all agents are in best response to their neighbors, then all agents are in Nash equilibrium, as seen in the proof of Theorem 6.1.

8

However, a counterexample shows the problems with the definition of Nash equilibrium in graphical games. Consider the completely disconnected graph with empty edge set where each node has no neighbors. Then Definition 6.4 holds if each agent simply chooses his single-player optimal control solution $J_i^* = J_i(u_i^*)$, since, for the disconnected graph case one has

$$J_i\,(u_i\,) = J_i\,(u_i\,,u_{Gr-i}) = J_i\,(u_i,u{'}_{Gr-i}),\ \ \forall i \tag{6.25}$$

for any choices of the two sets $u_{Gr-i}, u{'}_{Gr-i}$ of the policies of all the other nodes. That is, the value function of each node does not depend on the policies of any other nodes.

Note, however, that Definition 6.3 also holds, that is, the nodes are in a global Nash equilibrium. Pathological cases such as this counterexample cannot occur in the standard games with centralized dynamics (6.23), particularly because stabilizability conditions are usually assumed.

6.4   Interactive Nash Equilibrium

The counterexample in the previous section shows that in pathological cases when the graph is disconnected, agents can be in Nash equilibrium, yet have no influence on each others' games. In such situations, the definition of coalition-proof Nash equilibrium [28] may also hold, that is, no set of agents has an incentive to break away from the Nash equilibrium and seek a new Nash solution among themselves.

To rule out such undesirable situations and guarantee that all agents in a graph are involved in the same game, we make the following stronger definition of global Nash equilibrium.

Definition 6.5. Interactive Global Nash equilibrium. *An N-tuple of policies* $\left\{u_1^*,u_2^*,...,u_N^*\right\}$ *is said to constitute an interactive global Nash equilibrium solution for an N player game if, for all $i \in N$, the Nash condition (6.22) holds and in addition there exists a policy $u{'}_k$ such that*

$$J_i\,(u_k^*,u_{Gr-k}^*) \neq J_i\,(u{'}_k,u_{Gr-k}^*) \tag{6.26}$$

for all $i,k \in N$. That is, at equilibrium there exists a policy of every player k that influences the performance of all other players i.

If the systems are in Interactive Nash equilibrium, the graphical game is well-defined in the sense that all players are in a single Nash equilibrium, with each player affecting the decisions of all other players. Condition (6.26) means that the reaction curve [5] of any player i is not constant with respect to all variations in the policy of any other player k. A similar interactivity condition was used in [26].

The next results give conditions under which the local best responses in Definition 6.4 imply the interactive global Nash of Definition 6.5.

Consider the systems (6.9) in closed-loop with admissible feedbacks (6.16), (6.21) denoted by $u_k = K_k p_k - v_k$ for a single node k and $u_j = K_j p_j, \forall j \neq k$. Then

$$\dot{\delta}_i = A\delta_i + (d_i + g_i)B_i K_i p_i - \sum_{j \in N_i} e_{ij} B_j K_j p_j + e_{ik} B_k v_k, \quad k \neq i \tag{6.27}$$

The global closed-loop dynamics are

$$\begin{bmatrix}\dot{\delta}\\\dot{p}\end{bmatrix}=\begin{bmatrix}(I_N \otimes A) & ((L+G)\otimes I_n)diag(B_i K_i)\\-diag(Q_{ii}) & -(I_N \otimes A^T)\end{bmatrix}\begin{bmatrix}\delta\\p\end{bmatrix}+\begin{bmatrix}((L+G)\otimes I_n)\underline{B}_k\\0\end{bmatrix}\bar{v}_k \equiv \bar{A}\begin{bmatrix}\delta\\p\end{bmatrix}+\bar{B}\bar{v}_k \tag{6.28}$$

with $\underline{B}_k = diag(B_i)$ and $\overline{v}_k = \begin{bmatrix} 0 & \cdots & v_k^T & \cdots & 0 \end{bmatrix}^T$ has all block entries zero with v_k in block k. Consider node i and let $M > 0$ be the first integer such that $[(L+G)^M]_{ik} \neq 0$, where $[.]_{ik}$ denotes the element (i,k) of a matrix. That is, M is the length of the shortest directed path from k to i. Denote the nodes along this path by $k = k_0, k_1, \cdots, k_{M-1}, k_M = i$. Denote element (i,k) of $L+G$ by ℓ_{ik}. Then the $n \times m$ block element in block row i and block column k of matrix $\overline{A}^{2(M-1)}\overline{B}$ is equal to

$$\left[\overline{A}^{2(M-1)}\overline{B} \right]^{ik} = \sum_{k_{M-1}, \cdots, k_1} \ell_{i, k_{M-1}} \cdots \ell_{k_1, k} B_{k_{M-1}} K_{k_{M-1}} Q_{k_{M-1}} B_{k_{M-2}} \cdots B_{k_1} K_{k_1} Q_{k_1} B_k \equiv \sum_{k_{M-1}} B_{k_{M-1}} \overline{B}_{k_{M-1}, k} \qquad (6.29)$$

where $\overline{B}_{k_{M-1}, k} \in R^{m_{k_{M-1}} \times m_k}$ and $\begin{bmatrix} \ \end{bmatrix}^{ik}$ denotes the position of the block element in the block matrix.

The next assumption 6.2a is required. Assumption 6.2b simplifies the upcoming proof of Theorem 6.1

Assumption 6.2.

  a. $\overline{B}_{k_{M-1}, k} \in R^{m_{k_{M-1}} \times m_k}$ has rank $m_{k_M - 1}$, for all i, k.

  b. All shortest paths to node i from node k pass through a single neighbor $k_M - 1$ of i, for all i, k.

An example case where Assumption 6.2a holds is when there is a single shortest path from k to i, $m_i = m, \forall i$, $rank(B_i) = m, \forall i$.

Lemma 6.2. Let (A, B_j) be reachable for all $j \in N$ and let Assumption 6.2 hold. Then the i-th closed-loop system (6.27) is reachable from input v_k if and only if there exists a directed path from node k to node i.

Proof.

  Sufficiency. If $k = i$ the result is obvious. Otherwise, the reachability matrix from node k to node i has the $n \times m$ block element in block row i and block column k given as

$$\left[\overline{A}^{2(M-1)}\overline{B} \quad \overline{A}^{2(M-1)+1}\overline{B} \quad \overline{A}^{2(M-1)+2}\overline{B} \quad \cdots \right]^{ik}$$

$$= \left[\sum_{k_{M-1}} B_{k_{M-1}} \quad \sum_{k_{M-1}} A B_{k_{M-1}} \quad \sum_{k_{M-1}} A^2 B_{k_{M-1}} \quad \cdots \right] \begin{bmatrix} \overline{B}_{k_{M-1},k} & * & * \\ 0 & \overline{B}_{k_{M-1},k} & * \\ \vdots & 0 & \overline{B}_{k_{M-1},k} \\ 0 & \cdots & 0 & \ddots \end{bmatrix}$$

where * denotes nonzero entries. Under the assumptions, the matrix on the right has full row rank and the matrix on the left is written as

$$\left[B_{k_{M-1}} \quad A B_{k_{M-1}} \quad A^2 B_{k_{M-1}} \quad \cdots \right].$$

However, $(A, B_{k_{M-1}})$ is reachable.

  Necessity. If there is no path from node k to node i, then the control input of node k cannot influence the state or value of node i. ∎

Theorem 6.1. Interactive Nash Equilibrium. *Let* (A, B_i) *be reachable for all* $i \in N$. *Let every node i be in best response to all his neighbors* $j \in N_i$. *Let Assumption 6.2 hold. Then all nodes in the graph are in interactive global Nash equilibrium if and only if the graph is strongly connected.*

Proof. Let every node i be in best response to all his neighbors $j \in N_i$. Then $J_i (u_i^*, u_{-i}) \le J_i (u_i, u_{-i}), \forall i$. Hence $u_j = u_j^*, \forall u_j \in u_{-i}$ and $J_i (u_i^*, u_{-i}^*) \le J_i (u_i, u_{-i}^*), \forall i$. However, according to (6.12) $J_i (u_i^*, u_{-i}^*) = J_i (u_i^*, u_{-i}^*, u_k), \forall k \notin \{i\} \cup N_i$ so that $J_i (u_i^*, u_{Gr-i}^*) \le J_i (u_i, u_{Gr-i}^*), \forall i$ and the nodes are in Nash equilibrium.

Necessity. If the graph is not strongly connected, then there exist nodes k and i such that there is no path from node k to node i. Then, the control input of node k cannot influence the state or the value of node i. Therefore, the Nash equilibrium is not interactive.

Sufficiency. Let (A, B_i) be reachable for all $i \in N$. Then if there is a path from node k to node i, the state δ_i is reachable from u_k, and from (6.12) input u_k can change the value J_i. Strong connectivity means there is a path from every node k to every node i and condition (6.26) holds for all $i, k \in N$. ∎

The reachability condition is sufficient but not necessary for Interactive Nash equilibrium.
According to the results just established, the following assumption is specified.

Assumption 6.3. (A, B_i) *is reachable for all* $i \in N$.

6.5 Stability and Solution of Graphical Games

Substituting control policies (6.21) into (6.20) yields the coupled cooperative game Hamilton-Jacobi (HJ) equations

$$\frac{\partial V_i}{\partial \delta_i}^T A_i^c + \tfrac{1}{2} \delta_i^T Q_{ii} \delta_i + \tfrac{1}{2}(d_i + g_i)^2 \frac{\partial V_i}{\partial \delta_i}^T B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i}$$
$$+ \tfrac{1}{2} \sum_{j \in N_i} (d_j + g_j)^2 \frac{\partial V_j}{\partial \delta_j}^T B_j R_{jj}^{-1} R_{ij} R_{jj}^{-1} B_j^T \frac{\partial V_j}{\partial \delta_j} = 0, i \in N \quad (6.30)$$

where the closed-loop matrices are

$$A_i^c = A\delta_i - (d_i + g_i)^2 B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i} + \sum_{j \in N_i} e_{ij}(d_j + g_j) B_j R_{jj}^{-1} B_j^T \frac{\partial V_j}{\partial \delta_j}, i \in N \quad (6.31)$$

For a given V_i, define $u_i^* = u_i(V_i)$ as (6.21) given in terms of V_i. Then HJ equations (6.30) can be written as

$$H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}, u_i^*, u_{-i}^*) = 0 \quad (6.32)$$

There is one coupled HJ equation corresponding to each node, so solution of this *N*-player game problem is blocked by requiring a solution to *N* coupled partial differential equations. In the next sections we show how to solve this *N*-player cooperative game online in a distributed fashion at each node, requiring only measurements from neighbor nodes, by using techniques from reinforcement learning.

6.5.1   Coupled Riccati Equations

It is now shown that the coupled HJ equations (6.30) can be written as coupled Riccati equations. For the global state δ given in (6.4) we can write the dynamics as

$$\dot{\delta} = (I_N \otimes A)\delta + (L+G) \otimes I_n diag(B_i)u \tag{6.33}$$

where u is the control given by

$$u = -diag(R_{ii}^{-1}B_i^T)\big((D+G) \otimes I_n p\big) \tag{6.34}$$

where $diag(.)$ denotes diagonal matrix of appropriate dimensions. Furthermore the global costate dynamics are

$$-\dot{p} = \frac{\partial H}{\partial \delta} \equiv (I_N \otimes A)^T p + diag(Q_{ii})\delta \tag{6.35}$$

This is the same set of coupled dynamic equations as in the single agent optimal control (Lewis, and Syrmos, 1995). The solution can be determined using coupled Algebraic Riccati equations. To accomplish this the costate is written as

$$p = \overline{P}\delta \tag{6.36}$$

for some matrix $\overline{P} > 0 \in \mathbb{R}^{nNxnN}$.
   The next result shows that (6.30) can be written in the form of coupled Riccati equations.

Lemma 6.3. *HJ equations (6.30) are equivalent to the coupled Riccati equations*

$$\delta^T \overline{P}^T \overline{A}_i\delta - \delta^T \overline{P}^T \overline{B}_i \overline{P}\delta + \tfrac{1}{2}\delta^T \overline{Q}_i\delta + \tfrac{1}{2}\delta^T \overline{P}^T \overline{R}_i \overline{P}\delta = 0 \text{ (6.37)} \tag{6.38}$$

or equivalently, in closed-loop form,

$$(\overline{P}^T \overline{A}_{ic} + \overline{A}_{ic}{}^T \overline{P} + \overline{Q}_i + \overline{P}^T \overline{R}_i \overline{P}) = 0 \tag{6.39}$$

where $\overline{P}$ is defined by (6.36), and

$$\overline{A}_i = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & [A]^{ii} & \\ & & & 0 \end{bmatrix}$$

$$\overline{B}_i = \begin{bmatrix} 0 & & & \\ & \left[(d_i + g_i)I_n\right]^{ii} & \left[-e_{ij}I_n\right]^{ij} \\ & & 0 \end{bmatrix} diag((d_i + g_i)B_i R_{ii}^{-1}B_i^T), \qquad \overline{A}_{ic} = \overline{A}_i - \overline{B}_i\overline{P}$$

$$\bar{Q}_i = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & [Q_{ii}]^{ii} & \\ & & & 0 \end{bmatrix}, \qquad \bar{R}_i = diag((d_i + g_i)B_i R_{ii}^{-1}) \begin{bmatrix} R_{i1} & & & & \\ & \ddots & & & \\ & & R_{ij} & & \\ & & & \ddots & \\ & & & & R_{ii} \\ & & & & & R_{iN} \end{bmatrix} diag((d_i + g_i)R_{ii}^{-1}B_i^T)$$

Proof. Take (6.20) and write it with respect to the global state and costate as

$$H_i \equiv \begin{bmatrix} \dfrac{\partial V_1}{\partial \delta_1} \\ \vdots \\ \vdots \\ \dfrac{\partial V_N}{\partial \delta_N} \end{bmatrix}^T \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & [A]^{ii} & \\ & & & 0 \end{bmatrix} \delta + \begin{bmatrix} \dfrac{\partial V_1}{\partial \delta_1} \\ \vdots \\ \vdots \\ \dfrac{\partial V_N}{\partial \delta_N} \end{bmatrix}^T \begin{bmatrix} 0 & \cdots & & 0 & & 0 \\ \vdots & 0 & & \vdots & & \vdots \\ \vdots & \vdots & [(d_i+g_i)I_n]^{ii} & [-e_{ij}I_n]^{ij} & \\ 0 & \cdots & & 0 & & 0 \end{bmatrix} \begin{bmatrix} B_1 & & & \\ & \ddots & & \\ & & B_i & \\ & & & B_N \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_i \\ u_N \end{bmatrix}$$

$$+ \tfrac{1}{2}\delta^T \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & [Q_{ii}]^{ii} & \\ & & & 0 \end{bmatrix} \delta + \tfrac{1}{2} \begin{bmatrix} u_1 \\ \vdots \\ u_i \\ u_N \end{bmatrix}^T \begin{bmatrix} R_{i1} & & & \\ & R_{ij} & & \\ & & R_{ii} & \\ & & & R_{iN} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_i \\ u_N \end{bmatrix} = 0 \qquad (6.40)$$

By definition of the costate one has

$$p \equiv \begin{bmatrix} \dfrac{\partial V_1}{\partial \delta_1} & \cdots & \cdots & \dfrac{\partial V_N}{\partial \delta_N} \end{bmatrix}^T = \bar{P}\delta \qquad (6.41)$$

From the control policies (6.21), (6.40) becomes (6.37). ∎

6.5.2 Stability and Solution for Cooperative Nash Equilibrium

It is now shown that if solutions can be found for the coupled HJ design equations (6.30), then they provide the solution to the graphical game problem. It is shown how to solve these HJ equations online using reinforcement learning in Section 6.7. It requires Assumption 6.1, that the graph has a spanning tree with pinning into at least one root node.

Theorem 6.2. Stability and Solution for Cooperative Nash Equilibrium. *Let Assumptions 6.1 and 6.3 hold. Let $V_i > 0 \in C^1$, $i \in N$ be smooth solutions to HJ equations (6.30) and control policies u_i^*, $i \in N$ be given by (6.21) in terms of these solutions V_i. Then:*

a. Systems (6.9) are asymptotically stable so that all agents synchronize.

b. $\{u_1^*, u_2^*, ..., u_N^*\}$ are in global Nash equilibrium and the corresponding game values are

$$J_i^*(\delta_i(0)) = V_i \ , i \in N. \qquad (6.42)$$

Proof.

a. If $V_i > 0$ satisfies (6.30) then it also satisfies (6.20). Take the time derivative to obtain

13

$$\dot{V}_i = \frac{\partial V_i}{\partial \delta_i}^T \dot{\delta}_i = \frac{\partial V_i}{\partial \delta_i}^T \left(A\delta_i + (d_i + g_i)B_i u_i - \sum_{j \in N_i} e_{ij} B_j u_j \right) = -\frac{1}{2} \left(\delta_i^T Q_{ii} \delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j \right) \qquad (6.43)$$

which is negative definite since $Q_{ii} > 0$. Therefore V_i is a Lyapunov function for δ_i and systems (6.9) are asymptotically stable.

b. According to part a, $\delta_i(t) \to 0$ for the selected control policies. For any smooth functions $V_i(\delta_i), i \in N$, such that $V_i(0) = 0$, setting $V_i(\delta_i(\infty)) = 0$ one can write (6.12) as

$$J_i(\delta_i(0), u_i, u_{-i}) = \frac{1}{2} \int_0^\infty (\delta_i^T Q_{ii} \delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j)\, dt + V_i(\delta_i(0)) + \int_0^\infty \dot{V}_i dt$$

or

$$J_i(\delta_i(0), u_i, u_{-i}) = \frac{1}{2} \int_0^\infty (\delta_i^T Q_{ii} \delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j)\, dt + V_i(\delta_i(0)) + \int_0^\infty \frac{\partial V_i}{\partial \delta_i}^T (A\delta_i + (d_i + g_i)B_i u_i - \sum_{j \in N_i} e_{ij} B_j u_j)\, dt$$

Now let V_i satisfy (6.30) and u_i^*, u_{-i}^* be the optimal controls given by (6.21). By completing the squares one has

$$J_i(\delta_i(0), u_i, u_{-i}) = V_i(\delta_i(0))$$

$$+ \int_0^\infty (\frac{1}{2} \sum_{j \in N_i} (u_j - u_j^*)^T R_{ij}(u_j - u_j^*) + \frac{1}{2}(u_i - u_i^*)^T R_{ii}(u_i - u_i^*)$$

$$- \frac{\partial V_i}{\partial \delta_i}^T \sum_{j \in N_i} e_{ij} B_j(u_j - u_j^*) + \sum_{j \in N_i} u_j^{*T} R_{ij}(u_j - u_j^*))\, dt$$

At the equilibrium point $u_i = u_i^*$ and $u_j = u_j^*$ so

$$J_i^*(\delta_i(0), u_i^*, u_{-i}^*) = V_i(\delta_i(0))$$

Define

$$J_i(u_i, u_{-i}^*) = V_i(\delta_i(0)) + \frac{1}{2} \int_0^\infty (u_i - u_i^*)^T R_{ii}(u_i - u_i^*)\, dt$$

and $J_i^* = V_i(\delta_i(0))$. Then clearly J_i^* and $J_i(u_i, u_{-i}^*)$ satisfy (6.24). Since this is true for all i, Nash condition (6.22) is satisfied.

∎

The next result shows when the systems are in Interactive Nash equilibrium. This means that the graphical game is well defined in the sense that all players are in a single Nash equilibrium with each player affecting the decisions of all other players. This result requires a stronger type of connectivity than the existence of a spanning tree required in Theorem 6.2.

Corollary 6.1. *Let the hypotheses of Theorem 6.2 hold. Let Assumption 6.2 hold and the graph be strongly connected. Then $\{u_1^*, u_2^*, ..., u_N^*\}$ are in Interactive Nash equilibrium and all agents synchronize.*

Proof. From Theorems 6.1 and 6.2. ∎

According to Theorem 6.2 and Corollary 6.1, solutions to the HJB can be used to construct policies that result in Nash equilibrium and synchronization if the graph has a spanning tree with pinning into at least one root node. One of the beauties of Nash equilibrium is its symmetry or anonymity, in that all nodes play the same role. Unfortunately, from Theorem 6.2 we can deduce that the symmetry of Nash equilibrium is lost in graphical games. Indeed all that is needed for Nash equilibrium in graphical games is existence of a spanning tree, where one node (the root) may have a privileged role. To restore the symmetry and anonynimity of the nodes we call on Interactive Nash equilibrium, which requires that the graph be strongly connected.

6.6   Policy Iteration Algorithms for Cooperative Multi-player Games

Reinforcement learning (RL) techniques have been used to solve the single-player optimal control problem by using adaptive learning techniques to determine the optimal value function. Especially effective are the neurodynamic programming [6] and approximate dynamic programming (ADP) methods [40],[41]. ADP is a method for online learning of optimal control solutions in real-time. RL techniques have also been applied for multi-player games with centralized dynamics (6.23). See for example [8],[37],[33]. Most applications of RL for solving optimal control problems or games online have been to finite-state systems or discrete-time dynamical systems. In this section is given a policy iteration algorithm for solving continuous-time differential games on graphs. The structure of this algorithm is used in the next section to provide online adaptive solutions for graphical games.

6.6.1   Best Response

Theorem 6.2 and Corollary 6.1 reveal that, under Assumptions 6.2 and 6.3, the systems are in Interactive Nash equilibrium if, for all $i \in N$ node i selects his best response policy to his neighbors' policies and the graph is strongly connected. Define the best response HJ equation as the Bellman equation (6.20) with control $u_i = u_i^*$ given by (6.21) and arbitrary policies $u_{-i} = \{u_j : j \in N_i\}$

$$0 = H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}, u_i^*, u_{-i}) \equiv \frac{\partial V_i}{\partial \delta_i}^T A_i^c + \frac{1}{2}\delta_i^T Q_{ii}\delta_i + \frac{1}{2}(d_i + g_i)^2 \frac{\partial V_i}{\partial \delta_i}^T B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i} + \frac{1}{2}\sum_{j \in N_i} u_j^T R_{ij} u_j \tag{6.44}$$

where the closed-loop matrix is

$$A_i^c = A\delta_i - (d_i + g_i)^2 B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i} - \sum_{j \in N_i} e_{ij} B_j u_j \tag{6.45}$$

Theorem 6.3. Solution for Best Response Policy. *Given fixed neighbor policies* $u_{-i} = \{u_j : j \in N_i\}$, *assume there is an admissible policy* u_i. *Let* $V_i > 0 \in C^1$ *be a smooth solution to the best response HJ equation (6.44) and let control policy* u_i^* *be given by (6.21) in terms of this solution* V_i. *Then:*

   a.   *Systems (6.9) are asymptotically stable so that all agents synchronize.*
   b. u_i^* *is the best response to the fixed policies* u_{-i} *of its neighbors.*

Proof.

   a. Let $V_i > 0$ satisfy (6.44). Then the proof follows Theorem 6.2, part a.
   b. According to part a, $\delta_i(t) \to 0$ for the selected control policies. For any smooth functions $V_i(\delta_i), i \in N$, such that $V_i(0) = 0$, setting $V_i(\delta_i(\infty)) = 0$ one can write (6.12) as

15

$$J_i(\delta_i(0), u_i, u_{-i}) = \tfrac{1}{2} \int_0^\infty (\delta_i^T Q_{ii} \delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j)\, dt + V_i(\delta_i(0)) + \int_0^\infty \frac{\partial V_i}{\partial \delta_i}^T (A\delta_i + (d_i + g_i)B_i u_i - \sum_{j \in N_i} e_{ij} B_j u_j)dt$$

Now let V_i satisfy (6.44), u_i^* be the optimal controls given by (6.21), and u_{-i} be arbitrary policies. By completing the squares one has

$$J_i(\delta_i(0), u_i, u_{-i}) = V_i\ (\delta_i(0)) + \int_0^\infty \tfrac{1}{2}(u_i - u_i^*)^T R_{ii}(u_i - u_i^*)dt$$

The agents are in best response to fixed policies u_{-i} when $u_i = u_i^*$ so

$$J_i(\delta_i(0), u_i^*, u_{-i}) = V_i\ (\delta_i(0))$$

Then clearly $J_i(\delta_i(0), u_i, u_{-i})$ and $J_i(\delta_i(0), u_i^*, u_{-i})$ satisfy (6.24).   ∎

6.6.2   Policy Iteration Solution for Graphical Games

The following algorithm for the *N*-player distributed games is motivated by the structure of policy iteration algorithms in reinforcement learning [6],[30], which rely on repeated policy evaluation (e.g. solution of Bellman equations (6.20)) and policy improvement (solution of (6.21)). These two steps are repeated until the policy improvement step no longer changes the present policy. If the algorithm converges for every i, then (6.20) and (6.21) hold simultaneously, so that it converges to the solution to HJ equations (6.30), and hence provides the distributed Nash equilibrium according to Theorem 6.2. One must note that the value functions can be evaluated only in the case of admissible control policies, for only then do the Bellman equations (6.20) have positive definite solutions.

Algorithm 1. Policy Iteration (PI) Solution for *N*-player Distributed Graphical Games.

Step 0: Set k=0. Start with admissible initial policies u_i^0, $\forall i$. Do for k until convergence.

Step 1: (Policy Evaluation) Solve for V_i^k using Bellman equations (6.20)

$$H_i(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_i^k, u_{-i}^k) = 0\,, \forall i = 1, \dots, N \qquad\qquad (6.46)$$

Step 2: (Policy Improvement) Update the N-tuple of control policies using

$$u_i^{k+1} = \arg\min_{u_i} H_i(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_i, u_{-i}^k), \forall i = 1, \dots, N$$

which explicitly is

$$u_i^{k+1} = -(d_i + g_i)R_{ii}^{-1}B_i^T \frac{\partial V_i^k}{\partial \delta_i}\,, \forall i = 1, \dots, N. \qquad\qquad (6.47)$$

Go to step 1.
On convergence- End   ∎

The following two theorems prove convergence of the policy iteration algorithm for distributed games for two different cases. The two cases considered are the following: i) *only* agent *i* updates its policy, and ii) *all* the agents update their policies simultaneously.

Theorem 6.4. Convergence of Policy Iteration algorithm when only i^{th} agent updates its policy and all players u_{-i} in its neighborhood do not change. *Given fixed neighbors policies u_{-i}, assume there exists an admissible policy u_i. Assume that agent i performs Algorithm 1 and the its neighbors do not update their control policies. Then the algorithm converges to the best response u_i to policies u_{-i} of the neighbors and to the solution V_i to the best response HJ equation (6.44).*

Proof. It is clear that

$$H_i^o(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_{-i}) \equiv \min_{u_i^k} H_i(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_i^k, u_{-i}) = H_i(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_i^{k+1}, u_{-i})$$

(6.48)

Let $H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_i^k, u_{-i}) = 0$ from (6.46) then according to (6.48) it is clear that

$$H_i^o(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_{-i}) \leq 0$$

(6.49)

Using the next control policy u_i^{k+1} and the current policies u_{-i} one has the orbital derivative (Leake, Wen Liu, 1967)

$$\dot{V}_i^k = H_i(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_i^{k+1}, u_{-i}) - L_i(\delta_i, u_i^{k+1}, u_{-i})$$

From (6.48) and (6.49) one has

$$\dot{V}_i^k = H_i^0(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_{-i}) - L_i(\delta_i, u_i^{k+1}, u_{-i}) \leq -L_i(\delta_i, u_i^{k+1}, u_{-i})$$

(6.50)

Because only agent *i* update its control it is true that

$$H_i(\delta_i, \frac{\partial V_i^{k+1}}{\partial \delta_i}, u_i^{k+1}, u_{-i}) = 0 .$$

But since $\dot{V}_i^{k+1} = -L_i(\delta_i, u_i^{k+1}, u_{-i})$, from (6.50) one has

$$\dot{V}_i^k = H_i^0(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_{-i}) - L_i(\delta_i, u_i^{k+1}, u_{-i}) \leq -L_i(\delta_i, u_i^{k+1}, u_{-i}) = \dot{V}_i^{k+1}$$

(6.51)

So that $\dot{V}_i^k \leq \dot{V}_i^{k+1}$ and by integration it follows that

$$V_i^{k+1} \leq V_i^k$$

(6.52)

17

which shows that it is a nonincreasing function bounded below by zero. As such V_i^k is convergent as $k \to \infty$. We can write $\lim_{k \to \infty} V_i^k = V_i^\infty$. According to the principle of optimality [21] we have

$$V_i^k \geq \min_{u_i} \int_t^\infty \tfrac{1}{2}(\delta_i^T Q_{ii}\delta_i + u_i^T R_{ii}u_i + \sum_{j \in N_i} u_j^T R_{ij}u_j)\, dt = \int_t^\infty \tfrac{1}{2}(\delta_i^T Q_{ii}\delta_i + u_i^{*T} R_{ii}u_i^* + \sum_{j \in N_i} u_j^T R_{ij}u_j)\, dt$$

where u_i^* is the optimal control. Let $k \to \infty$ then

$$V_i^\infty \geq \int_t^\infty \tfrac{1}{2}(\delta_i^T Q_{ii}\delta_i + u_i^{*T} R_{ii}u_i^* + \sum_{j \in N_i} u_j^T R_{ij}u_j)\, dt \equiv V_i^*$$

Since $V_i^* \leq V_i^\infty$, the algorithm converges, to V_i^*, to the best response HJ equation (6.44). ∎

The next result concerns the case where all nodes update their policies simultaneously at each step of the algorithm. Then, certain weak coupling conditions must be satisfied. Define the relative control weighting as $\rho_{ij} = \bar{\sigma}(R_{jj}^{-1}R_{ij})$, the maximum singular value of the ratio of control weightings $R_{jj}^{-1}R_{ij}$.

Theorem 6.5. Convergence of Policy Iteration algorithm when all agents update their policies. *Assume all nodes i update their policies at each iteration of PI using Algorithm 1. Then for small edge weights e_{ij} and ρ_{ij}, u_i the algorithm converges to the global Nash equilibrium and for all i , and the values converge to the optimal game values $V_i^k \to V_i^*$.*

Proof. It is clear that

$$H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}^{k+1}, u_i^{k+1}, u_{-i}^{k+1}) \equiv H_i^0(\delta_i, \frac{\partial V_i}{\partial \delta_i}^{k+1}, u_{-i}^k) + \tfrac{1}{2}\sum_{j \in N_i}(u_j^{k+1} - u_j^k)^T R_{ij}(u_j^{k+1} - u_j^k)$$

$$+ \sum_{j \in N_i}\left(u_j^k\right)^T R_{ij}(u_j^{k+1} - u_j^k) + \left(\frac{\partial V_i}{\partial \delta_i}^{k+1}\right)^T \sum_{j \in N_i} e_{ij}B_j(u_j^k - u_j^{k+1})$$

and so

$$\dot{V}_i^{k+1} = -L_i(\delta_i, u_i^{k+1}, u_{-i}^{k+1})$$

$$= -L_i(\delta_i, u_i^{k+1}, u_{-i}^k) + \tfrac{1}{2}\sum_{j \in N_i}(u_j^{k+1} - u_j^k)^T R_{ij}(u_j^{k+1} - u_j^k) + \left(\frac{\partial V_i}{\partial \delta_i}^{k+1}\right)^T \sum_{j \in N_i} e_{ij}B_j(u_j^k - u_j^{k+1}) + \sum_{j \in N_i}\left(u_j^k\right)^T R_{ij}(u_j^{k+1} - u_j^k)$$

Therefore,

$$\dot{V}_i^k \leq \dot{V}_i^{k+1} - \tfrac{1}{2}\sum_{j \in N_i}(u_j^{k+1} - u_j^k)^T R_{ij}(u_j^{k+1} - u_j^k) + \left(\frac{\partial V_i}{\partial \delta_i}^{k+1}\right)^T \sum_{j \in N_i} e_{ij}B_j(u_j^{k+1} - u_j^k) - \sum_{j \in N_i}\left(u_j^k\right)^T R_{ij}(u_j^{k+1} - u_j^k)$$

18

A sufficient condition for $\dot{V}_i^k \leq \dot{V}_i^{k+1}$ is

$$\tfrac{1}{2}\underline{\sigma}(R_{ij})\|\Delta u_j\| > e_{ij}\|p_i^{k+1}\| \cdot \|B_j\| + (d_j + g_j)\rho_{ij}\|p_j^{k-1}\| \cdot \|B_j\|$$

where $\Delta u_j = (u_j^{k+1} - u_j^k)$, p_i the costate and $\underline{\sigma}(R_{ij})$ is the minimum singular value of R_{ij}. This holds if $e_{ij} = 0$, $\rho_{ij} = 0$. By continuity, it holds for small values of e_{ij}, ρ_{ij}.

By integration of $\dot{V}_i^k \leq \dot{V}_i^{k+1}$, it follows that

$$V_i^{k+1} \leq V_i^k \tag{6.53}$$

which shows that it is a nonincreasing function bounded below by zero. As such V_i^k is convergent as $k \to \infty$. We can write $\lim_{k \to \infty} V_i^k = V_i^\infty$. According to the principle of optimality [21] we have

$$V_i^k \geq \int_t^\infty \tfrac{1}{2}(\delta_i^T Q_{ii}\delta_i + u_i^{*T} R_{ii}u_i^* + \sum_{j \in N_i} u_j^{*T} R_{ij}u_j^*)\, dt$$

where u_i^*, u_j^* are the optimal controls. Let $k \to \infty$ then

$$V_i^\infty \geq \int_t^\infty \tfrac{1}{2}(\delta_i^T Q_{ii}\delta_i + u_i^{*T} R_{ii}u_i^* + \sum_{j \in N_i} u_j^{*T} R_{ij}u_j^*)\, dt \equiv V_i^*$$

Since $V_i^* \leq V_i^\infty$, the algorithm converges, to V_i^*, to the cooperate Nash equilibrium HJ equation (6.30).

∎

Remark 6.1. Some applications in graph theory require small edge weights ($e_{ij} \leq e_{max}$ where e_{max} is a relatively small positive number). Shortest path graph problems based on scaling [12] and fast matrix multiplication [42] have running times that depend on the edge weights, and therefore yield improved algorithms for sufficiently small edge weights.

This proof indicates that for the PI algorithm to converge, the neighbors' controls should not unduly influence the i-th node dynamics (6.9), and the j-th node should weight its own control u_j in its performance index J_j relatively more than node i weights u_j in J_i. These requirements are consistent with selecting the weighting matrices to obtain proper performance in the simulation examples. An alternative condition to small edge weights for convergence in Theorem 6.5 is that the norm $\|B_j\|$ should be small. This is similar to the case of weakly coupled dynamics in multi-player games in [5].

6.6.3   Policy Iteration Solution for Graphical Games

In this section an online algorithm for solving cooperative Hamilton-Jacobi equations (6.30) based on [34].is presented. This algorithm uses the structure in the PI Algorithm 1 to develop an actor/critic adaptive control architecture for approximate online solution of (6.30). The implementation procedure in terms of adaptive tuning algorithms is based on the technology in [20] for online neural network adaptive control systems. Approximate solutions of (6.46), (6.47) are obtained using value function approximation (VFA). The algorithm uses two approximator structures at each node, which are taken here as neural networks (NN) [20],[2],[6],[32],[40],[41]. One critic NN is used at each node for value function approximation, and one actor NN is used at each node to approximate the control policy (6.47). The critic NN seeks to solve Bellman equation

(6.46). We give tuning laws for the actor NN and the critic NN such that equations (6.46) and (6.47) are solved simultaneously online for each node. Then, the solutions to the coupled HJ equations (6.30) are determined. Though these coupled HJ equations are difficult to solve, and may not even have analytic solutions, we show how to tune the NN so that approximate solutions are learned online. The next assumption is made.

Assumption 6.4. *For each admissible control policy the nonlinear Bellman equations (6.46) have smooth solutions $V_i \geq 0$.*

In fact, only local smooth solutions are needed. To solve the Bellman equations (6.46), approximation is required of both the value functions V_i and their gradients $\partial V_i / \partial \delta_i$. This requires approximation in Sobolev space [2].

6.6.4   Critic Neural Network

This approach is based on the ideas in [20] for neural network adaptive control. According to the Weierstrass higher-order approximation Theorem [2] there are NN weights W_i such that the smooth value functions V_i and their gradients are approximated using a critic NN as

$$V_i(\delta_i) = W_i^T \phi_i(z_i) + \varepsilon_i \tag{6.54}$$

where $z_i(t)$ is an information vector constructed at node i using locally available measurements, e.g. $\delta_i(t), \{\delta_j(t) : j \in N_i\}$. Vectors $\phi_i(z_i) \in \mathbb{R}^h$ are the critic NN activation function vectors, with h the number of neurons in the critic NN hidden layer. According to the Weierstrass Theorem, the NN approximation error ε_i converges to zero uniformly as $h \to \infty$. Assuming current weight estimates $\hat{W}_i$, the outputs of the critic NN are given by

$$\hat{V}_i = \hat{W}_i^T \phi_i(z_i) \tag{6.55}$$

Then, the Bellman equation (6.46) can be approximated at each step k as

$$H_i(\delta_i, \hat{W}_i, u_i, u_{-i}) = \delta_i^T Q_{ii} \delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j + \hat{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} \left(A\delta_i + (d_i + g_i) B_i u_i - \sum_{j \in N_i} e_{ij} B_j u_j \right) \equiv e_{H_i} \tag{6.56}$$

It is desired to select $\hat{W}_i$ to minimize the square residual error

$$E_1 = \tfrac{1}{2} e_{H_i}^T e_{H_i} \tag{6.57}$$

Then $\hat{W}_i \to W_i$ which solves (6.56) in a least-squares sense and e_{H_i} becomes small. Theorem 6.6 below gives a tuning law for the critic weights that achieves this.

6.6.5   Action Neural Network and Online Learning

Define the control policy in the form of an action neural network which computes the control input (6.47) in the structured form [34]

$$\hat{u}_i \equiv \hat{u}_{i+N} = -\tfrac{1}{2}(d_i + g_i) R_{ii}^{-1} B_i^T \frac{\partial \phi_i(z_i)}{\partial \delta_i}^T \hat{W}_{i+N} \tag{6.58}$$

where $\hat{W}_{i+N}$ denotes the current estimated values of the ideal actor NN weights W_i. The notation $\hat{u}_{i+N}$ is used to keep indices straight in the proof (the first N weights are used for the critics and the rest for the actors). Define the critic and actor NN estimation errors as $\tilde{W}_i = W_i - \hat{W}_i$ and $\tilde{W}_{i+N} = W_i - \hat{W}_{i+N}$.

The next results show how to tune the critic NN and actor NN in real time at each node so that equations (6.46) and (6.47) are simultaneously solved, while closed-loop system stability is also guaranteed. Simultaneous solution of (6.46) and (6.47) guarantees that the coupled HJ equations (6.30) are solved for each node i.

Definition 6.6. Uniformly Ultimately Bounded (UUB). *System (6.9) is said to be uniformly ultimately bounded (UUB) if there exists a compact set $S \subset \mathbb{R}^n$ so that for all $\delta_i(0) \in S$ there exists a bound B and a time $T(B, \delta_i(0))$ such that $\|\delta_i(t)\| \leq B$ for all $t \geq t_0 + T$.*

According to Lemma 6.1, UUB guarantees approximate synchronization of all the nodes' states.

Select the tuning law for the ith critic NN as

$$
\begin{aligned}
\dot{\hat{W}}_i &= -a_i \frac{\partial E_1}{\partial \hat{W}_i} \\
&= -a_i \frac{\sigma_{i+N}}{(1+\sigma_{i+N}{}^T \sigma_{i+N})^2}[\sigma_{i+N}{}^T \hat{W}_i + \delta_i^T Q_{ii}\delta_i + \tfrac{1}{4}\hat{W}_{i+N}{}^T \bar{D}_i \hat{W}_{i+N} \\
&\quad + \tfrac{1}{4}\sum_{j \in N_i}(d_j + g_j)^2 \hat{W}_{j+N}{}^T \frac{\partial \phi_j}{\partial \delta_j} B_j R_{jj}^{-T} R_{ij} R_{jj}^{-1} B_j{}^T \frac{\partial \phi_j^T}{\partial \delta_j}\hat{W}_{j+N}]
\end{aligned}
\tag{6.59}
$$

where $\sigma_{i+N} = \frac{\partial \phi_i}{\partial \delta_i}(A\delta_i + (d_i + g_i)B_i\hat{u}_{i+N} - \sum_{j \in N_i} e_{ij}B_j\hat{u}_{j+N})$. Select the tuning law for the ith actor NN as

$$
\begin{aligned}
\dot{\hat{W}}_{i+N} = -a_{i+N}\{(S_i\hat{W}_{i+N} - F_i\bar{\sigma}_{i+N}^T\hat{W}_i) - \frac{1}{4}\bar{D}_i\hat{W}_{i+N}\frac{\bar{\sigma}_{i+N}}{m_{si}}^T \hat{W}_i \\
-\frac{1}{4}\sum_{j \in N_i}(d_j + g_j)^2 \frac{\partial \phi_j}{\partial \delta_j} B_j R_{jj}^{-T} R_{ij} R_{jj}^{-1} B_j{}^T \frac{\partial \phi_j}{\partial \delta_j}^T \hat{W}_j \frac{\bar{\sigma}_{i+N}}{m_{s_i}}^T \hat{W}_{i+N}\}
\end{aligned}
\tag{6.60}
$$

where $\bar{D}_i(x) \equiv \frac{\partial \phi_i}{\partial \delta_i} B_i R_{ii}^{-1} B_i{}^T \frac{\partial \phi_i}{\partial \delta_i}^T$, $m_{s_i} \equiv (\sigma_{i+N}{}^T \sigma_{i+N} + 1)$, $\bar{\sigma}_{i+N} = \sigma_{i+N}/(\sigma_{i+N}{}^T \sigma_{i+N} + 1)$, and $a_i > 0, \ldots a_{i+N} > 0$ and $F_i > 0, S_i > 0, \quad i \in N$ are tuning parameters. The actor tuning law is based on our work on non zero-sum Nash games with centralized dynamics [33] and it is modified accordingly to capture the required distributed nature of control protocols and tuning laws graphical games [34]. It is designed in such a way to ensure both stability of the system and convergence to the Nash equilibrium.

Definition 6.7. Persistence of Excitation (PE). *A time signal $s(t)$ is persistently exciting over the time interval $[t,t+T]$ if there exist constants $\beta_1 > 0, \beta_2 > 0, T > 0$ such that, for all t,*

$$
\beta_1 I \leq S_0 \equiv \int_t^{t+T} s(\tau)s^T(\tau)d\tau \leq \beta_2 I.
\tag{6.61}
$$

The PE assumption is needed in adaptive control if one desires to perform system identification using, for instance, recursive least squares [13]. It is needed here because one effectively desires to tune the critic parameters to identify $V_i(\delta_i)$ in (6.54).

Theorem 6.6. Online Cooperative Graphical Games. *Let Assumptions 6.3 and 6.4 hold. Let the error dynamics be given by (6.9), and consider the cooperative game formulation in (6.13). Let the critic NN at each node be given by (6.55) and the control input be given for each node by actor NN (6.58). Let the tuning law for the ith critic NN be provided by (6.59) and the tuning law for the ith actor NN be provided by (6.60). Assume* $\bar{\sigma}_{i+N} = \sigma_{i+N} / (\sigma_{i+N}^T \sigma_{i+N} + 1)$ *is persistently exciting. Then the closed-loop system states* $\delta_i(t)$*, the critic NN errors* $\tilde{W}_i$*, and the actor NN errors* $\tilde{W}_{i+N}$ *are uniformly ultimately bounded.*

Proof. [34]. ∎

Remark 6.2. Note that in graphical games, the information available to each node is limited by the graph topology so that each node can obtain information only from itself and its direct neighbors. As such, the control protocols and the NN tuning laws must be distributed. Note that, if z_i is appropriately selected (see Remark 6.5), then $\phi_i(z_i)$ only depends on local information and the tuning laws (6.59), (6.60) only use information from each node and its neighbors. Theorem 6.6 shows that these tuning algorithms, when run simultaneously in real time, both guarantee stability and make the system errors $\delta_i(t)$ small and the NN approximation errors bounded. Small errors guarantee approximate synchronization of all the node trajectories.

Remark 6.3. Persistence of excitation (PE) is needed in standard adaptive control for system identification [13]. PE is needed here because one effectively desires to identify the value functions in (6.54). Nonstandard tuning algorithms are required here for the actor NNs to guarantee stability. It is important to notice that the actor NN tuning law (6.60) of every agent needs information of the critic weights of all his neighbors, while the critic NN tuning law (6.59) of every agent needs information of the actor weights of all his neighbors,

Remark 6.4. NN usage suggests starting with random, nonzero control NN weights in (6.58) in order to converge to the coupled HJ equation solutions. However, extensive simulations show that convergence is more sensitive to the persistence of excitation in the control inputs than to the NN weight initialization. If the proper persistence of excitation is not selected, the control weights may not converge to the correct values.

Remark 6.5. The issue of which inputs $z_i(t)$ to use for the critic and actor NNs needs to be addressed. According to the dynamics (6.9), the value functions (6.18), and the control inputs (6.21), the NN inputs at node i should consist of its own error state $\delta_i(t)$, the error states of its neighbors, and the costates of its neighbors. However, in view of (6.36) the costates are functions of the error states. In view of the approximation capabilities of NN and the desire to obtain distributed adaptive laws it is suitable to take as the NN inputs at node i its own error state and the error states of its neighbors. Furthermore since we are considering Linear Quadratic Regulator (LQR) case the values are selected as quadratic in these NN inputs as shown in the simulation.

The next result shows that the tuning laws given in Theorem 6.6 guarantee approximate solution to the coupled HJ equations (6.30) and convergence to the Nash equilibrium.

Theorem 6.7. Convergence to Cooperative Nash Equilibrium and Synchronization. *Suppose the hypotheses of Theorem 6.6 hold. Then:*

a. $H_i(\delta_i, \hat{W}_i, \hat{u}_i, \hat{u}_{-i})$, $\forall i \in N$ *are uniformly ultimately bounded, where* $\hat{u}_i = -\frac{1}{2}(d_i + g_i) R_{ii}^{-1} B_i^T \dfrac{\partial \phi_i}{\partial \delta_i}^T \hat{W}_i$.

   That is, $\hat{W}_i$ *converge to the approximate cooperative coupled HJ-solution.*

b. $\hat{u}_{i+N}$ *converge to the approximate cooperative Nash equilibrium (Definition 6.2) for every* i .

If Asumption 6.1 holds, then synchronization is reached.

Proof. The proof is similar to [33] but is done only with respect to the neighbors (local information) of each agent and not with respect to all the agents.

Consider the weights $\hat{W}_i, \hat{W}_{i+N}$ to be UUB as proved in Theorem 6.6.

a. The approximate coupled HJ equations are $H_i(\delta_i, \hat{W}_i, \hat{u}_i, \hat{u}_{-i})$, $\forall i \in N$.

$$H_i(\delta_i, \hat{W}_i, \hat{u}_i, \hat{u}_{-i}) \equiv H_i(\delta_i, \hat{W}_i \; \hat{W}_{-i}) = \delta_i^T Q_{ii} \delta_i + \hat{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} A \delta_i$$

$$-\tfrac{1}{4}(d_i + g_i)^2 \hat{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} B_i R_{ii}^{-1} B_i^T \frac{\partial \phi_i}{\partial \delta_i}^T \hat{W}_i$$

$$+\tfrac{1}{4} \sum_{j \in N_i} (d_j + g_j)^2 \hat{W}_j^T \frac{\partial \phi_j}{\partial \delta_j} B_j R_{jj}^{-1} R_{ij} R_{jj}^{-1} B_j^T \frac{\partial \phi_j}{\partial \delta_j}^T \hat{W}_j$$

$$+\tfrac{1}{2} \hat{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} \sum_{j \in N_i} e_{ij} B_j R_{jj}^{-1} B_j^T \frac{\partial \phi_j}{\partial \delta_j}^T \hat{W}_j - \varepsilon_{HJ_i}$$

where ε_{HJ_i}, $\forall i$ are the residual errors due to approximation.

After adding zero we have

$$H_i(\delta_i, \hat{W}_i \; \hat{W}_{-i}) = -\tilde{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} A \delta_i - \tfrac{1}{4}(d_i + g_i)^2 \tilde{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} B_i R_{ii}^{-1} B_i^T \frac{\partial \phi_i}{\partial \delta_i}^T \tilde{W}_i + \tfrac{1}{2}(d_i + g_i)^2 W_i^T \frac{\partial \phi_i}{\partial \delta_i} B_i R_{ii}^{-1} B_i^T \frac{\partial \phi_i}{\partial \delta_i}^T W_i$$

$$+\tfrac{1}{2}(d_i + g_i)^2 W_i^T \frac{\partial \phi_i}{\partial \delta_i} B_i R_{ii}^{-1} B_i^T \frac{\partial \phi_i}{\partial \delta_i}^T \hat{W}_i$$

$$-\tfrac{1}{4} \sum_{j \in N_i} (d_j + g_j)^2 \tilde{W}_j^T \frac{\partial \phi_j}{\partial \delta_j} B_j R_{jj}^{-1} R_{ij} R_{jj}^{-1} B_j^T \frac{\partial \phi_j}{\partial \delta_j}^T \tilde{W}_j$$

$$+\tfrac{1}{2} \sum_{j \in N_i} (d_j + g_j)^2 W_j^T \frac{\partial \phi_j}{\partial \delta_j} B_j R_{jj}^{-1} R_{ij} R_{jj}^{-1} B_j^T \frac{\partial \phi_j}{\partial \delta_j}^T W_j$$

$$+\tfrac{1}{2} \sum_{j \in N_i} (d_j + g_j)^2 W_j^T \frac{\partial \phi_j}{\partial \delta_j} B_j R_{jj}^{-1} R_{ij} R_{jj}^{-1} B_j^T \frac{\partial \phi_j}{\partial \delta_j}^T \hat{W}_j$$

$$+\hat{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} \sum_{j \in N_i} e_{ij} B_j R_{jj}^{-1} B_j^T \frac{\partial \phi_j}{\partial \delta_j}^T \hat{W}_j - \varepsilon_{HJ_i}$$

$$-\tfrac{1}{2} \tilde{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} \sum_{j \in N_i} e_{ij} B_j R_{jj}^{-1} B_j^T \frac{\partial \phi_j}{\partial \delta_j}^T \tilde{W}_j$$

$$-\tfrac{1}{2} W_i^T \frac{\partial \phi_i}{\partial \delta_i} \sum_{j \in N_i} e_{ij} B_j R_{jj}^{-1} B_j^T \frac{\partial \phi_j}{\partial \delta_j}^T \hat{W}_j$$

$$-\tfrac{1}{2} \hat{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} \sum_{j \in N_i} e_{ij} B_j R_{jj}^{-1} B_j^T \frac{\partial \phi_j}{\partial \delta_j}^T W_j \tag{6.62}$$

But

$$\hat{W}_i = -\tilde{W}_i + W_i, \quad \forall i. \tag{6.63}$$

After taking norms in (6.63) and letting $\|W_i\| < W_{i\max}$ one has $\|\hat{W}_i\| = \|-\tilde{W}_i + W_i\| \le \|\tilde{W}_i\| + \|W_i\| \le \|\tilde{W}_i\| + W_{i\max}$.

Now (6.62) with $\sup\|\varepsilon_{HJ_i}\| < \bar{\varepsilon}_i$ becomes

$$\left\|H_i(\delta_i,\hat{W}_i\ \hat{W}_{-i})\right\| \le \left\|\tilde{W}_i\right\|\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|\|A\|\|\delta_i\|$$

$$+\frac{1}{4}(d_i+g_i)^2\left\|\tilde{W}_i\right\|^2\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|^2\|B_i\|^2\left\|R_{ii}^{-1}\right\|$$

$$+\frac{1}{2}(d_i+g_i)^2\left\|W_{i\max}\right\|^2\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|^2\|B_i\|^2\left\|R_{ii}^{-1}\right\|$$

$$+\frac{1}{2}(d_i+g_i)^2\left\|W_{i\max}\right\|\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|^2\|B_i\|^2\left\|R_{ii}^{-1}\right\|\left(\left\|\tilde{W}_i\right\|+W_{i\max}\right)$$

$$+\frac{1}{4}\sum_{j\in N_i}(d_j+g_j)^2\left\|\tilde{W}_j\right\|^2\left\|\frac{\partial\phi_j}{\partial\delta_j}\right\|^2\|B_j\|^2\left\|R_{jj}^{-1}R_{ij}R_{jj}^{-1}\right\|$$

$$+\frac{1}{2}\sum_{j\in N_i}(d_j+g_j)^2\left\|W_{j\max}\right\|^2\left\|\frac{\partial\phi_j}{\partial\delta_j}\right\|^2\|B_j\|^2\left\|R_{jj}^{-1}R_{ij}R_{jj}^{-1}\right\|$$

$$+\left(\left\|\tilde{W}_i\right\|+W_{i\max}\right)\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|\sum_{j\in N_i}e_{ij}\|B_j\|^2\left\|R_{jj}^{-1}\right\|\left\|\frac{\partial\varphi_j}{\partial\delta_j}\right\|\left(\left\|\tilde{W}_j\right\|+W_{j\max}\right)$$

$$+\frac{1}{2}\left\|\tilde{W}_i\right\|\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|\sum_{j\in N_i}e_{ij}\|B_j\|^2\left\|R_{jj}^{-1}\right\|\left\|\frac{\partial\phi_j}{\partial\delta_j}\right\|\left\|\tilde{W}_j\right\|$$

$$+\frac{1}{2}\left\|W_{i\max}\right\|\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|\sum_{j\in N_i}e_{ij}\|B_j\|^2\left\|R_{jj}^{-1}\right\|\left\|\frac{\partial\phi_j}{\partial\delta_j}\right\|\left(\left\|\tilde{W}_j\right\|+W_{j\max}\right)$$

$$+\frac{1}{2}\left(\left\|\tilde{W}_i\right\|+W_{i\max}\right)\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|\sum_{j\in N_i}e_{ij}\|B_j\|^2\left\|R_{jj}^{-1}\right\|\left\|\frac{\partial\phi_j}{\partial\delta_j}\right\|W_{j\max}+\bar{\varepsilon}_2 \tag{6.64}$$

All the signals on the right hand side of (6.64) are UUB and convergence to the approximate coupled HJ solution is obtained for every agent.

b. According to Theorem 6.6, $\left\|\hat{W}_{i+N}-W_i\right\|$, $\forall i$ are UUB. Then it is obvious that $\hat{u}_{i+N}$, $\forall i$ give the approximate cooperative Nash equilibrium (Definition 6.2).

If Asumption 6.1 holds, then synchronization is reached according to Theorem 6.2.

∎

6.7 Simulation Results

This section shows the effectiveness of the online approach described in Theorem 6.6 for two different cases. Consider the five-node strongly connected digraph structure shown in Figure 6.1 with a leader node connected to node 3. The edge weights and the pinning gains are taken equal to 1 so that $d_1=2, d_2=1, d_3=2, d_4=2, d_5=2$.

Select the weight matrices in (6.12) as

$$Q_{11}=Q_{22}=Q_{33}=Q_{44}=Q_{55}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},$$

$$R_{11}=4,\ R_{13}=1,\ R_{14}=1, R_{31}=1,\ R_{32}=1,\ R_{33}=9,\ R_{41}=2,\ R_{44}=9,\ R_{45}=1,\ R_{52}=2,\ R_{54}=1,\ R_{55}=9.$$

We selected large values for $R_{ii}, i = 1,\ldots,5$ so that every agent's cost is influenced most by his own control. Also, we followed the result from Theorem 6.5 that $\rho_{ij} = \bar{\sigma}(R_{jj}^{-1}R_{ij})$ must be small.

In the examples below, every node is a second-order system. Then, for every agent $\delta_i = \begin{bmatrix} \delta_{i1} & \delta_{i2} \end{bmatrix}^T$. According to the graph structure, the information vector at each node is

$$z_1 = \begin{bmatrix} \delta_1^T & \delta_3^T & \delta_4^T \end{bmatrix}^T, z_2 = \begin{bmatrix} \delta_1^T & \delta_2^T \end{bmatrix}^T, z_3 = \begin{bmatrix} \delta_1^T & \delta_2^T & \delta_3^T \end{bmatrix}^T, z_4 = \begin{bmatrix} \delta_1^T & \delta_4^T & \delta_5^T \end{bmatrix}^T, z_5 = \begin{bmatrix} \delta_2^T & \delta_4^T & \delta_5^T \end{bmatrix}^T$$

Since the value is quadratic, the critic NNs basis sets were selected as the quadratic vector in the agent's components and its neighbors' components. Thus the NN activation functions are

$$\phi_1(\delta_1,0,\delta_3,\delta_4,0) = \begin{bmatrix} \delta_{11}^2 & \delta_{11}\delta_{12} & \delta_{12}^2 & 0 & 0 & 0 & \delta_{31}^2 & \delta_{31}\delta_{32} & \delta_{32}^2 & \delta_{41}^2 & \delta_{41}\delta_{42} & \delta_{42}^2 & 0 & 0 & 0 \end{bmatrix}^T$$

$$\phi_2(\delta_1,\delta_2,0,0,0) = \begin{bmatrix} \delta_{11}^2 & \delta_{11}\delta_{12} & \delta_{12}^2 & \delta_{21}^2 & \delta_{21}\delta_{22} & \delta_{22}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$\phi_3(\delta_1,\delta_2,\delta_3,0,0) = \begin{bmatrix} \delta_{11}^2 & \delta_{11}\delta_{12} & \delta_{12}^2 & \delta_{21}^2 & \delta_{21}\delta_{22} & \delta_{22}^2 & \delta_{31}^2 & \delta_{31}\delta_{32} & \delta_{32}^2 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$\phi_4(\delta_1,0,0,\delta_4,\delta_5) = \begin{bmatrix} \delta_{11}^2 & \delta_{11}\delta_{12} & \delta_{12}^2 & 0 & 0 & 0 & 0 & 0 & \delta_{41}^2 & \delta_{41}\delta_{42} & \delta_{42}^2 & \delta_{51}^2 & \delta_{51}\delta_{52} & \delta_{52}^2 \end{bmatrix}^T$$

$$\phi_5(0,\delta_2,0,\delta_4,\delta_5) = \begin{bmatrix} 0 & 0 & 0 & \delta_{21}^2 & \delta_{21}\delta_{22} & \delta_{22}^2 & 0 & 0 & 0 & \delta_{41}^2 & \delta_{41}\delta_{42} & \delta_{42}^2 & \delta_{51}^2 & \delta_{51}\delta_{52} & \delta_{52}^2 \end{bmatrix}^T$$

6.7.1 Position and velocity regulated to zero

For the graph structure shown, consider the node dynamics

$$\dot{x}_1 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix} x_1 + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u_1, \quad \dot{x}_2 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix} x_2 + \begin{bmatrix} 2 \\ 3 \end{bmatrix} u_2$$

$$\dot{x}_3 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix} x_3 + \begin{bmatrix} 2 \\ 2 \end{bmatrix} u_3, \quad \dot{x}_4 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix} x_4 + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_4$$

$$\dot{x}_5 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix} x_5 + \begin{bmatrix} 3 \\ 2 \end{bmatrix} u_5$$

and the command generator $\dot{x}_0 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix} x_0$.

The graphical game is implemented as in Theorem 6.6. Persistence of excitation was ensured by adding a small exponentially decreasing probing noise to the control inputs. Figures 6.2 and 6.3 show the evolution of the states of every agent.
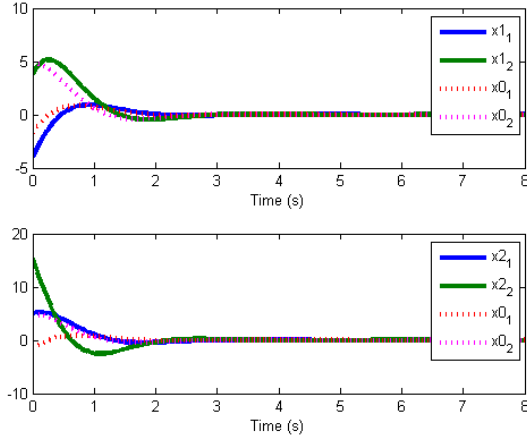
Fig. 6.2 Evolution of the states for agents 1, 2.
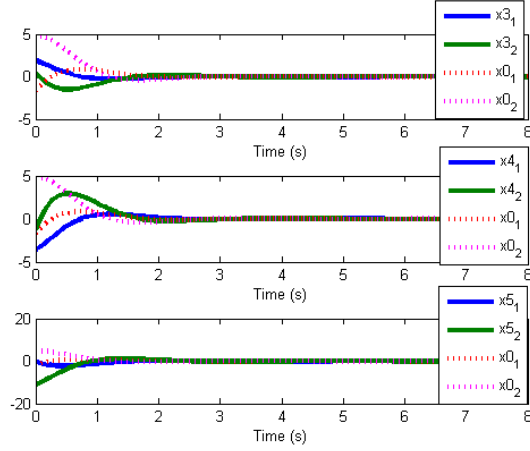
Fig. 6.3 Evolution of the system for agents 3, 4, 5.

6.7.2 All the nodes synchronize to the curve behavior of the leader node

For the graph structure shown above consider the following node dynamics

$$\dot{x}_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x_1 + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u_1, \quad \dot{x}_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x_2 + \begin{bmatrix} 2 \\ 3 \end{bmatrix} u_2$$

$$\dot{x}_3 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x_3 + \begin{bmatrix} 2 \\ 2 \end{bmatrix} u_3, \quad \dot{x}_4 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x_4 + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_4$$

$$\dot{x}_5 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x_5 + \begin{bmatrix} 3 \\ 2 \end{bmatrix} u_5$$

with target generator $\dot{x}_0 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x_0$.

The command generator is marginally stable with poles at $s = \pm j$, so it generates a sinusoidal reference trajectory.

The graphical game is implemented as in Theorem 6.6. Persistence of excitation was ensured by adding a small exponential decreasing probing noise to the control inputs. Figure 6.4 shows the synchronization of all the agents to the leader's behavior as given by the circular Lissajous plot in the phase plane.
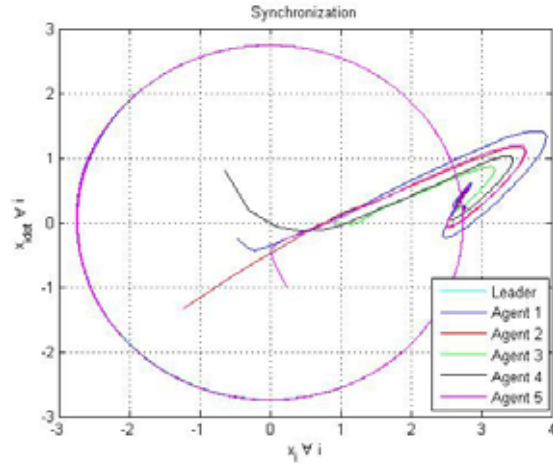
Fig. 6.4 Synchronization of all five agents to the leader.

6.8 Conclusion

This chapter brings together cooperative control, reinforcement learning, and game theory to define and solve multi-player differential games on communication graph topologies. It formulates graphical games for dynamic systems and provides policy iteration and online learning algorithms for computing the Nash equilibrium or best response online in real time using data measured along the ystem trajectories. Simulation results show the effectiveness of the proposed algorithms.

References

[1]   Abou-Kandil H., Freiling G., Ionescu V., & Jank G., (2003). *Matrix Riccati Equations in Control and Systems Theory*, Birkhäuser.

[2]   Abu-Khalaf M., & Lewis F. L., (2005). Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network HJB Approach, *Automatica*, *41*(5), 779-791.

[3]   A. Al-Tamimi, M. Abu-Khalaf, and F.L. Lewis, "Adaptive Critic Designs for Discrete-Time Zero-Sum Games with Application to H-Infinity Control," IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 37, no. 1, pp. 240-247, Feb. 2007.

[4]   A. Al-Tamimi, F.L. Lewis, and M. Abu-Khalaf, "*Discrete-Time Nonlinear HJB Solution Using* Approximate Dynamic Programming: Convergence Proof," Trans. Systems, Man, Cybernetics, Part B, vol. 38, no. 4, pp. 943-949, special issue on ADP/RL, August, 2008.

[5]   Başar T., & Olsder G. J.,(1999). *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA: SIAM.

[6]   Bertsekas D. P., & Tsitsiklis J. N. (1996). *Neuro-Dynamic Programming*, Athena Scientific, MA.

[7]   Brewer J.W., (1978). Kronecker products and matrix calculus in system theory, *IEEE Transactions Circuits and Systems*, *25*, 772-781.

[8]   Busoniu L., Babuska R., & De Schutter B., (2008). A Comprehensive Survey of Multi-Agent Reinforcement Learning, *IEEE Transactions on Systems, Man, and Cybernetics — Part C: Applications and Reviews*, *38(2)*, *156–172*.

[9]   Dierks T. & Jagannathan S., (2010). Optimal Control of Affine Nonlinear Continuous-time Systems Using an Online Hamilton-Jacobi-Isaacs Formulation1, *Proc. IEEE Conf Decision and Control*, Atlanta, *3048-3053*.

[10]   Freiling G., Jank G., & Abou-Kandil H., (2002). On global existence of Solutions to Coupled Matrix Riccati equations in closed loop Nash Games, *IEEE Transactions on Automatic Control*, *41(2)*, *264- 269*.

[11]   Gajic Z., & Li T-Y., (1988). Simulation results for two new algorithms for solving coupled algebraic Riccati equations, *Third Int. Symp. On Differential Games*, Sophia, Antipolis, France.

[12]   Goldberg A. V., (1995) Scaling algorithms for the shortest paths problem, *SIAM J. Comput.*, 24, 494–504.

[13]   Ioannou P., & Fidan B. (2006). *Adaptive Control Tutorial*, SIAM, Advances in Design and Control, PA.

[14]   Johnson M., Hiramatsu T., Fitz-Coy N., & Dixon W. E., (2010). Asymptotic Stackelberg Optimal Control Design for an Uncertain Euler Lagrange System, *IEEE Conference on Decision and Control*, *6686-6691*.

[15]   Kakade S., Kearns M., Langford J., & Ortiz L., (2003). Correlated equilibria in graphical games, *Proc. 4th ACM Conference on Electronic Commerce*, *42–47*.

[16]   Kearns M., Littman M., & Singh S., (2001) Graphical models for game theory, *Proc. 17th Annual Conference on Uncertainty in Artificial Intelligence*, *253–260*.

[17]   Khoo S., Xie L., & Man Z., (2009) Robust Finite-Time Consensus Tracking Algorithm for Multirobot Systems, *IEEE Transactions on Mechatronics*, *14*, *219-228*.

[18]   Leake R. J., Liu Ruey-Wen, (1967). Construction of Suboptimal Control Sequences, *J. SIAM Control*, *5 (1)*, *54-63*.

[19]   Lewis F. (1992). *Applied Optimal Control and Estimation: Digital Design and Implementation*, New Jersey: Prentice-Hall.

[20]   F.L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems,* Taylor and Francis, London, 1999.

[21]   Lewis F. L., Vrabie, D., Syrmos V. L. (2012). *Optimal Control*, 3rd edition, John Wiley.

[22]   F.L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," IEEE Circuits & Systems Magazine, Invited Feature Article, pp. 32-50, Third Quarter 2009.

[23]   F.L. Lewis, D. Vrabie, and K.G. Vamvoudakis, "Reinforcement learning and feedback control," IEEE Control Systems Magazine, pp. 76-105, December, 2012.

[24]   Li X., Wang X., & Chen G., (2004). Pinning a complex dynamical network to its equilibrium, *IEEE Trans. Circuits Syst. I, Reg. Papers, 51(10), 2074–2087*.

[25]   Littman M.L., (2001). Value-function reinforcement learning in Markov games, *Journal of Cognitive Systems Research 1*.

[26]   Marden, J.R., Young, H.P., and Pao, L.Y., "Achieving Pareto Optimality Through Distributed Learning," Proc. IEEE Conf. Decision and Control, pp. 7419-7424, Maui, Dec. 2012.

[27]   Qu Z., (2009). *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*, New York: Springer-Verlag.

[28]   Shinohara R., "Coalition proof equilibria in a voluntary participation game," *International Journal of Game Theory*, vol. 39, no. 4, pp. 603-615, 2010.

[29]   Shoham Y., Leyton-Brown K., (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge University Press.

[30]   Sutton R. S., Barto A. G. (1998) *Reinforcement Learning – An Introduction*, MIT Press, Cambridge, Massachusetts.

[31]   Tijs S (2003), *Introduction to Game Theory*, Hindustan Book Agency, India.

[32]   Vamvoudakis Kyriakos G., & Lewis F. L., (2010). Online Actor-Critic Algorithm to Solve the Continuous-Time Infinite Horizon Optimal Control Problem, *Automatica*,46(5), 878-888.

[33]   Vamvoudakis K.G., & Lewis F. L., (2011). Multi-Player Non-Zero Sum Games: Online Adaptive Learning Solution of Coupled Hamilton-Jacobi Equations, *Automatica*, 47(8), 1556-1569.

[34]   K.G. Vamvoudakis, F.L. Lewis, and G.R. Hudas, "Multi-Agent Differential Graphical Games: online adaptive learning solution for synchronization with optimality," Automatica, vol. 48, no. 8, pp. 1598-1611, Aug. 2012.

[35]   Vrabie, D., Pastravanu, O., Lewis, F. L., & Abu-Khalaf, M. (2009). Adaptive Optimal Control for continuous-time linear systems based on policy iteration, *Automatica*, 45(2), 477-484.

[36]   D. Vrabie and F.L. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially-unknown nonlinear systems," Neural Networks, vol. 22, no. 3, pp. 237-246, Apr. 2009.

[37]   Vrancx P., Verbeeck K., & Nowe A., (2008). Decentralized learning in markov games, *IEEE Transactions on Systems, Man and Cybernetics*, *38(4)*, *976-981*.

[38]   F. Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," IEEE Computational Intelligence Magazine, vol. 4, no. 2, pp. 39–47, May 2009.

[39]   Wang X. & Chen G., (2002). Pinning control of scale-free dynamical networks, Physica A, *310(3-4)*, *521–531*.

[40]   Werbos P. J., (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavior Sciences*, Ph.D. Thesis.

[41]   Werbos P. J. (1992). Approximate dynamic programming for real-time control and neural modeling, *Handbook of Intelligent Control*, ed. D.A. White and D.A. Sofge, New York: Van Nostrand Reinhold.

[42]   Zwick, U., (2002), All pairs shortest paths using bridging sets and rectangular matrix multiplication, *J. ACM*, 49, pp. 289–317.

Part II
Distributed Adaptive Control for Multi-agent Cooperative Systems

The optimal cooperative controllers studied in Part I of the book require complete information of the system dynamics and rely on off-line solutions of matrix design equations. Adaptive control [1], [7], [10] is a powerful method for the design of dynamic controllers that are tuned online in real time to learn stabilizing feedback controllers for systems with unknown dynamics. In cooperative adaptive control, moreover, the dynamics of the agents can be heterogeneous, that is different for each agent. The agent dynamics can also have unknown disturbances.

In Part II of the book we show how to design cooperative adaptive controllers for multi-agent systems on graphs. The challenge is to design adaptive parameter tuning laws that are distributed in the sense that they only require information from the neighbors in the graph topology.

In Part I of the book we studied local and global optimal control for cooperative multi-agent systems on communication on graphs. In these systems, any control protocol must be distributed in the sense that the control for each agent is allowed to depend only on information about that agent and its neighbors in the graph. In Chapter 5 we saw this means that globally optimal controls of distributed form may not exist on a given graph. To obtain globally optimal performance using distributed protocols, the global performance index must be selected to depend on the graph topology in a certain way, specifically, through the graph Laplacian matrix.

Likewise, we shall now see in Part II of the book that adaptive control design for cooperative multi-agent systems requires the use of special Lyapunov functions that depend on the graph topology in a certain way. In adaptive controllers that are admissible for a prescribed communication graph topology, only distributed control protocols and distributed adaptive tuning laws are permitted. It is not straightforward to develop adaptive tuning laws for cooperative agents on graphs that only require information from each agent and its neighbors. The key to this is selecting special Lyapunov functions for adaptive control design that depend in specific ways on the graph topology. Such Lyapunov functions can be constructed using the concept of graph Laplacian potential, which depends on the communication graph topology. The Laplacian potential captures the notion of a virtual potential energy stored in the graph.

7
Graph Laplacian Potential and Lyapunov Functions for Multi-agent Systems

In Part I of the book we studied local and global optimal control for <mark>cooperative multi-agent systems on communication on graphs.</mark> In these systems, <mark>any control protocol must be distributed in the sense that it must respect the communication topology limitations.</mark> That is, the control protocol for each agent is allowed to depend only on information about that agent and its neighbors in the graph. In Chapter 5 we saw this means that globally optimal controls of distributed form may not exist on a given graph. It was shown that for the standard linear quadratic performance index with state and control weighting, globally optimal controls of linear distributed form only exist on a class of directed graphs whose Laplacian matrix is simple, that is, has a diagonal Jordan form. If state-control cross-weighting is allowed, then globally optimal controls of linear distributed form exist on any given digraph. However, in either case, global optimal performance can be attained for all agents using distributed protocols only if the global performance index is selected to depend on the graph topology, specifically, through its Laplacian matrix.

Similarly, we shall see in Part II of the book that adaptive control design for cooperative multi-agent systems requires the use of special Lyapunov functions that depend on the graph topology in a certain way. In the design of admissible adaptive controllers, only distributed control protocols and distributed adaptive tuning laws are permitted. It is not straightforward to develop adaptive tuning laws for cooperative agents on graphs that only require information from that agent and its neighbors. The key to this is selecting special Lyapunov functions for adaptive control design that depend in specific ways on the graph topology. This is related to the selection of performance indices that depend on graph topology for globally optimal performance in Chapter 5.

In this chapter we show that for networked multi-agent systems, there is an energy-like function, called the graph Laplacian potential, that depends on the com-

munication graph topology. The Laplacian potential captures the notion of a virtual potential energy stored in the graph. We shall study the Laplacian potential for both undirected graphs and directed graphs. The Laplacian potential is further used here to construct Lyapunov functions that are suitable for the analysis of cooperative control systems on graphs. These Lyapunov functions depend on the graph topology, and based on them a Lyapunov analysis technique is introduced for cooperative multi-agent systems on graphs. Control protocols coming from such Lyapunov functions are distributed in form, depending only on information about the agent and its neighbors. Such Lyapunov functions that depend on the graph topology are used for the design of distributed adaptive controllers for multi-agent systems on graphs in Chapters 8, 9, 10.

   The results in this chapter were first published in [17].

7.1 Graph Laplacian Potential

Before introducing the graph Laplacian potential, let us first recall the concept of potential energy for some familiar physical systems. Potential energy often means the energy stored in a spring or in a potential field, such as the gravity field or the electric field, when work is done to stretch a spring or against the potential field. Take the spring for example. If a spring, with the spring constant k, is stretched by a length of x, then the potential energy stored in the spring is $\frac{1}{2}kx^2$. Consider further a networked spring-mass system, say three point masses linked by three springs, as shown in Fig. 7.1. Suppose the ideal free lengths of these springs are all zero. Then the potential energy P_E stored in these springs is

$$P_E = \frac{1}{2}k_1 |\overrightarrow{ab}|^2 + \frac{1}{2}k_2 |\overrightarrow{ac}|^2 + \frac{1}{2}k_3 |\overrightarrow{bc}|^2, \qquad (7.1)$$

where k_1, k_2 and k_3 are the spring constants, and $|\overrightarrow{ab}|$, $|\overrightarrow{ac}|$ and $|\overrightarrow{bc}|$ are the lengths between each two masses.
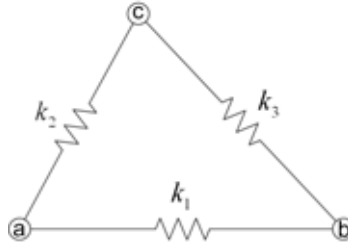

Fig. 7.1. Networked spring-mass system

At this point, one may naturally raise a question that, since multi-agent systems are networked systems, with their connections described by a graph, is there a counterpart of the "potential energy" for the multi-agent systems? The answer is yes. The "potential energy" for a multi-agent system can be treated as the virtual energy stored in a graph, and thus is called the graph Laplacian potential. In this case, the nodes are connected not by springs, but a communication links with edge weights.

The graph Laplacian potential was used in [11] for unweighted undirected graphs as a measure of the total disagreement among all agents. Later it was extended to weighted undirected graphs and balanced graphs in [12], and further to generalized strongly connected digraphs in [17]. As a kind of energy, albeit virtual, zero Laplacian potential implies a steady-state condition of the graph, i.e., which under certain conditions is equivalent to consensus of all agents. In this section, we shall study graph Laplacian potentials for different graphs and their properties.

Consider an interconnected multi-agent system with N agents. Let the communication topology be described by a graph $\mathcal{G}$, with associated adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$, diagonal matrix of in-degrees $D = diag\{d_i\}$, and Laplacian matrix $L = D - \mathcal{A} = [l_{ij}] \in \mathbb{R}^{N \times N}$, where $l_{ii} = d_i = \sum_{j=1}^{N} a_{ij}$ and $l_{ij} = -a_{ij}$ for $j \neq i$. If there is an edge from agent j to agent i, then $a_{ij} > 0$, otherwise $a_{ij} = 0$. Also, we assume $a_{ii} = 0$. Denote the neighbor set of agent i as $N_i = \{j \mid a_{ij} > 0\}$. Let the nodes have dynamical systems, with $x_i \in \mathbb{R}^n$ be the state of agent i. Let $X = col(x_1, ..., x_N) \in \mathbb{R}^{nN}$ be the global state of the multi-agent system obtained by concatenating $x_1, ..., x_N$ column-wise. In the sequel, we shall only consider the scalar case $x_i \in \mathbb{R}$ to simplify the expression, unless otherwise specified. Bear in in mind that all results apply also in the vector case by using Kronecker product.

7.1.1 Laplacian Potential for Undirected Graphs

For unweighted undirected graphs, the graph Laplacian potential is defined as [11]

$$P_L = \sum_{i=1, \, j \in N_i}^{N} (x_i - x_j)^2. \tag{7.2}$$

For weighted undirected graphs, the graph Laplacian potential is defined as [12]

$$P_L = \sum_{i, j=1}^{N} a_{ij}(x_i - x_j)^2, \tag{7.3}$$

where $a_{ij} = a_{ji}$. Note the graph Laplacian potential can also be defined as (7.3), when a_{ij} and a_{ji} are not necessarily the same. It is clear that (7.2) is a special case of (7.3) with unit edge weights. Compare (7.3) and (7.1) to get an "energy-like" concept of the Laplacian potential.

For both unweighted and weighted undirected graphs, the following property holds. It relates the Laplacian potential with the graph Laplacian matrix.

Lemma 7.1. ([11],[12]) *The Laplacian potential for undirected graph satisfies the following identity:*

$$P_L = \sum_{i,j=1}^{N} a_{ij}(x_i - x_j)^2 = 2X^T L X. \tag{7.4}$$

Proof. Expanding $X^T L X$ leads to

$$X^T L X = \sum_{i,j=1}^{N} l_{ij} x_i x_j = \sum_{i=1}^{N} l_{ii} x_i^2 + \sum_{i=1; j \neq i}^{N} l_{ij} x_i x_j$$

$$= \sum_{i=1}^{N} \left(x_i^2 \sum_{j=1}^{N} a_{ij} \right) - \sum_{i,j=1}^{N} a_{ij} x_i x_j$$

$$= \sum_{i,j=1}^{N} a_{ij} x_i (x_i - x_j).$$

Since $a_{ij} = a_{ji}$, $X^T L X$ can also be written as

$$X^T L X = \sum_{i,j=1}^{N} a_{ji} x_j (x_j - x_i)$$

$$= \sum_{i,j=1}^{N} a_{ij} x_j (x_j - x_i).$$

It then follows that

$$2X^T L X = \sum_{i,j=1}^{N} a_{ij} x_i (x_i - x_j) + \sum_{i,j=1}^{N} a_{ij} x_j (x_j - x_i)$$

$$= \sum_{i,j=1}^{N} a_{ij} (x_i - x_j)^2 = P_L.$$

The Laplacian potential is closely related to consensus of the multi-agent system, as will be shown in Lemma 7.4. Some technical lemmas are needed before we proceed.

Lemma 7.2. ([4] Lemma 13.1.1) *If an undirected graph is connected, then* $rank(L) = N - 1$.

Lemma 7.3. ([1] Fact 8.15.2) *Let* $A = A^T \in \mathbb{R}^{n \times n}$. *Then the null space of A is* $null(A) = \{x \in \mathbb{R}^n \mid x^T A x = 0\}$ *if and only if A is positive semidefinite or negative semidefinite, i.e.,* $A \geq 0$ *or* $A \leq 0$.

Lemma 7.4. *For a connected undirected graph,* $P_L = 0$ *if and only if consensus of the multi-agent system is achieved, i.e.,* $x_i = x_j$, $\forall i, j = 1, \ldots, N$.

Proof. If $P_L = \sum_{i,j=1}^{N} a_{ij}(x_i - x_j)^2 = 0$, then $x_i = x_j$ for all $a_{ij} > 0$. For connected graphs, this implies the consensus, i.e., $x_i = x_j$ for all $i, j = 1, \ldots, N$. It is obvious that consensus implies $P_L = 0$. Using the properties of the Laplacian matrix, an alternative way to prove the necessity is shown as follows. For undirected graphs, the Laplacian matrix L is symmetric and positive semidefinite. This can be trivially shown by using *Geršgorin Disc Theorem* ([5] Lemma 3.1). By Lemma 7.3, the null space of L is $null(L) = \{X \in \mathbb{R}^N \mid X^T L X = 0\}$. Let $1_N = [1, \ldots, 1]^T \in \mathbb{R}^N$. The Laplacian matrix has the property that $L1_N = 0$. If the undirected graph is connected, then $rank(L) = N - 1$. Then the dimension of the null space of L is 1. Therefore, the null space of L is $null(L) = \{X \mid X = \alpha 1_N, \forall \alpha \in \mathbb{R}\}$. This completes the proof.

7.1.2 Laplacian Potential for Directed Graphs

For directed graphs, the Laplacian potential can still be defined as (7.3), but the identity (7.4) does not hold due to the non-symmetric property of the Laplacian matrix L. In this subsection, we shall show how the Laplacian potential relates to the Laplacian matrix and propose a generalized Laplacian potential for strongly connected graphs.

Let us begin by defining a different Laplacian matrix for graph $\mathcal{G}$, called the graph column Laplacian matrix L^o.

Definition 7.1. (Column Laplacian matrix) *Let the out-degree of node i be* $d_i^o = \sum_{j=1}^{N} a_{ji}$ *and the out-degree matrix be* $D^o = diag\{d_i^o\} \in \mathbb{R}^{N \times N}$. *Then the column Laplacian matrix of a digraph* $\mathcal{G}$ *is* $L^o = D^o - \mathcal{A}^T$.

Note that the column Laplacian matrix of graph $\mathcal{G}$ is just the Laplacian matrix of its reverse graph $\mathcal{G}'$, that is the graph with all edges reversed. If graph $\mathcal{G}$ is balanced, i.e., $\sum_{j=1}^{N} a_{ij} = \sum_{j=1}^{N} a_{ji}$, then $L^o = L^T$.

Lemma 7.5. *For general digraphs, we have the identity*

$$P_L = \sum_{i,j=1}^{N} a_{ij}(x_j - x_i)^2 = X^T(L + L^o)X. \tag{7.5}$$

For balanced digraphs, we have the identity

$$P_L = \sum_{i,j=1}^{N} a_{ij}(x_j - x_i)^2 = X^T(L + L^T)X. \tag{7.6}$$

Proof. For general digraphs, straightforward computation gives

$$X^T L X = \sum_{i,j=1}^{N} a_{ij} x_i (x_i - x_j)$$

$$X^T L^o X = \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ji} x_i^2 - \sum_{i,j=1}^{N} a_{ij} x_i x_j$$

$$= \sum_{i,j=1}^{N} a_{ij} x_j^2 - \sum_{i,j=1}^{N} a_{ij} x_i x_j$$

$$= \sum_{i,j=1}^{N} a_{ij} x_j (x_j - x_i).$$

Thus,

$$X^T(L + L^o)X = \sum_{i,j=1}^{N} a_{ij}(x_i - x_j)^2.$$

For balanced digraphs, we have $L^o = L^T$, therefore (7.6) holds.

Lemma 7.5 implies that $L + L^o \geq 0$ for general digraphs, and $L + L^T \geq 0$ for balanced digraphs. The case $L + L^T \geq 0$ was discussed in [12] under the term "mirror graph".

Lemma 7.6. *For weakly connected digraphs, $P_L = 0$ if and only if consensus of the multi-agent systems is achieved, i.e., $x_i = x_j, \forall i, j = 1, \ldots, N$.*

Proof. The sufficiency is obvious. Now we show the necessity. We know that $P_L = \sum_{i,j=1}^{N} a_{ij}(x_j - x_i)^2 = 0$ implies $x_i = x_j$ for all $a_{ij} > 0$. Then setting $a_{ji} = a_{ij}$ for all $a_{ij} > 0$ will not change the consensus result. This amounts to replacing all the directed edges with undirected edges. By definition of the weakly connected digraph, neglecting the direction of all edges yields a connected undirected graph. Thus consensus follows from Lemma 7.4.

In Section 7.2, we will see that Laplacian potential identities (7.4) and (7.6) can be used in Lyapunov analysis for consensus, where the time derivative of the proposed Lyapunov function is just the negative of the Laplacian potential. But the identity (7.5) does not have such a property. Motivated by this point, a generalized graph Laplacian potential is defined for strongly connected graphs as follows.

Definition 7.2. (Generalized Laplacian potential) *Suppose a digraph $\mathcal{G}$ is strongly connected. Let $p = [p_1, p_2, \ldots, p_N]^T$ be the left eigenvector of its Laplacian matrix L associated with eigenvalue $\lambda_1 = 0$. The generalized graph Laplacian potential is defined as*

$$P_L = \sum_{i,j=1}^{N} p_i a_{ij}(x_j - x_i)^2. \tag{7.7}$$

The next result is of major importance in Lyapunov analysis and design for cooperative multi-agent systems.

Lemma 7.7. *Suppose the digraph is strongly connected. Let $P = diag\{p_i\} \in \mathbb{R}^{N \times N}$ and*

$$Q = PL + L^T P, \tag{7.8}$$

where p_i is defined as in Definition 7.2. Then the following identity holds

$$P_L = \sum_{i,j=1}^{N} p_i a_{ij}(x_j - x_i)^2 = X^T QX. \tag{7.9}$$

Moreover, $P > 0$ and $Q \geq 0$.

Proof. It is straightforward that

$$X^T PLX = \sum_{i=1}^{N} p_i x_i \sum_{j=1}^{N} a_{ij}(x_i - x_j) = \sum_{i,j=1}^{N} p_i a_{ij} x_i (x_i - x_j)$$

Since $p^T L = 0$ implies $p_i \sum_{j=1}^{N} a_{ij} = \sum_{j=1}^{N} p_j a_{ji}$, we have

$$\sum_{i=1}^{N} p_i x_i \sum_{j=1}^{N} a_{ij}(x_i - x_j)$$
$$= \sum_{i=1}^{N} x_i^2 p_i \sum_{j=1}^{N} a_{ij} - \sum_{i,j=1}^{N} p_i a_{ij} x_i x_j$$
$$= \sum_{i=1}^{N} x_i^2 \sum_{j=1}^{N} p_j a_{ji} - \sum_{i,j=1}^{N} p_i a_{ij} x_i x_j$$
$$= \sum_{j=1}^{N} x_j^2 \sum_{i=1}^{N} p_i a_{ij} - \sum_{i,j=1}^{N} p_i a_{ij} x_i x_j$$
$$= \sum_{i,j=1}^{N} p_i a_{ij} x_j^2 - \sum_{i,j=1}^{N} p_i a_{ij} x_i x_j$$
$$= \sum_{i,j=1}^{N} p_i a_{ij} x_j (x_j - x_i).$$

Then

$$X^T Q X$$
$$= x^T (PL + L^T P) x$$
$$= 2 x^T P L x$$
$$= 2 \sum_{i=1}^{N} p_i x_i \sum_{j=1}^{N} a_{ij}(x_i - x_j) \qquad (7.10)$$
$$= \sum_{i,j=1}^{N} p_i a_{ij} x_i (x_i - x_j) + \sum_{i,j=1}^{N} p_i a_{ij} x_j (x_j - x_i)$$
$$= \sum_{i,j=1}^{N} p_i a_{ij} (x_j - x_i)^2.$$

By Theorem 4.31 in [13], $p_i > 0$, $\forall i = 1, \ldots, N$, implying $P > 0$. It then follows from (7.10) that $X^T Q X \geq 0$, $\forall X \in \mathbb{R}^N$, thus $Q \geq 0$.

Lemma 7.7 relates the generalized graph Laplacian potential with the Laplacian matrix. In fact, we shall see that it provides a way to construct a Lyapunov function for digraphs. A similar result for constructing the Lyapunov equation (7.8) can also be found in [13], as follows.

Lemma 7.8. [13] *Let the Laplacian matrix L be an irreducible singular M-matrix. Let $x > 0$ and $y > 0$ be the right and left eigenvectors of L associated with eigenvalue $\lambda = 0$, i.e., $Lx = 0$ and $L^T y = 0$. Define*

$$P = diag\{p_i\} = diag\{y_i / x_i\},$$
$$Q = PL + L^T P,$$

Then $P > 0$ and $Q \geq 0$.

Remark 7.1. When the graph is strongly connected, its Laplacian matrix *L* is an irreducible singular M-matrix and $rank(L) = N - 1$ ([13] Th.4.31). Then the dimension of the null space of *L* is 1 ([1] Corollary 2.5.5). Since $L1_N = 0$, then $null(L) = span\{1_N\}$. Thus in Lemma 7.8, $x = \alpha 1_N$, $\forall \alpha > 0$ and $\alpha \in \mathbb{R}$. It is clear then that the methods for constructing the Lyapunov equations in Lemma 7.7 and Lemma 7.8 are essentially the same.

Lemma 7.9. *Let the digraph be strongly connected and its Laplacian potential P_L be defined by (7.7). Then $P_L = 0$ if and only if consensus of the multi-agent systems is achieved, i.e., $x_i = x_j$, $\forall i, j = 1, \ldots, N$.*

Proof. Sufficiency is trivial by noticing the identity (7.9). Necessity is shown as follows.

Since $Q \geq 0$, by Lemma 7.2, $P_L = X^T Q X = 0$ implies that $X \in null(Q)$. Next we shall show $null(Q) = span\{1_N\}$. For any $X \in null(L)$, it is clear that $X^T Q X = X^T (PL + L^T P) X = 2X^T PLX = 0$. Lemma 7.2 implies $null(L) \subseteq null(Q)$. When the graph is strongly connected, $null(L) = span\{1_N\}$, then we have $span\{1_N\} \subseteq null(Q)$. We are left to show the dimension of the null space of *Q* is 1. The fact $Q1_N = 0$ and the definition (7.8) implies that *Q* can be treated as a valid Laplacian matrix of an augmented graph $\overline{\mathcal{G}}$, which has the same node set as graph $\mathcal{G}$ and the weight of edge (v_j, v_i) being $\overline{a}_{ij} = p_i a_{ij} + p_j a_{ji}$. Obviously, graph $\overline{\mathcal{G}}$ is undirected. Since $p_i \geq 0$, it is clear that if $a_{ij} > 0$, then $\overline{a}_{ij} = \overline{a}_{ji} > 0$. Then strong connectedness of digraph $\mathcal{G}$ implies connectedness of the undirected graph $\overline{\mathcal{G}}$. Thus, $rank(Q) = N - 1$ and the dimension of the null space of *Q* is 1. Therefore, $null(Q) = span\{1_N\}$. Then $X \in null(Q)$, i.e., $X = \alpha 1_N$, $\alpha \in \mathbb{R}$ is equivalent to the consensus of the multi-agent system.

The next section shows how the graph Laplacian potential plays an important role in the Lyapunov analysis of consensus.

7.2 Lyapunov Analysis for Cooperative Regulator Problems

In this section, Lyapunov analysis is demonstrated for two classes of consensus problems. Note that we are not attempting to propose new control protocols, but to provide Lyapunov analysis techniques for existing protocols to show how Lyapunov functions are related to graph Laplacian potentials. Such Lyapunov functions are used in the next chapters for the design of novel adaptive control laws for cooperative control that are distributed on communication graphs.

7.2.1 Consensus of Single Integrator Cooperative Systems

Consider a group of N agents with scalar single-integrator dynamics

$$\dot{x}_i = u_i, \quad i = 1, \dots, N, \tag{7.11}$$

where $x_i \in \mathbb{R}$ is the state and $u_i \in \mathbb{R}$ is the control input. Consider the standard linear consensus protocol (e.g. [8],[11],[12])

$$u_i = -\sum_{j=1}^{N} a_{ij}(x_i - x_j). \tag{7.12}$$

The closed loop system can be written collectively as $\dot{X} = -LX$, where $X = [x_1, \dots, x_N]^T$.

It is well known that consensus can be reached using the linear control protocol (7.12) for undirected graphs, balanced digraphs, strongly connected digraphs, and digraphs containing a spanning tree. These results exist in literature [8],[12], where the analysis is based on eigenvalue properties. Here we shall provide an alternative analysis using a Lyapunov method and show how graph Laplacian potential plays a key role in the Lyapunov analysis. We present a technique that extends consensus/synchronization analysis methods for undirected graphs or balanced digraphs to strongly connected digraphs.

Lemma 7.10. *If the undirected graph $\mathcal{G}$ is connected, consensus of (7.11) is reached using the control law (7.12).*

Proof. Consider the Lyapunov function candidate $V = \sum_{i=1}^{N} x_i^2 = X^T X$. Then

$$\dot{V} = 2X^T \dot{X} = -2X^T LX.$$

Lemma 7.1 implies $\dot{V} = -P_L \leq 0$. By LaSalle's invariance principle [9], the trajectories converge to the largest invariant set $S = \{X \in \mathbb{R} \,|\, \dot{V} = 0\}$. Lemma 7.4 leads to $S = \{X \,|\, X = \alpha 1_N, \forall \alpha \in \mathbb{R}\}$, i.e., consensus is reached.

For directed graph, generally $L \geq 0$ does not hold, thus the development in Lemma 7.10 fails. However, when the directed graph is balanced, we have $L + L^T \geq 0$, which leads to a method of Lyapunov analysis for consensus.

Lemma 7.11. *If the digraph $\mathcal{G}$ is balanced and weakly connected, consensus of (7.11) is reached using the linear control law (7.12).*

Proof. Consider the Lyapunov function candidate $V = \sum_{i=1}^{N} x_i^2 = X^T X$. Then

$$\dot{V} = 2X^T \dot{X} = -2X^T L X = -X^T (L + L^T) X$$

By Lemma 7.5, $\dot{V} = -P_L \leq 0$. By LaSalle's invariance principle [9], the trajectories converge to the largest invariant set $S = \{X \in \mathbb{R}^N \,|\, \dot{V} = 0\}$, i.e. $S = \{X \in \mathbb{R}^N \,|\, P_L = 0\}$. Since digraph $\mathcal{G}$ is weakly connected, Lemma 7.6 yields $S = \{X \in \mathbb{R}^N \,|\, X = \alpha 1_N, \forall \alpha \in \mathbb{R}\}$, i.e., consensus is reached.

Note that when a digraph is balanced and weakly connected, it is strongly connected ([4] Lemma 2.6.1), but not vice versa. For a general strongly connected graph, when it is not balanced, if we still define its Laplacian potential as (7.3), then $\dot{V} \neq -P_L$. Similar Lyapunov analysis methods to those just used do not work. To overcome this drawback, in the next proof we define a generalized Laplacian potential for strongly connected graphs based on Lemma 7.7. The next Lemma appears in [12], where an eigenvalue approach for consensus analysis is carried out. In the proof given here, we introduce a Lyapunov function suitable for consensus analysis on general strongly connected digraphs.

Lemma 7.12. ([12] Corollary 1) *For strongly connected digraphs, consensus of (7.11) is reached using the linear control law (7.12).*

Proof. Consider the Lyapunov function candidate

$$V = \sum_{i=1}^{N} p_i x_i^2 = X^T P X. \tag{7.13}$$

where $P = diag\{p_i\}$ is defined in Lemma 7.7. Then

$$\dot{V} = 2X^T P \dot{X} = -2X^T PLX = -X^T(PL + L^T P)X = -X^T QX.$$

By Lemma 7.7, $Q \geq 0$, hence $\dot{V} = -P_L \leq 0$. By LaSalle invariance principle, the trajectories converge to the largest invariant set $S = \{X \in \mathbb{R}^N \mid \dot{V} = 0\}$. By Lemma 7.9, it is straightforward that $S = \{X \in \mathbb{R}^N \mid X = \alpha 1_N, \forall \alpha \in \mathbb{R}\}$, i.e., the consensus of (7.11) is achieved.

Remark 7.2. Note that in Lemma 7.10, Lemma 7.11, and Lemma 7.12, the derivative of the Lyapunov function is exactly the associated negative graph Laplacian potential $-P_L$. We have introduced in the proof of Lemma 7.12 a technique for Lyapunov analysis of consensus on directed graphs. The method relies on using a Lyapunov function whose quadratic terms are weighted by the elements p_i of the first left eigenvector of the graph Laplacian matrix *L*. This is equivalent to using the Lyapunov equation (7.8) where *P* is a diagonal matrix of the p_i. This highlights the importance of the first left eigenvector elements in studying decreasing flows on graphs. Through an example in Section 7.2.2, we show the importance of the elements p_i in preserving passivity properties on digraphs.

7.2.2 Synchronization of Passive Nonlinear Systems

This section explores the technique mentioned in Remark 7.2 for analysis of synchronization of passive nonlinear systems [3]. It is shown that analysis of passive systems on digraphs can be accomplished by using a Lyapunov function based on storage functions that are weighted by the elements p_i of the first left eigenvector of the graph Laplacian matrix *L*, as motivated in Lemma 7.7. This highlights the importance of the elements p_i in preserving the passivity properties of systems on digraphs. Similar techniques are used in [6].

Passive systems are important because many mechanical systems built from masses, springs and dampers have the passivity property [3],[16] and can be modeled by passive nonlinear systems, such as the robot manipulator. Note also that the single-integrator dynamics is a special type of passive systems.

Chopra and Spong [3] studied the output synchronization of multi-agent systems, with each agent modeled by an input affine nonlinear system that is input-output passive. One of their results assumed that the digraph is strongly connected and balanced. Using the technique indicated in Remark 7.2, here we relax their condition to general strongly connected digraphs.

The problem formulation and synchronization result of [3] are briefly presented as follows. Consider *N* agents and let the communication graph be unweighted, strongly connected, and balanced. Each agent *i* is modeled by a passive nonlinear system

$$\dot{x}_i = f_i(x_i) + g_i(x_i)u_i$$
$$y_i = h_i(x_i),$$
(7.14)

where $x_i \in \mathbb{R}^n$ is the state, $u_i \in \mathbb{R}^m$ is the control input, $y_i \in \mathbb{R}^m$ is the output. The nonlinear functions $f_i(\cdot)$, $g_i(\cdot)$ and $h_i(\cdot)$ are sufficiently smooth with $f_i(0) = 0$ and $h_i(0) = 0$.

The control protocol for each agent is the static output feedback

$$u_i = \sum_{j \in N_i} (y_j - y_i)$$
(7.15)

The group of agents are said to output synchronize if $\lim_{t \to \infty} |y_i(t) - y_j(t)| = 0, \forall i, j$. The following Lemma states an important property of passive systems.

Lemma 7.13. [3] *System (7.14) is passive if and only if there exists a continuously differentiable storage function $V_i : \mathbb{R}^n \to \mathbb{R}$ with $V_i(x_i) \geq 0$ and $V_i(0) = 0$, and a function $S_i(x_i) \geq 0$ such that*

$$\left(\frac{\partial V_i}{\partial x_i}\right)^T f_i(x_i) = -S_i(x_i) \text{ and } \left(\frac{\partial V_i}{\partial x_i}\right)^T g_i(x_i) = h_i^T(x_i)$$

One of the results in [3] is given in the following Theorem. An outline of the proof is presented here to better illustrate our aforementioned technique in Remark 7.2. For details of the proof of Theorem 7.1, readers are referred to [3].

Theorem 7.1. ([3] Theorem 2) *Consider the system (7.14) with control protocol (7.15). Suppose the communication digraph $\mathcal{G}$ is strongly connected and balanced. Then the group of agents output synchronize.*

Proof. Consider the Lyapunov function candidate for the group of N agents defined as

$$V = 2(V_1 + V_2 + \ldots + V_N)$$
(7.16)

where V_i is the storage function for agent i. Then using the passivity property in Lemma 7.13 and a balanced digraph property, it is shown in [3] that

$$\dot{V} = -2\sum_{i=1}^{N} S_i(x_i) - \sum_{i=1}^{N} \sum_{j \in N_i} (y_i - y_j)^T (y_i - y_j) \leq 0.$$

LaSalle's Invariance Principle and strong connectivity of the digraph then imply the output synchronization of system (7.14).

In the next theorem, we extend Theorem 7.1 to weighted strongly connected digraphs. Here we consider the control protocol based on output feedback

$$u_i = \sum_{j \in N_i} a_{ij}(y_j - y_i) = \sum_{i=1}^{N} a_{ij}(y_j - y_i), \ \forall i = 1, 2, ..., N. \tag{7.17}$$

This is a general case of (7.15) where the edge weights a_{ij} take the values of *0* or *1*.

Theorem 7.2. *Consider the system (7.14) with control law (7.17). Suppose the communication graph $\mathcal{G}$ is a strongly connected digraph. Then the group of agents output synchronize, i.e., $y_i = y_j, \forall i, j = 1, ..., N$.*

Proof. Since the digraph $\mathcal{G}$ is strongly connected, we can define $p = [p_1, ..., p_N]^T$ as in Definition 7.2, such that $p^T L = 0$. Consider the Lyapunov function candidate

$$V = 2(p_1 V_1 + p_2 V_2 + ... + p_N V_N), \tag{7.18}$$

where V_i are same storage functions as in Theorem 7.1. Then

$$\begin{aligned}
\dot{V} &= 2\sum_{i=1}^{N} p_i \dot{V}_i \\
&= -2\sum_{i=1}^{N} p_i S_i(x_i) + 2\sum_{i=1}^{N} p_i y_i^T u_i \\
&= -2\sum_{i=1}^{N} p_i S_i(x_i) + 2\sum_{i=1}^{N}\sum_{j=1}^{N} p_i a_{ij} y_i^T (y_j - y_i)
\end{aligned} \tag{7.19}$$

Since $p^T L = 0$ implies that $p_i \sum_{j=1}^{N} a_{ij} = \sum_{j=1}^{N} a_{ji} p_j$, it follows that

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{j=1}^{N} p_i a_{ij} y_i^T (y_j - y_i) \\
&= \sum_{i=1}^{N} p_i y_i^T \sum_{j=1}^{N} a_{ij}(y_j - y_i) \\
&= \sum_{i,j=1}^{N} a_{ij} p_i y_i^T y_j - \sum_{i=1}^{N} p_i y_i^T y_i \sum_{j=1}^{N} a_{ij}
\end{aligned}$$

$$= \sum_{i,j=1}^{N} a_{ij} p_i y_i^T y_j - \sum_{i=1}^{N} y_i^T y_i \sum_{j=1}^{N} a_{ji} p_j$$

$$= \sum_{i,j=1}^{N} a_{ij} p_i y_i^T y_j - \sum_{i,j=1}^{N} a_{ij} p_i y_j^T y_j$$

$$= \sum_{i,j=1}^{N} a_{ij} p_i y_j^T (y_i - y_j)$$

Equation (7.19) can be written as

$$\dot{V} = -2\sum_{i=1}^{N} p_i S_i(x_i) + \sum_{i,j=1}^{N} p_i a_{ij} y_i^T (y_j - y_i) + \sum_{i,j=1}^{N} a_{ij} p_i y_j^T (y_i - y_j)$$

$$= -2\sum_{i=1}^{N} p_i S_i(x_i) + \sum_{i,j=1}^{N} p_i a_{ij} (y_i - y_j)^T (y_j - y_i)$$

$$= -2\sum_{i=1}^{N} p_i S_i(x_i) - \sum_{i,j=1}^{N} p_i a_{ij} (y_i - y_j)^T (y_i - y_j) \leq 0$$

By LaSalle's Invariance Principle, the trajectories of system (7.14) converge to the largest invariant set $S = \{x_i \mid p_i S_i(x_i) = 0, p_i a_{ij}(y_i - y_j)^T(y_i - y_j) = 0, \forall i, j = 1, ..., N\}$. Since $p_i > 0, \forall i = 1, ..., N$ and the graph is strongly connected, $p_i a_{ij}(y_i - y_j)^T(y_i - y_j) = 0, \forall i, j = 1, ..., N$ implies that $y_i = y_j, \forall i, j = 1, ..., N$.

Remark 7.3. Theorem 7.2 extends the result in Theorem 7.1 to general strongly connected graphs. This is achieved by simply modifying the Lyapunov function (7.16) to (7.18), which weights the node storage functions by the elements p_i of the first left eigenvector of the graph Laplacian matrix *L*. This shows the importance of the elements p_i in preserving the passivity properties of multi-agent systems on digraphs.

References

[1] K.J. Astrom and B. Wittenmark, *Adaptive Control*, 2nd edition, Dover, Mineola, NY, 2008.

[2] Bernstein D (2009) Matrix mathematics: theory, facts, and formulas, 2nd edn. Princeton University Press, Princeton, NJ.

[3] Chopra N, Spong M (2006) Passivity-based control of multi-agent systems. In: Kawamura S, Svinin M (ed) Advances in robot control: from everyday physics to human-like movements. Springer-Verlag, Berlin.

[4] Godsil C, Royle G (2001) Algebraic graph theory. Springer-Verlag, New York.

[5] Horn R, Johnson C (1990) Matrix analysis. Cambridge University Press, New York.

[6] Igarashi Y, Hatanaka T, Fujita M, Spong M (2008) Passivity-based output synchronization and flocking algorithm in SE (3). In: Proceedings of the American Control Conference, pp. 723–728, Seattle, WA.

[7] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*, SIAM Press, Philadelphia, 2006.

[8] Jadbabaie A, Lin J, Morse A (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Transactions on Automatic Control, 48 (6): 988–1001.

[9] Khalil H (2002) Nonlinear Systems, 3rd edn. Prentice Hall, Upper Saddle River, NJ.

[10] I.D. Landau, R. Lozano, and M.M. Saad, *Adaptive Control*, Springer-Verlag, Berlin, 2011.

[11] Olfati-Saber R, Murray R (2003) Consensus protocols for networks of dynamic agents. In: Proceedings of the American Control Conference, pp. 951–956, Denver, CO.

[12] Olfati-Saber R, Murray R (2004) Consensus problems in networks of agents with switching topology and time-delays. IEEE Transactions on Automatic Control, 49(9): 1520–1533.

[13] Qu Z (2009) Cooperative control of dynamical systems: applications to autonomous vehicles. Springer-Verlag, London.

[14] Ren W, Beard R (2008) Distributed consensus in multi-vehicle cooperative control: theory and applications. Springer-Verlag, London.

[15] Ren W, Beard R, Atkins E (2007) Information consensus in multivehicle cooperative control. IEEE Control Systems Magazine, 27(2): 71–82.

[16] Spong M, Hutchinson S, Vidyasagar M (2006) Robot modeling and control. Wiley, New York.

[17] Zhang H, Lewis F, Qu Z (2012) Lyapunov, adaptive, and optimal design techniques for cooperative systems on directed communication graphs. IEEE Transactions on Industrial Electronics, 59(7): 3026-3041.

8

Cooperative Adaptive Control for Systems with First-Order Nonlinear Dynamics

In cooperative control systems, the control protocol for each agent is allowed to depend only on information about itself and its neighbors in the graph topology. We have confronted this problem for optimal cooperative control design in Part I of the book. In cooperative adaptive control systems, there is an additional problem. In adaptive control systems, the control law depends on unknown parameters that are tuned online in real time to improve the performance of the controller. The challenge in cooperative adaptive control is to make sure that both the control protocols and the parameter tuning laws are distributed in terms of the allowed graph topology. That is, they are allowed to depend only on locally available information about the agent and its neighbors. We shall see in this chapter that the key to the design of distributed adaptive tuning algorithms is the selection of suitable Lyapunov functions that depend on the graph topology. This is closely connected to the selection of global performance indices that depend on the graph topology in Chapter 5. The basis for the selection of suitable graph-dependent Lyapunov functions was laid in the discussion on the graph Laplacian potential in Chapter 7.

In Part I, the agent dynamics were taken as homogeneous, that is, all agents had the same dynamics. Moreover, the dynamics were assumed to be known and were used to compute the control protocols. By contrast, in Part II of this book the agent dynamics are taken as heterogeneous, that is, non-identical. Moreover, they are unknown, meaning that the dynamics are not allowed to appear in the control protocols. Finally, the agent dynamics may be affected by unknown disturbances. Synchronization control protocols for multi-agent systems with heterogeneous unknown dynamics are designed here using adaptive control techniques.

The results in this chapter were published first in [2].

8.1 Synchronization Control Formulation and Error Dynamics

In this section we formulate the synchronization control problem for multi-agent cooperative systems linked by an interaction graph structure. Unlike in Part I, the agent dynamics are heterogeneous, that is, non-identical. Moreover, they are unknown, meaning that the dynamics are not allowed to appear in the control protocols. This section formulates the basic structure of the distributed control protocols to confront these problems. First, we develop the error dynamics for synchronization, which are key in the design of cooperative control protocols. Based on the error dynamics, we derive the structure of a distributed control law that is suitable for synchronization control for heterogeneous agents with unknown dynamics. These control protocols depend on unknown parameters. In the next section it is shown how to learn these unknown parameters online to guarantee synchronization by using adaptive control techniques.

8.1.1 Graph Theory Basics

We start with a review of some graph theory basics. Consider a graph $G = (V, E)$ with a nonempty finite set of N nodes $V = \{v_1, \cdots, v_N\}$ and a set of edges or arcs $E \subseteq V \times V$. We assume the graph is simple, e.g. no repeated edges and $(v_i, v_i) \notin E, \forall i$ no self-loops. General directed graphs are considered. Denote the connectivity matrix as $A = [a_{ij}]$ with $a_{ij} > 0$ *if* $(v_j, v_i) \in E$ and $a_{ij} = 0$ otherwise. Note $a_{ii} = 0$. The set of neighbors of a node v_i is $N_i = \{v_j : (v_j, v_i) \in E\}$, i.e. the set of nodes with arcs incoming to v_i. Define the in-degree matrix as a diagonal matrix $D = [d_i]$ with $d_i = \sum_{j \in N_i} a_{ij}$ the weighted in-degree of node i (i.e. i *-th* row sum of A). Define the graph Laplacian matrix as $L = D - A$, which has all row sums equal to zero. Define $d_i^o = \sum_j a_{ji}$, the (weighted) out-degree of node i, that is the i *-th* column sum of A.

We assume the communication digraph is strongly connected, i.e. there is a directed path from v_i to v_j for all distinct nodes $v_i, v_j \in V$. Then A and L are irreducible [14]. That is they are not cogredient to a lower triangular matrix, i.e., there is no permutation matrix U such that

$$L = U \begin{pmatrix} * & 0 \\ * & * \end{pmatrix} U^T \tag{8.1}$$

The results of this research can easily be extended to graphs having a spanning tree (i.e. not necessarily strongly connected) using the Frobenius form shown in (8.1) [14].

8.1.2 Synchronization Control Problem

Consider node dynamics defined for the i-th node as

$$\dot{x}_i = f_i(x_i) + u_i + w_i(t) \tag{8.2}$$

where $x_i(t) \in R$ is the state of node i, $u_i(t) \in R$ is the control input, and $w_i(t) \in R$ is an unknown disturbance acting upon each node. Note that each node may have its own distinct dynamics. This is referred to as non-identical or heterogeneous agent dynamics. Standard assumptions for existence of unique solutions are made, e.g. $f_i(x_i)$ is either continuously differentiable or Lipschitz.

It is assumed that the agent drift dynamics $f_i(x_i)$ are unknown. Thus, they are not available for use in the control protocols designed in this chapter.

It is assumed for ease of notation that the agents' states are scalars, $x_i(t) \in R$. If they states are vectors $x_i(t) \in R^n$, the results of this chapter can be extended using standard methods involving the Kronecker products employed in various chapters in Part I of the book.

The overall global graph dynamics is

$$\dot{x} = f(x) + u + w \tag{8.3}$$

where the overall (global) state vector is $x = (x_1, x_2, \cdots, x_N)^T \in R^N$, global node dynamics vector is $f(x) = \{f_1(x_1), f_2(x_2), \cdots, f_N(x_N)\}^T \in R^N$, input $u = (u_1, u_2, \cdots, u_N) \in R^N$, and $w = (w_1, w_2, \cdots, w_N) \in R^N$.

Definition 8.1 The *local neighborhood synchronization error* for node *i is* defined as [12],[10]

$$e_i = \sum_{j \in N_i} a_{ij}(x_j - x_i) + b_i(x_0 - x_i) \tag{8.4}$$

with x_0 the state of a leader or control node, pinning gains $b_i \geq 0$, and $b_i > 0$ for at least one agent i who has direct knowledge of the state of the leader [12],[19]. Then, $b_i \neq 0$ if and only if there exist an arc from the control node to the i-th

node in G. We refer to the nodes i for which $b_i \neq 0$ as the pinned or controlled nodes

Note that (8.4) contains the entirety of the information that is available to any node i for control purposes.

A desired synchronization trajectory is given as the state $x_0(t)$ of a leader or control node having non-autonomous command generator dynamics

$$\dot{x}_0 = f(x_0, t) \tag{8.5}$$

A special case is the standard constant consensus value with $\dot{x}_0 = 0$. The drift term $f(x_0, t)$ could represent, e.g. motion dynamics of the control node. That is, we assume that the control node can have a time-varying state. The interesting cases are when the leader's trajectory does not converge to zero. These command generator dynamics can produce a wide variety of reference trajectories, including position step commands, velocity ramp commands, sinusoidal trajectories, and so on.

The **distributed synchronization control design problem** confronted herein is as follows: Design control protocols for all the nodes in G to synchronize to the state of the control node, i.e. one requires $x_i(t) \to x_0(t), \forall i$. The control protocols must be distributed in the sense that they can only depend on local information about the agent and its neighbors in the graph. It is assumed that the dynamics of the control node is unknown to any of the nodes in G. It is assumed further that both the node nonlinearities $f_i(.)$ and the node disturbances $w_i(t)$ are unknown. Thus, the synchronization protocols cannot contain any of the dynamics of agents or leader, and so must be robust to unmodeled dynamics and unknown disturbances.

In fact, (8.5) is a command generator. Therefore we are considering a distributed command generator tracker problem with unknown node dynamics and unknown command generator dynamics.

8.1.3 Synchronization Error Dynamics

From (8.4), the global error vector for the network is given by

$$e = -(L+B)(x - \underline{1}x_0) = -(L+B)(x - \underline{x}_0) \tag{8.6}$$

where, $e = \begin{bmatrix} e_1 & e_2 & \cdots & e_N \end{bmatrix}^T \in R^N$ and $\underline{x}_0 = \underline{1}x_0 \in R^N$. $B \in R^{N \times N}$ is a diagonal matrix with diagonal entries b_i and $\underline{1}$ is the N-vector of ones.

Differentiating (8.6) one obtains the synchronization error dynamics

$$\dot{e} = -(L+B)(\dot{x}-\underline{\dot{x}}_0) = -(L+B)\left[f(x)-\underline{f}(x_0,t)+u+w(t)\right] \qquad (8.7)$$

where $\underline{f}(x_0,t) = \underline{1}f(x_0(t),t) \in R^N$.

Remark 8.1 If the node states are vectors $x_i(t) \in R^n$ and $x_0(t) \in R^n$, then $x,e \in R^{nN}$ and (8.6) becomes

$$\dot{e} = -\left[(L+B)\otimes I_n\right]\left[f(x)-\underline{f}(x_0,t)+u+w(t)\right] \qquad (8.8)$$

with $\otimes$ the Kronecker product. To avoid obscuring the essentials, throughout the research we take the node states as scalars, $x_i(t) \in R$. If they are vectors, all of the following development is easily modified by introducing the Kronecker product terms as appropriate. In fact, a simulation example is presented for the case $x_i(t) \in R^2$, namely 1-D motion control for coupled inertial agents.

Remark 8.2 Note that

$$\delta = (x-\underline{x}_0) \qquad (8.9)$$

is the disagreement vector in [13]. We do not use this error herein because it is a global quantity that cannot be computed locally at each node, in contrast to the local neighborhood error (8.4). As such, it is suitable for analysis but not for distributed adaptive controls design using Lyapunov techniques.

Remark 8.3 We take the communication digraph as strongly connected. Therefore, if $b_i \neq 0$ for at least one i then $(L+B)$ is an irreducibly diagonally dominant M-matrix and hence nonsingular [14]. It has all poles in the open right-half s-plane. A milder condition that guarantees $(L+B)$ is nonsingular is existence of a spanning tree with $b_i \neq 0$ for at least one root node i.

An M-matrix is a square matrix having its off-diagonal entries nonpositive and all principal minors nonnegative. Based on Remark 8.3, the next result is therefore obvious from (8.6) and the Cauchy Schwartz inequality. (see also [10].

Lemma 8.1 Let the graph be strongly connected and $B \neq 0$. Then
$$\|\delta\| \leq \|e\|/\underline{\sigma}(L+B) \qquad (8.10)$$

with $\underline{\sigma}(L+B)$ the minimum singular value of $(L+B)$, and $e = 0$ if and only if the nodes synchronize, that is

$$x(t) = \underline{x}_0(t) \tag{8.11}$$

8.1.4 Synchronization Control Design

The control problem confronted in this chapter is to design a control strategy so that local neighborhood error $e(t)$ is bounded to a small residual set. Then according to Lemma 8.1 all nodes synchronize so that $\|x_i(t) - x_0(t)\|$ is small $\forall i$. It is assumed that the dynamics $f(x_0, t)$ of the control node is unknown to any of the nodes in G. It is assumed further that both the node nonlinearities $f_i(.)$ and the node disturbances $w_i(t)$ are unknown. As such, the synchronization protocols must be robust to unmodeled dynamics and unknown disturbances.

To achieve this goal, define the input u_i for node i as

$$u_i = v_i - \hat{f}_i(x_i) \tag{8.12}$$

where, $\hat{f}_i(x_i)$ is an estimate of $f_i(x_i)$ and $v_i(t)$ is an auxiliary control signal to be designed via Lyapunov techniques in Theorem 8.1. This can be written in vector form for the overall network of N nodes as

$$u = v - \hat{f}(x) \tag{8.13}$$

where $v = (v_1, v_2, \dots, v_N)^T \in R^N$ and $\hat{f}(x) = \{\hat{f}_1(x_1), \hat{f}_2(x_2), \dots, \hat{f}_N(x_N)\}^T \in R^N$. Then from (8.7) one gets

$$\dot{e} = -(L + B)\left[f(x) - \underline{f}(x_0, t) - \hat{f}(x) + v + w \right] \tag{8.14}$$

Following the techniques in [4],[5],[11], assume that the unknown nonlinearities in (8.2) are locally smooth and thus can be approximated on a compact set $\Omega_i \in R$ by

$$f_i(x_i) = W_i^T \varphi_i(x_i) + \varepsilon_i \tag{8.15}$$

with $\varphi_i(x_i) \in R^{v_i}$ a suitable basis set of v_i functions at each node i and $W_i \in R^{v_i}$ a set of unknown coefficients. The approximation error is ε_i.

According to the neural network (NN) approximation literature [4],[5],[11], a variety of basis sets can be selected, including sigmoids, gaussians, etc. There

$\varphi_i(x_i) \in R^{\nu_i}$ is known as the NN activation function vector and $W_i \in R^{\nu_i}$ as the NN weight matrix. The ideal approximating weights $W_i \in R^{\nu_i}$ in (8.15) are assumed unknown. The NN in (8.15) are local at each agent. For computational efficiency, the intention is to select only a small number ν_i of NN neurons at each node (see Simulation Examples).

To compensate for unknown nonlinearities, each node will maintain a neural network locally to keep track of the current estimates for the nonlinearities. The idea is to use the information of the states from the neighbors of node i to evaluate the performance of the current control protocol along with the current estimates of the nonlinear functions. Therefore, select the local node's approximation $\hat{f}_i(x_i)$ as

$$\hat{f}_i\left(x_i\right) = \hat{W}_i^T \varphi_i(x_i) \tag{8.16}$$

where $\hat{W}_i \in R^{\nu_i}$ is a current estimate of the NN weights for node i, and ν_i is the number of NN neurons maintained at each node i. The control protocol for agent i now becomes

$$u_i = v_i - \hat{W}_i^T \varphi_i(x_i) \tag{8.17}$$

It will be shown in Theorem 8.1 how to select the estimates of the parameters $\hat{W}_i \in R^{\nu_i}$ using the local neighborhood synchronization errors (8.4) to guarantee synchronization.

The global node nonlinearity $f(x)$ for the entire graph G is now written as

$$f(x) = W^T \varphi(x) + \varepsilon \tag{8.18}$$

where $W^T = diag\{W_i^T\}$, $\varphi(x) = \{\varphi_1^T\left(x_1\right), \varphi_2^T\left(x_2\right), ..., \varphi_N^T\left(x_N\right)\}^T$, and $\varepsilon = \{\varepsilon_1, \varepsilon_2, ..., \varepsilon_N\}^T \in R^N$. The global estimate $\hat{f}(x)$ is

$$\hat{f}(x) = \hat{W}^T \varphi(x) \tag{8.19}$$

with $\hat{W}^T = diag\{\hat{W}_i^T\}$. Now, the error dynamics (8.14) takes the form $v(t)$

$$\dot{e} = -\left(L+B\right)\left[\tilde{f}(x) + v + w(t) - \underline{f}(x_0, t)\right] \tag{8.20}$$

where the parameter estimation error is $\tilde{W}_i = W_i - \hat{W}_i$ and the function estimation error is

$$\tilde{f}(x) = f(x) - \hat{f}(x) = \tilde{W}^T \varphi(x) + \varepsilon \tag{8.21}$$

with $\tilde{W}^T = W^T - \hat{W}^T$. Therefore, one obtains finally the error dynamics

$$\dot{e} = -\left(L + B\right)\left[\tilde{W}^T \varphi(x) + v + \varepsilon + w(t) - \underline{f}(x_0, t)\right] \tag{8.22}$$

with $v(t)$ an auxiliary control yet to be designed.

8.2 Adaptive Design and Distributed Tuning Law

In this section we show how to select the auxiliary controls $v_i(t)$ and the tuning laws for the NN weights $\hat{W}_i$ in the protocols (8.17) so as to guarantee that all nodes synchronize to the desired control node signal, i.e., $x_i(t) \to x_0(t), \forall i$. It is assumed that the dynamics $f(x_0, t)$ of the control node (which could represent its motion) are unknown to any of the nodes in G. The Lyapunov analysis approaches of [8],[11] are used, though there are some complications arising from the fact that $v(t)$ and the NN weight tuning laws must be implemented as distributed protocols. This entails a careful selection of the Lyapunov function. The background for the choice of suitable Lyapunov functions for cooperative control is given in Chapter 7.

The singular values of a matrix M are denoted $\sigma_i(M)$ with $\bar{\sigma}(M)$ the maximum singular value and $\underline{\sigma}(M)$ the minimum singular value. The Frobenius norm is $\|M\|_F = \sqrt{tr\{M^T M\}}$ with $tr\{\cdot\}$ the trace. The Frobenius inner product of two matrices is $\langle M_1, M_2 \rangle_F = \sqrt{tr\{M_1^T M_2\}}$.

The following Fact gives two standard results used in neural adaptive control [11].

Fact 8.2 Let the nonlinearities $f(x)$ in (8.18) be smooth on a compact set $\Omega \in R^N$. Then:

The NN estimation error $\varepsilon(x)$ is bounded by $\|\varepsilon\| \leq \varepsilon_M$ on Ω, with ε_M a fixed bound [4],[11].

Weierstrass higher-order approximation Theorem [17]. Select the activation functions $\varphi(x)$ as a complete independent basis (e.g. polynomials). Then NN estimation error $\varepsilon(x)$ converges uniformly to zero on Ω as $v_i \to \infty, i = 1, N$. That is $\forall \xi > 0$ there exist $\bar{v}_i, i = 1, N$ such that $v_i > \bar{v}_i, \forall i$ implies $\sup_{x \in \Omega} \|\varepsilon(x)\| < \xi$.

The following standard assumptions are required. Although the bounds mentioned are assumed to exist, they are not used in the design and do not have to be known. They appear in the error bounds in the proof of Theorem 8.1. (Though not required, if desired, standard methods can be used to estimate these bounds including [16].)

Assumption 8.1

a. The unknown disturbance w_i is bounded for all i. Thus the overall disturbance vector w is also bounded by $\|w\| \leq w_M$ with w_M a fixed bound.

b. The unknown consensus variable dynamics $\underline{f}(x_0, t)$ is bounded so that

$$\|\underline{f}(x_0, t)\| \leq F_M, \forall t.$$

c. The target trajectory is in a bounded region, e.g. $\|x_0(t)\| < X_0, \forall t$, with X_0 a constant bound.

d. Unknown ideal NN weight matrix W is bounded by $\|W\|_F \leq W_M$.

e. NN activation functions φ_i are bounded $\forall i$, so that one can write for the overall network that $\|\varphi\| \leq \phi_M$.

Assumption 8.1.b means that the maximum velocity of the leader node is unknown but bounded above.

The next definitions extend standard notions [8],[11] to synchronization for distributed systems.

Definition 8.2 The global neighborhood error $e(t) \in R^N$ is uniformly ultimately bounded (UUB) if there exists a compact set $\Omega \subset R^N$ so that $\forall e(t_0) \in \Omega$ there exists a bound B and a time $t_f(B, e(t_0))$, both independent of $t_0 \geq 0$, such that $\|e(t)\| \leq B \ \forall t \geq t_0 + t_f$ . .

Definition 8.3 The control node trajectory $x_0(t)$ given by (8.5) is cooperative UUB with respect to solutions of node dynamics (8.2) if there exist a compact set $\Omega \subset R$ so that $\forall \left(x_i(t_0) - x_0(t_0) \right) \in \Omega$ there exist a bound B and a time

$t_f\left(B,\left(x_i\left(t_0\right)-x_i\left(t_0\right)\right)\right)$, both independent of $t_0 \geq 0$, such that $\left\|x_i(t)-x_0(t)\right\| \leq B \ \forall i, \forall t \geq t_0 + t_f$.

The next key constructive result is needed. It extends the discussion about co-operative Lyapunov Functions in Chapter 7 to the case of pinning control, that is, to the cooperative tracker problem. It provides a Lyapunov equation that allows the design of distributed control protocols that only depend on local information available at each node about its own state and those of its neighbors. An M-matrix is a square matrix having nonpositive off-diagonal elements and all principal minors nonnegative.

Lemma 8.2 Lyapunov Functions for Distributed Control on Graphs [14]. *Let* L *be irreducible and* B *have at least one diagonal entry* $b_i > 0$. *Then* $(L+B)$ *is a nonsingular* $M - matrix$. *Define*

$$q = \begin{bmatrix} q_1 & q_2 & \cdots & q_N \end{bmatrix}^T = (L+B)^{-1} \underline{1} \tag{8.23}$$

$$P = \operatorname{diag}\{p_i\} \equiv \operatorname{diag}\{1/q_i\} \tag{8.24}$$

then $P > 0$ and the matrix Q defined as

$$Q = P(L+B)+(L+B)^T P \tag{8.25}$$

is positive definite.   ∎

The main result of this research is given by Theorem 8.1, which shows how to design the control protocols (8.17) and tune the NN weights such that the local neighborhood cooperative errors (8.4) for all nodes are UUB. According to Lemma 8.1 this implies consensus variable $x_0(t)$ is cooperative UUB, thereby showing synchronization and cooperative stability for the whole network G.

Theorem 8.1: Distributed Adaptive Control Protocol for Synchronization.
Consider the networked systems given by (8.2), (8.3) under Assumption 8.1. Let the communication digraph be strongly connected. Select the auxiliary control signals in (8.17) as $v_i(t) = ce_i(t)$ *with the neighborhood synchronization errors* $e_i(t)$ *defined in (8.4) so that the local node control protocols are given by*

$$u_i = ce_i - \hat{f}_i(x_i) = c\sum_{j \in N_i} a_{ij}\left(x_j - x_i\right) + cb_i\left(x_0 - x_i\right) \ -\hat{W}_i^T \varphi_i(x_i) \tag{8.26}$$

or

$$u = ce - \hat{W}^T \varphi(x) \tag{8.27}$$

with control gains $c > 0$. *Let local node NN tuning laws be given by*

$$\dot{\hat{W}}_i = -F_i \varphi_i e_i^T p_i (d_i + b_i) - \kappa F_i \hat{W}_i \tag{8.28}$$

with $F_i = \Pi_i I_{v_i}$, I_{v_i} *the* $v_i \times v_i$ *identity matrix,* $\Pi_i > 0$ *and* $\kappa > 0$ *scalar tuning gains, and* $p_i > 0$ *defined in Lemma 8.2. Select* $\kappa = \frac{1}{2} c \underline{\sigma}(Q)$ *and the control gain c so that*

$$c\underline{\sigma}(Q) > \tfrac{1}{2} \phi_M \bar{\sigma}(P) \bar{\sigma}(A) \tag{8.29}$$

with P>0, Q>0 the matrices in Lemma 8.2 and A the graph adjacency matrix. Then there exist numbers of neurons $\bar{v}_i, i = 1, N$ *such that for* $v_i > \bar{v}_i, \forall i$ *the over-all local cooperative error vector e(t) and the NN weight estimation errors* $\tilde{W}$ *are UUB, with practical bounds given by (8.47) and (8.48) respectively. Therefore the control node trajectory* $x_0(t)$ *is cooperative UUB and all nodes synchronize close to* $x_0(t)$. *Moreover, the bounds on local consensus errors (8.4) can be made small by increasing the control gains c.*

Proof.

Part a. We claim that for a fixed $\varepsilon_M > 0$, there exist numbers of neurons $\bar{v}_i, i = 1, N$ such that for $v_i > \bar{v}_i, \forall i$ the NN approximation error is bounded by $\|\varepsilon\| \leq \varepsilon_M$. The claim is proven in Part b of the Proof 8.1. Consider now the Lyapunov function candidate

$$V = \frac{1}{2} e^T P e + \frac{1}{2} tr \left\{ \tilde{W}^T F^{-1} \tilde{W} \right\} \tag{8.30}$$

with *e(t)* the vector of local neighborhood cooperative errors (8.4), $0 < P = P^T \in R^{N \times N}$ the diagonal matrix defined in (8.24), and F^{-1} a block diagonal matrix defined in terms of $F = diag\{F_i\}$. Then,

$$\dot{V} = e^T P \dot{e} + tr \left\{ \tilde{W}^T F^{-1} \dot{\tilde{W}} \right\} \tag{8.31}$$

and from (8.22)

$$\dot{V} = -e^T P (L + B) \left[\tilde{W}^T \varphi(x) + ce + \varepsilon + w - \underline{f}(x_0, t) \right] + tr \left\{ \tilde{W}^T F^{-1} \dot{\tilde{W}} \right\}$$
$$\tag{8.32}$$

$$\dot{V} = -ce^T P(L+B)e - e^T P(L+B)\{\varepsilon + w - \underline{f}(x_0,t)\} -$$
$$e^T P(L+B)\tilde{W}^T \varphi(x) + tr\{\tilde{W}^T F^{-1}\dot{\tilde{W}}\} \tag{8.33}$$

$$\dot{V} = -ce^T P(L+B)e - e^T P(L+B)\{\varepsilon + w - \underline{f}(x_0,t)\} +$$
$$tr\{\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} - \varphi e^T P(L+B)\right)\} \tag{8.34}$$

$$\dot{V} = -ce^T P(L+B)e - e^T P(L+B)\{\varepsilon + w - \underline{f}(x_0,t)\} +$$
$$tr\{\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} - \varphi e^T P(D+B-A)\right)\} \tag{8.35}$$

Since L is irreducible and B has at least one diagonal entry $b_i > 0$, then $(L+B)$ is a nonsingular $M-$matrix. Defining therefore Q according to (8.25) one has

$$\dot{V} = -\frac{1}{2}ce^T Qe - e^T P(L+B)\{\varepsilon + w - \underline{f}(x_0,t)\} +$$
$$tr\{\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} - \varphi e^T P(D+B)\right)\} + tr\{\tilde{W}^T \varphi e^T PA\} \tag{8.36}$$

Adopt now the NN weight tuning law (8.28) or $\dot{\hat{W}}_i = F_i\varphi_i e_i^T p_i(d_i + b_i) + \kappa F_i\hat{W}_i$. Since P and $(D+B)$ are diagonal and $\dot{\tilde{W}}$ has the form in (8.19), one has

$$\dot{V} = -\frac{1}{2}ce^T Qe - e^T P(L+B)\{\varepsilon + w(t) - \underline{f}(x_0,t)\} +$$
$$\kappa tr\{\tilde{W}^T(W - \tilde{W})\} + tr\{\tilde{W}^T \varphi e^T PA\} \tag{8.37}$$

Therefore, for fixed $\varepsilon_M > 0$

$$\dot{V} \le -\frac{1}{2}c\underline{\sigma}(Q)\|e\|^2 + \|e\|\bar{\sigma}(P)\bar{\sigma}(L+B)(\varepsilon_M + w_M + F_M) +$$
$$\kappa W_M\|\tilde{W}\|_F - \kappa\|\tilde{W}\|_F^2 + \|\tilde{W}\|_F\|e\|\phi_M\bar{\sigma}(P)\bar{\sigma}(A) \tag{8.38}$$

Then

$$\dot{V} \le -\begin{bmatrix}\|e\| & \|\tilde{W}\|_F\end{bmatrix}\begin{bmatrix}\frac{1}{2}c\underline{\sigma}(Q) & -\frac{1}{2}\phi_M\bar{\sigma}(P)\bar{\sigma}(A) \\ -\frac{1}{2}\phi_M\bar{\sigma}(P)\bar{\sigma}(A) & \kappa\end{bmatrix}\begin{bmatrix}\|e\| \\ \|\tilde{W}\|_F\end{bmatrix}$$
$$+\begin{bmatrix}B_M\bar{\sigma}(P)\bar{\sigma}(L+B) & \kappa W_M\end{bmatrix}\begin{bmatrix}\|e\| \\ \|\tilde{W}\|_F\end{bmatrix}$$

with $B_M \equiv \varepsilon_M + w_M + F_M$. Write this as

$$\dot{V} \leq -z^T R z + r^T z \tag{8.39}$$

Then $\dot{V} \leq 0$ if R is positive definite and

$$\|z\| > \frac{\|r\|}{\underline{\sigma}(R)} \tag{8.40}$$

According to (8.30) one has

$$\tfrac{1}{2}\underline{\sigma}(P)\|e\|^2 + \frac{1}{2\Pi_{max}}\|\tilde{W}\|_F^2 \leq V \leq \tfrac{1}{2}\bar{\sigma}(P)\|e\|^2 + \frac{1}{2\Pi_{min}}\|\tilde{W}\|_F^2 \tag{8.41}$$

$$\tfrac{1}{2}\begin{bmatrix} \|e\| & \|\tilde{W}\|_F \end{bmatrix} \begin{bmatrix} \underline{\sigma}(P) & 0 \\ 0 & \dfrac{1}{\Pi_{max}} \end{bmatrix} \begin{bmatrix} \|e\| \\ \|\tilde{W}\|_F \end{bmatrix} \leq V \leq$$

$$\tfrac{1}{2}\begin{bmatrix} \|e\| & \|\tilde{W}\|_F \end{bmatrix} \begin{bmatrix} \bar{\sigma}(P) & 0 \\ 0 & \dfrac{1}{\Pi_{min}} \end{bmatrix} \begin{bmatrix} \|e\| \\ \|\tilde{W}\|_F \end{bmatrix} \tag{8.42}$$

with Π_{min}, Π_{max} the minimum and maximum values of Π_i. Define variables to write $\tfrac{1}{2} z^T \underline{S} z \leq V \leq \tfrac{1}{2} z^T \bar{S} z$. Then

$$\tfrac{1}{2}\underline{\sigma}(\underline{S})\|z\|^2 \leq V \leq \tfrac{1}{2}\bar{\sigma}(\bar{S})\|z\|^2 \tag{8.43}$$

Therefore

$$V > \frac{1}{2}\frac{\bar{\sigma}(\bar{S})\|r\|^2}{\underline{\sigma}^2(R)} \tag{8.44}$$

implies (8.40).

One can write the minimum singular values of R as

$$\underline{\sigma}(R) = \frac{\left(\dfrac{1}{2}c\underline{\sigma}(Q)+\kappa\right) - \sqrt{\left(\dfrac{1}{2}c\underline{\sigma}(Q)-\kappa\right) + \dfrac{1}{4}\varphi_M^2 \bar{\sigma}^2(P)\bar{\sigma}^2(A)}}{2} \tag{8.45}$$

To obtain a cleaner form for $\underline{\sigma}(R)$, select $\kappa = \dfrac{1}{2}c\underline{\sigma}(Q)$. Then

$$\underline{\sigma}(R) = \frac{c\underline{\sigma}(Q) - \dfrac{1}{2}\varphi_M \bar{\sigma}(P)\bar{\sigma}(A)}{2} \tag{8.46}$$

which is positive under condition (8.29). Therefore, $z(t)$ is UUB [8],[11].

In view of the fact that, for any vector z, one has $\|z\|_1 \geq \|z\|_2 \geq \cdots \geq \|z\|_\infty$, sufficient conditions for (8.40) are

$$\|e\| > \frac{B_M \overline{\sigma}(P)\overline{\sigma}(L+B) + \kappa W_M}{\underline{\sigma}(R)} \qquad (8.47)$$

or

$$\|\tilde{W}\| > \frac{B_M \overline{\sigma}(P)\overline{\sigma}(L+B) + \kappa W_M}{\underline{\sigma}(R)} \qquad (8.48)$$

Now Lemma 8.1 shows that the consensus errors $\delta(t)$ are UUB. Then $x_0(t)$ is cooperative UUB. Note that increasing gain c decreases the bound in (8.47).

Part b. According to (8.39) $\dot{V} \leq -\underline{\sigma}(R)\|z\|^2 + \|r\|\|z\|$ and according to (8.43)

$$\dot{V} \leq -\alpha V + \beta\sqrt{V} \qquad (8.49)$$

with $\alpha \equiv 2\underline{\sigma}(R)/\overline{\sigma}(\overline{S})$, $\beta \equiv \sqrt{2}\|r\|/\sqrt{\underline{\sigma}(S)}$. Thence

$$\sqrt{V(t)} \leq \sqrt{V(0)}e^{-\alpha t/2} + \tfrac{\beta}{\alpha}(1 - e^{-\alpha t/2}) \leq \sqrt{V(0)} + \tfrac{\beta}{\alpha}$$

Using (8.43) one has

$$\|e(t)\| \leq \|z(t)\| \leq \sqrt{\frac{\overline{\sigma}(\overline{S})}{\underline{\sigma}(S)}}\sqrt{\|e(0)\|^2 + \|\tilde{W}(0)\|_F^2} + \frac{\overline{\sigma}(\overline{S})}{\underline{\sigma}(S)}\frac{\|r\|}{\underline{\sigma}(R)}$$

Then (8.6) shows that

$$\|x(t)\| \leq \frac{1}{\underline{\sigma}(L+B)}\|e(t)\| + \sqrt{N}\|x_0(t)\|$$

$$\|x(t)\| \leq \frac{1}{\underline{\sigma}(L+B)}\left[\sqrt{\frac{\overline{\sigma}(\overline{S})}{\underline{\sigma}(S)}}\sqrt{\|e(0)\|^2 + \|\tilde{W}(0)\|_F^2} + \frac{\overline{\sigma}(\overline{S})}{\underline{\sigma}(S)}\frac{\|r\|}{\underline{\sigma}(R)}\right] + \sqrt{N}X_0 \equiv r_0$$

$$(8.50)$$

where $\|r\| \leq B_M \overline{\sigma}(P)\overline{\sigma}(L+B) + \kappa W_M$. Therefore, the state is contained for all times $t \geq 0$ in a compact set $\Omega_0 = \{x(t) \mid \|x(t)\| \leq r_0\}$. According to the Weierstrass approximation theorem, given any NN approximation error bound ε_M there exist numbers of neurons $\overline{v}_i, i = 1, N$ such that $v_i > \overline{v}_i, \forall i$ implies $\sup_{x \in \Omega} \|\varepsilon(x)\| < \varepsilon_M$. ∎

Discussion on NN Adaptive Control Proofs for Cooperative Systems. If either (8.47) or (8.48) holds, the Lyapunov derivative is negative and V decreases. Therefore, these provide practical bounds for the neighborhood synchronization

error and the NN weight estimation error respectively. Part a of the Proof 8.1 contains the new material relevant to cooperative control of distributed systems. Part b of the Proof 8.1 is standard in the neural adaptive control literature, see e.g. [11]. Note that the set defined by (8.50) depends on the initial errors and the graph structural properties. Therefore, proper accommodation of large initial errors requires a larger number of neurons in the NN. The proof of Theorem 8.1 also reveals that, for a given number of neurons, the admissible initial condition set is bounded [11].

Lyapunov Functions for Distributed Cooperative Control Design. It is important to select the Lyapunov function candidate V in (8.30) in terms of locally available variables at each node, e.g. the local neighborhood synchronization error $e(t)$ in (8.4) and (8.6). It is also essential to select the kernel matrix P to depend on the graph topology as in Lemma 8.2. This means that any local control signals $v_i(t)$ and NN tuning laws developed in the proof of Theorem 8.1 are distributed and hence implementable at each node using information only about the node's state and those of its neighbors. The use of the Frobenius norm in the Lyapunov function is also instrumental, since it gives rise to Frobenius inner products in the proof of Theorem 8.1 that only depend on trace terms, where only the diagonal terms are important. In fact, the Frobenius norm is ideally suited for the design of distributed protocols. Equation (8.24) shows that P in (8.30) is chosen diagonal. This is important in allowing selection of the NN tuning law (8.28) that leads to expression (8.37).

The result of this Lyapunov design is the local node NN tuning law (8.28) which depends on the local cooperative error $e_i(t)$. By contrast, standard NN tuning laws [11] depend on a global tracking error, equivalent to the disagreement error vector $\delta = (x - \underline{x}_0)$ which is not available at each node. Note that (8.28) is a distributed version of the sigma-mod tuning law of P. Ioannou.

The parameters p_i in (8.28) are computed as in Lemma 8.2, which requires global information of the graph. As such, they are not known locally at the nodes. However, NN tuning parameters $f_i > 0$ are arbitrary, so that $p_i f_i > 0$ can be arbitrary.

Each node maintains a local NN to provide estimates of its nonlinear dynamics. The local nature of the NNs, as reflected in the distributed tuning protocols (8.28), means that the number of neurons v_i at each node can be selected fairly small (see the Simulation Examples). It is not necessary to select a large centralized NN to approximate the full global vector of nonlinearities $f(x)$ in (8.18).

The total number of neurons in all the neural networks in the graph is $v = \sum_{i=1}^{N} v_i$.

The unknown dynamics of the control node $f(x_0, t)$ in (8.5) (which could be, e.g., motion velocity) are treated as a disturbance to be rejected. The proof of Theorem 8.1 shows that even though these dynamics are unknown, synchronization of all nodes to the generally time-varying control node state $x_0(t)$ is guaranteed, within a small error. The simulations corroborate this. The control motion dynamics bound F_M in Assumption 8.1 appears in the term B_M appearing in the UUB error bounds (8.47), (8.48). As F_M becomes smaller, the bounds decrease.

Note that the node control gains c are similar to the pinning gain parameters defined in [19]. Note from (8.45) that increasing these gains results in a smaller synchronization error bound (8.47).

Corollary 8.1 Given the setup in Theorem 8.1, suppose the NN estimation errors in (8.15) are equal to zero. Then under the protocol (8.26) the local neighborhood errors converge to the residual set defined by

$$\|e\| \geq 2B_M \frac{\bar{\sigma}(P)\bar{\sigma}(L+B)}{c\, \underline{\sigma}(Q)} \tag{8.51}$$

with $B_M = d_M + F_M$ the sum of the disturbance bound and the bound on the consensus variable (control node) dynamics (Assumption 8.1) and P, Q defined in Lemma 8.2. Moreover, the synchronization error norm $\|e(t)\|$ convergences to this set exponentially as $\|e(0)\|e^{-\alpha t/2}$ with convergence rate

$$\alpha = c\frac{\underline{\sigma}(Q)}{\bar{\sigma}(P)} \tag{8.52}$$

The speed of synchronization can be increased by increasing the control gains c in protocol (8.26).

Proof. If the NN estimation errors ε_i in (8.15) are zero, then according to the proof of Theorem 8.1 one has

$$V = \tfrac{1}{2}e^T Pe$$

$$\dot{V} \leq -\tfrac{1}{2}c\,\underline{\sigma}(Q)\|e\|^2 + \|e\|\bar{\sigma}(P)\bar{\sigma}(L+P)B_M$$

which is negative if (8.51) holds. One has for large $\|e(t)\|$, approximately

$$V = \tfrac{1}{2}e^T Pe$$

$$\dot{V} \leq -\tfrac{1}{2}c\,\underline{\sigma}(Q)\|e\|^2$$

So that using standard techniques one has

$$\dot{V} \leq -\alpha V$$

with α given by (8.52). Therefore $V(t) \leq V(0)e^{-\alpha t}$ and the errors converge at the rate $\|e(0)\| e^{-\alpha t/2}$. ∎

According to this corollary and Lemma 8.1, the residual global disagreement vector error is practically bounded according to the quantity

$$\|\delta\| \geq 2 B_M \frac{\bar{\sigma}(P)}{c \, \underline{\sigma}(Q)} \tag{8.53}$$

Therefore, the error bounds and the rate of convergence (8.52) are related in a reciprocal manner.

Remark 8.4 If the node states are vectors $x_i \in R^n$ and $x_0 \in R^n$, the control protocols (8.26) and the error bounds given by (8.47) or (8.48) will remain unaltered. The NN weight tuning protocols will then be given by

$$\dot{\hat{W}}_i = -F_i \varphi_i e_i^T \left(p_i (d_i + b_i) \otimes I_n \right) - \kappa F_i \hat{W}_i \tag{8.54}$$

Remark 8.5 It is easy to extend the Theorem 8.1 to the case where the digraph only contains a spanning tree using the Frobenius form [14].

8.3 Relation of Error Bounds to Graph Structural Properties

According to the bounds (8.47), (8.48), (8.51) both the synchronization error and the NN weight estimation error increase with the maximum singular values $\bar{\sigma}(A)$, $\bar{\sigma}(L+B)$, and $\bar{\sigma}(P)$. It is desired to obtain bounds on these singular values in terms of graph properties. Recall that $d_i = \sum_{j \in N_i} a_{ij}$ is the (weighted) in-degree of node i, that is, the i-th row sum of adjacency matrix A, and $d_i^o = \sum_j a_{ji}$ is the (weighted) out-degree of node i, that is the i-th column sum of A. It is direct to obtain upper bounds on these singular values in terms of graph properties. Recalling several results from [1] one can easily show the following.

Lemma 8.3. Bounds on Maximum Singular Values of A and $(L+B)$.

$$\bar{\sigma}(A) \leq \sum_{i=1}^{N} d_i \equiv vol(G)$$

$$\bar{\sigma}(A) \leq \sqrt{\max_i(d_i) \times \max_i(d_i^o)}$$

$$\bar{\sigma}(L+B) \leq \sum_{i-1}^{N} (b_i + d_i + d_i^o)$$

$$\bar{\sigma}(L+B) \leq \sqrt{\max_i(b_i + d_i + d_i^o) \times \max_i(b_i + 2d_i)}$$

These results show that the residual synchronization error is bounded above in terms of the graph complexity expressed in terms of graph volumes (e.g. the sum of in-degrees) and maximum degree sums.

Note from Lemma 8.2 that the singular values of P and Q are related through the Lyapunov equation (8.25) and depend on the singular values of $L+B$. More work needs to be done to investigate these interrelationships and simplify the bounds (8.47), (8.48), (8.51).

8.4 Simulation Examples

This section will show the effectiveness of the distributed adaptive protocol of Theorem 8.1 on several fronts. First, it is compared to standard pinning control [12],[19]. Next, it is shown that the protocol effectively enforces synchronization, using only a few NN nodes at each node, for unknown nonlinear control node dynamics, unknown nonlinear node dynamics, and unknown disturbances at each node.

For this set of simulations, consider the 5-node strongly connected digraph structure in Fig. 8.1 with a leader node connected to node 3. The edge weights and the pinning gain in (8.4) were taken equal to 1.
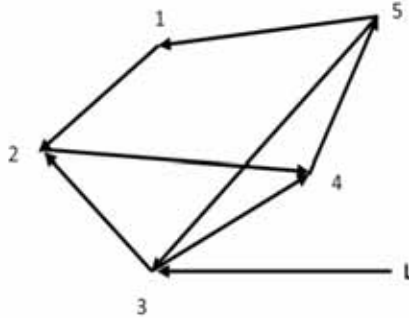

Fig. 8.1 Five node SC digraph with one leader node

a. Nonlinear Node Dynamics And Disturbances

For the graph structure shown consider the following node dynamics

$$\dot{x}_1 = x_1^3 + u_1 + d_1$$
$$\dot{x}_2 = x_2^2 + u_2 + d_2$$
$$\dot{x}_3 = x_3^4 + u_3 + d_3 \qquad (8.55)$$
$$\dot{x}_4 = x_4 + u_4 + d_4$$
$$\dot{x}_5 = x_5^5 + u_5 + d_5$$

which has nonlinearities and disturbances at each node, all assumed unknown. Random disturbances with normal distributions are given at the i-th node by $d_i = \text{randn}(1) \times \cos(t)$. (We use MATLAB symbology.)

Consider now the control protocol of Theorem 8.1. Take the desired consensus value of $x_0 = 2$, i.e. $\dot{x}_0 = f(x_0, t) = 0$. The following parameters are used in the simulation. Control gain $c = 300$, number of neurons at each node: $v_i = 3$, $\kappa = 0.8$, $F_i = 1500$. Fig. 8.2 shows how the agent states reach consensus with bounded input commands. Fig. 8.3 plots the disagreement vector $\delta = (x - \underline{x}_0)$ and the NN estimation errors $f_i(x_i) - \hat{f}_i(x_i)$. Fig. 8.4 shows the NN weights. The initial behavior of NN weight resulted during the search for appropriate NN weights for $\hat{f}_i(x_i)$ estimation. The steady-state consensus error is approximately zero. Table 8.1 shows that at steady-state, the NNs closely estimate the nonlinearities. (That is, the nonlinearities in (8.55) evaluated at the consensus value of $x_0 = 2$. Recall there are small random disturbances $w_i(t)$ present.)
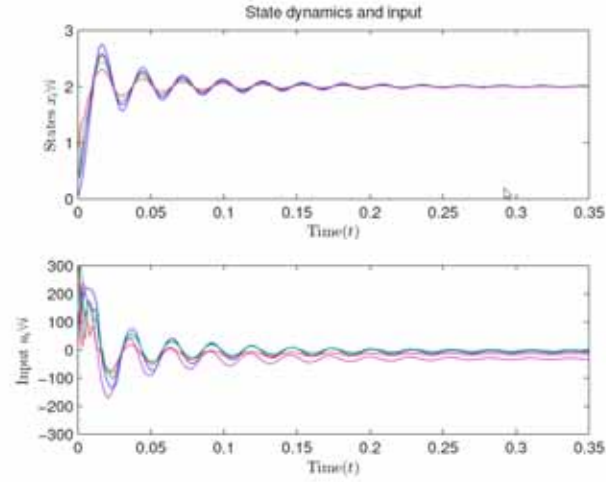
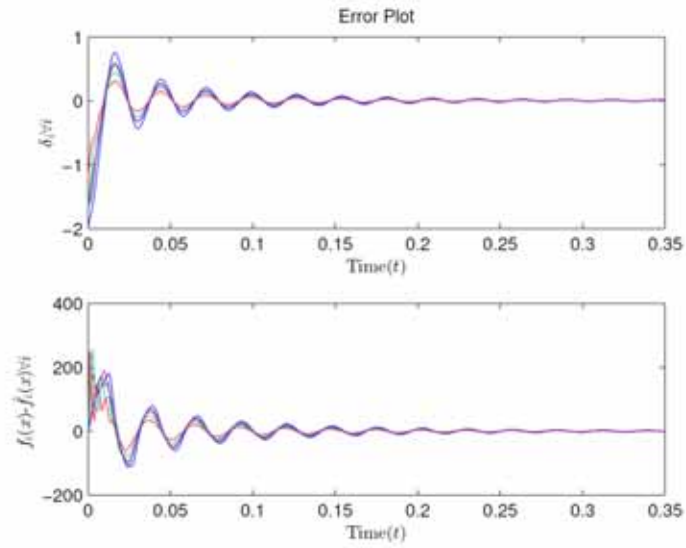**Fig. 8.2** Agents' state dynamics and corresponding input commands
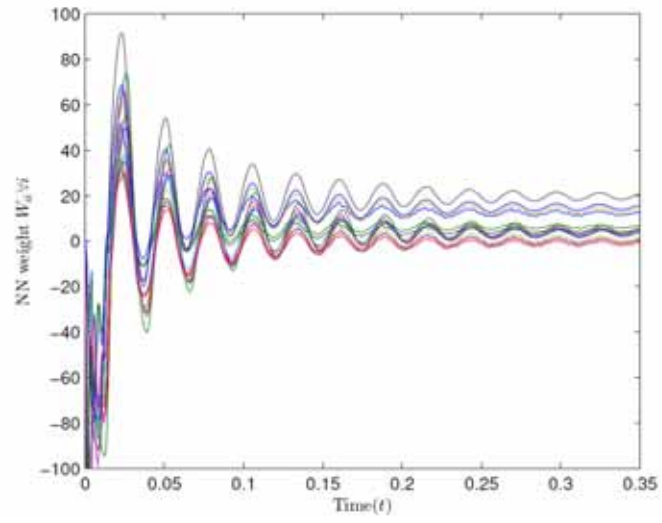

Fig. 8.3 State consensus errors and NN estimation errors

Fig. 8.4 NN weights are stable and bounded after initial search mode

$f_i(x)$	$\hat{f}_i(x) = \text{Est}\left(f_i(x)\right)$
8	7.9835
4	3.8701
16	15.8063
2	1.8051
32	31.5649

Table 8.1. Steady-state values of NN weights closely match the actual agent dynamics non-linearities evaluated at consensus value of $x_0 = 2$.

b. Synchronization of Second-Order Multi-Input Dynamics

Consider the node dynamics for node i given by the second-order inertial system dynamics

$$
\begin{aligned}
\dot{q}_{1_i} &= q_{2_i} + u_{1i} \\
\dot{q}_{2_i} &= J_i^{-1}\left[u_{2i} - B_i^r q_{2_i} - M_i g l_i \sin(q_{1_i}) \right]
\end{aligned}
\tag{8.56}
$$

where $q_i = \left[q_{1_i}, q_{2_i} \right]^T \in R^2$ is the state vector, J_i is the inertia, B_i^r is the damping coefficient, M_i is total mass, g is gravitational acceleration, and l_i is a length parameter. J_i, B_i^r, M_i, l_i, are considered unknown and may be different for each node.

Note that here we take a control input into each state component of each agent, so it is not the same as second-order consensus, where $u_{1_i} = 0$. Second-order consensus using adaptive control is considered in Chapter 9.

The desired target state trajectory is generated by the inertial leader node dynamics

$$
m_0 \ddot{q}_0 + d_0 \dot{q}_0 + k_o q_0 = u_0
\tag{8.57}
$$

Select the feedback linearization input for the leader

$$
u_0 = -\left[K_1 \left(q_0 - \sin(\beta t) \right) + K_2 \left(\dot{q}_0 - \beta \cos(\beta t) \right) \right] + d_0 \dot{q}_0 + k_o q_0 + \beta^2 m_0 \sin(\beta t)
$$

$$(8.58)$$

for a constant $\beta > 0$. Then, the target motion $q_0(t)$ tracks the desired reference trajectory $\sin(\beta t)$.

The cooperative adaptive control law of Theorem 8.1 was simulated. Since the agent states are of second-order, the Kronecker product with I_2 was used as in in (8.54). We used three neurons per node.

Fig. 8.5 verifies that both components (i.e. position and velocity) of the second-order node states synchronize to the leader trajectory. It also shows the control inputs. Fig. 8.6 shows the state consensus error and NN estimation errors. The dynamics of the actual and estimated $f(x)$ are shown in Fig. 8.7. In contrast to the first-order dynamics case of Fig. 8.4, for these second-order systems there are two sets of NN weights as shown in Fig. 8.8. Fig. 8.9 shows the phase-plane plot for all the nodes along with the target node. It can be seen that the phase-plane trajectories of all nodes, started with different initial conditions, synchronize to the target node phase plane trajectory, finally forming a lissajous pattern.
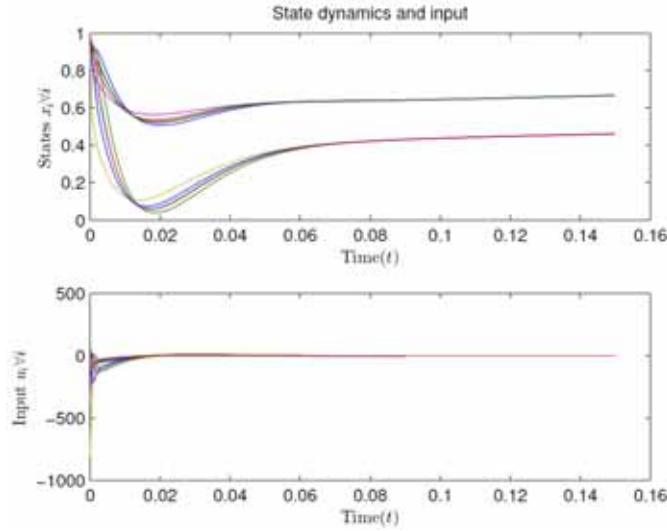

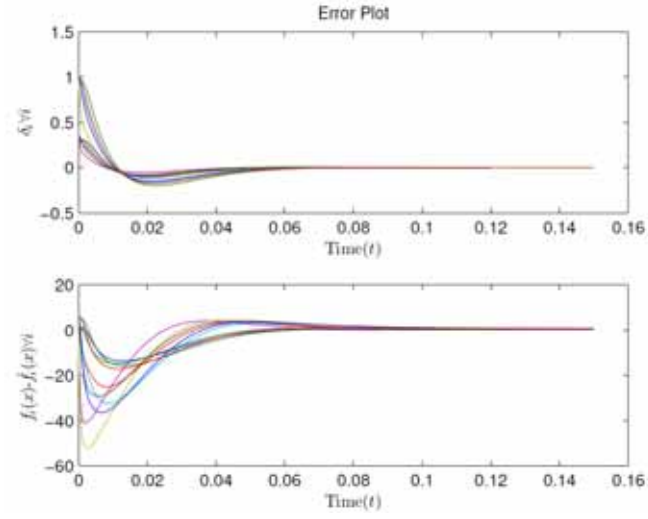**Fig. 8.5** State dynamics and control inputs

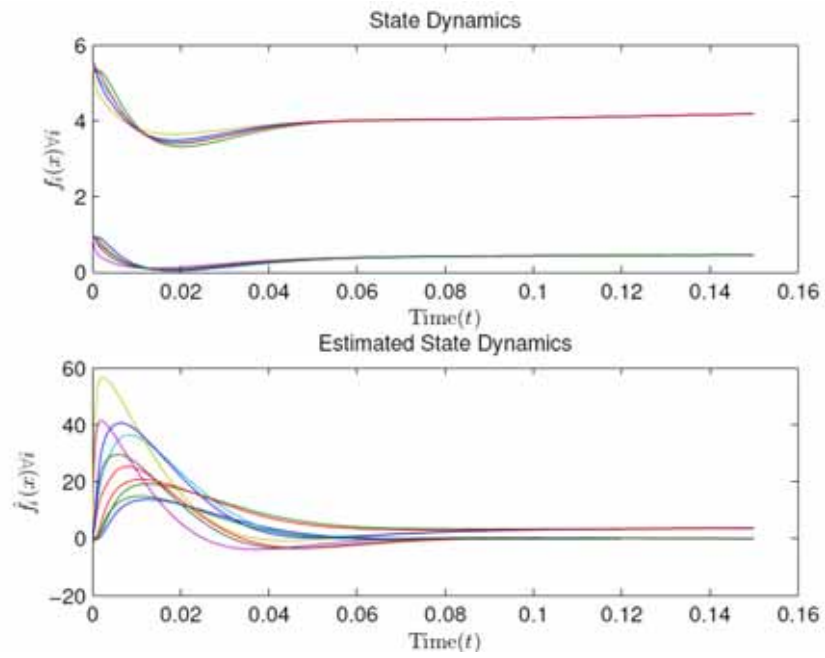**Fig. 8.6** Consensus errors and NN estimation errors.

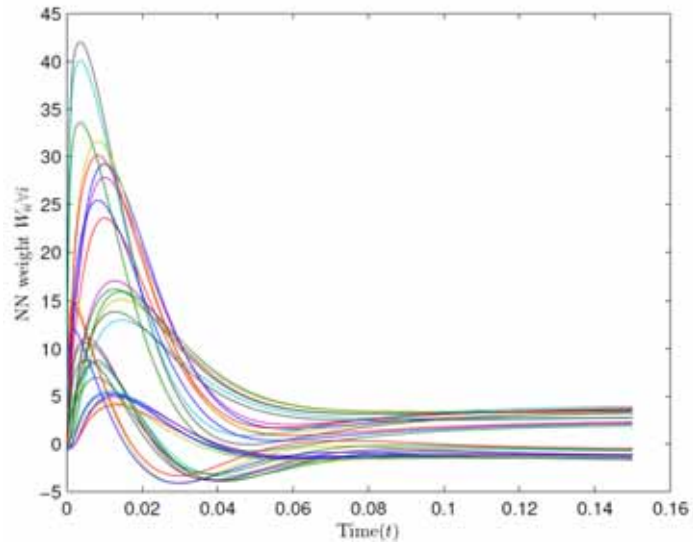**Fig. 8.7** State dynamics and estimated state dynamics

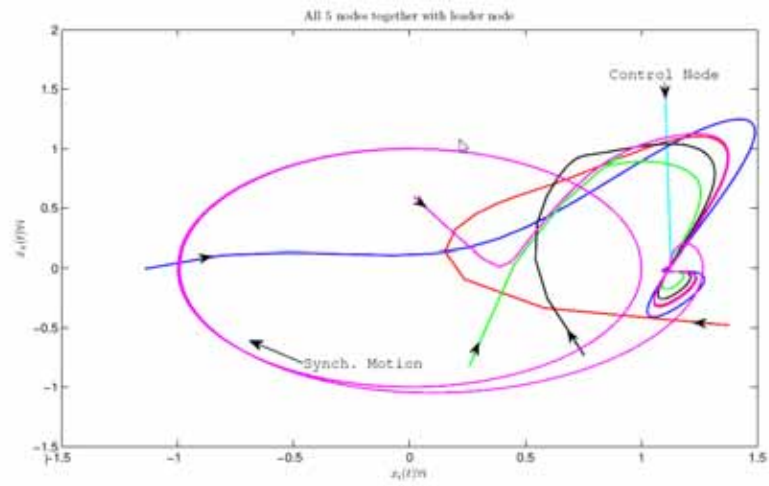**Fig. 8.8** NN weight dynamics: initial searching and stable mode

Fig. 8.9 Synchronized motion phase-plane plot

References

[1] D. S. Bernstein, *Matrix Mathematics*. NJ: Princeton Univ. Press, 2005.

[2] Abhijit Das and F.L. Lewis, "Distributed adaptive control for synchronization of unknown nonlinear networked systems," Automatica, vol. 46, pp. 2014-1021, 2010.

[3] J. A. Fax and R. M. Murray, "Information Flow and Cooperative Control of Vehicle Formations," *IEEE Trans. Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[4] K. Hornik, M. Stinchombe, and H. White, "Multilayer Feedforward Networks are Universal Approximations," *Neural Networks*, vol. 20, pp. 359–366, 1989.

[5] B. Igelnik and Y. H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Transaction of Neural Networks*, vol. 6, pp. 1320–1329, 1995.

[6] A. Jadbabaie, J. Lin, and S. Morse, "Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules," *IEEE Trans. Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[7] N. B. Karayiannis and A. N. Venetsanopoulos, *Artificial neural networks: learning algorithms, performance evaluation, and applications*. Springer, 1993.

[8] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, N.J: Prentice Hall, 2002.

[9] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural network structures for identification of dynamical systems," *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 422–431, Mar. 1995.

[10] S. Khoo, L. Xie, and Z. Man, "Robust Finite-Time Consensus Tracking Algorithm for Multirobot Systems," *IEEE Transaction on Mechatronics*, vol. 14, no. 2, pp. 219–228.

[11] F. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London: Taylor and Francis, 1999.

[12] X. Li, X. Wang, and G. Chen, "Pinning a complex dynamical network to its equilibrium," *IEEE Trans. Circuits and Systems*, vol. 51, no. 10, pp. 2074–2087, 2004.

[13] R. Olfati-Saber and R. M. Murray, "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Transaction of Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[14] Z. Qu, *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. New York: Springer-Verlag, 2009.

[15] W. Ren and R. W. Beard, "Consensus Seeking in Multiagent Systems Under Dynamically Changing Interaction Topologies," *IEEE Trans. Automatic Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[16] G. A. Rovithakis, "Proc. IEEE Int. Symposium on Intelligent Control," Patras, Greece, 2000, pp. 7–12.

[17] M. H. Stone, "The Generalized Weierstrass Approximation Theorem," *Mathematics Magazine*, vol. 21, no. 4,5, pp. 167–184,237–254, 1948.

[18] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1984.

[19] X. F. Wang and G. Chen, "Pinning control of scale-free dynamical networks," *Physica A*, vol. 310, no. 3, pp. 521–531, 2002.

9

Cooperative Adaptive Control for Systems with Second-Order Nonlinear Dynamics

In Chapter 8 we designed cooperative adaptive controllers for multi-agent systems having first-order nonlinear dynamics. In this chapter we study adaptive control for cooperative multi-agent systems having second-order nonidentical nonlinear dynamics. The study of second-order and higher-order consensus is required to implement synchronization in most real world applications such as formation control and coordination among UAVs, where both position and velocity must be controlled. Note that Lagrangian motion dynamics and robotic systems can be written in the form of second-order systems. Moreover, second-order integrator consensus design (as opposed to first-order integrator node dynamics) involves more details about the interaction between the system dynamics and control design problem and the graph structure as reflected in the Laplacian matrix. As such, second-order consensus is interesting because there one must confront more directly the interface between control systems and communication graph structure.

We are interested here in the second-order synchronization tracking problem for heterogeneous nodes with non-identical unknown nonlinear dynamics with unknown disturbances. 'Synchronization control' means the objective of enforcing all node trajectories to follow (in a 'close-enough' sense to be made precise) the trajectory of a leader or control node. The communication structures considered are general directed graphs with fixed topologies. Analysis of digraphs is significantly more involved than for undirected graphs. The dynamics of the leader or command node are also assumed nonlinear and unknown. A distributed adaptive control approach is taken, where cooperative adaptive controllers are designed at each node.

The challenge in cooperative adaptive control is to make sure that both the control protocols and the parameter tuning laws are distributed in terms of the allowed graph topology. That is, they are allowed to depend only depend on locally available information about the agent and its neighbors. We shall see in this chapter that the key to the design of distributed adaptive tuning algorithms is the selection

of suitable Lyapunov functions that depend on the graph topology. This is closely connected to the selection of global performance indices that depend on the graph topology in Chapter 5. The basis for the selection of suitable graph-dependent Lyapunov functions was laid in the discussion on the graph Laplacian potential in Chapter 7.

In Section 9.1 we lay the background for adaptive control design for second-order multi-agent systems by defining local neighborhood position and velocity errors that depend on local information in the graph, and introducing a sliding error variable. Neural network structures at each node are used to approximate the unknown nonlinearities, and a suitable form is developed for a distributed cooperative control protocol. These cooperative control protocols depend on certain unknown parameters. In Section 9.2 we show how to tune these parameters using adaptive tuning laws that only depend on locally available information at each node. In Section 9.3 simulations are given to show the effectiveness of these distributed adaptive control protocols in achieving synchronization for multi-agent systems with unknown heterogeneous dynamics and unknown disturbances.

The results of this chapter were first published in [2].

9.1 Sliding Variable Cooperative Control Formulation and Error Dynamics

Consider a graph $G = (V, E)$ with a nonempty finite set of N nodes $V = \{v_1, \cdots, v_N\}$ and a set of edges or arcs $E \subseteq V \times V$. We assume the graph is simple, e.g. no repeated edges and $(v_i, v_i) \notin E, \forall i$ no self loops. General directed graphs are considered. Denote the adjacency or connectivity matrix as $A = [a_{ij}]$, with $a_{ij} > 0$ *if* $(v_j, v_i) \in E$ and $a_{ij} = 0$ otherwise. Note that $a_{ii} = 0$. The set of neighbors of a node v_i is $N_i = \{v_j : (v_j, v_i) \in E\}$, i.e. the set of nodes with arcs incoming to v_i. Define the in-degree matrix as a diagonal matrix $D = diag\{d_i\}$ with $d_i = \sum_{j \in N_i} a_{ij}$ the weighted in-degree of node i (i.e. i-*th* row sum of A). Define the graph laplacian matrix as $L = D - A$, which has all row sums equal to zero. Define $d_i^o = \sum_j a_{ji}$, the (weighted) out-degree of node, that is the i-th column sum of A.

9.1.1 Cooperative Tracking Problem for Synchronization of Multiagent Systems

Consider second-order node dynamics defined for the i-th node in Brunovsky form as

$$\dot{x}_i^1 = x_i^2$$
$$\dot{x}_i^2 = f_i(x_i) + u_i + w_i$$

(9.1)

where the state is $x_i = [x_i^1 \ x_i^2]^T \in R^2$, $u_i(t) \in R$ is the control input and $w_i(t) \in R$ is a disturbance acting upon each node. Note that each node may have its own distinct nonlinear dynamics $f_i(x_i)$. Standard assumptions for existence of unique solutions are made, e.g. $f_i(x_i)$ either continuously differentiable or Lipschitz. The overall graph dynamics is

$$\dot{x}^1 = x^2$$
$$\dot{x}^2 = f(x) + u + w$$

(9.2)

where the overall (global) velocity vector is $x^2 = \begin{bmatrix} x_1^2 & x_2^2 & \cdots & x_N^2 \end{bmatrix}^T \in R^N$, the global position vector $x^1 = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \end{bmatrix}^T \in R^N$, the global state vector is $x = \left(x^1, x^2 \right)^T$, and the global dynamics vector is $f(x) = \begin{bmatrix} f_1(x_1) & f_2(x_2) & \cdots & f_N(x_N) \end{bmatrix}^T \in R^N$. The global control input is $u = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix}^T \in R^N$, and the global disturbance is $w = \begin{bmatrix} w_1 & w_2 & \cdots & w_N \end{bmatrix}^T \in R^N$.

The state $x_0 = [x_0^1 \ x_0^2]^T \in R^2$ of a leader or control node satisfies the (generally nonautonomous) dynamics in Brunovsky form

$$\dot{x}_0^1 = x_0^2$$
$$\dot{x}_0^2 = f_0(x_0, t)$$

(9.3)

This can be regarded as a command or reference generator. A special case is the standard constant consensus value with $\dot{x}_0^1 = 0$ and x_0^2 absent. Here, we assume that the control node can have a time-varying (nonautonomous) flow field. This command generator captures a wide variety of possible reference trajectories inckuding unit step position commands, unit ramp velocity commands, sinusoidal trajectories, and more.

The **Synchronization tracking control problem** confronted herein is as follows: Design control protocols u_i for all the nodes in G to synchronize to the state of the control node, i.e. one requires $x_i^k(t) \to x_0^k(t), \quad k = 1, 2, \forall i$.

It is assumed that the dynamics $f_0(x_0, t)$ of the control node is unknown to any of the nodes in G. It is assumed further that both the node nonlinearities $f_i(\cdot)$ and the node disturbances $w_i(t)$ are unknown. Thus, the synchronization protocols must be robust to unmodelled dynamics and unknown disturbances.

If the states x_i^k are not scalars, this analysis carries over using standard methods involving the Kronecker product term [1]. See Remark 8.1 in Chapter 8.

We shall proceed by analyzing the error dynamics of a suitable defined cooperative control error.

Definition 9.1 *The local neighborhood tracking synchronization errors for node i are defined as [7], the local position error*

$$e_i^1 = \sum_{j \in N_i} a_{ij}\left(x_j^1 - x_i^1\right) + b_i\left(x_0^1 - x_i^1\right) \qquad (9.4)$$

and the local velocity error

$$e_i^2 = \sum_{j \in N_i} a_{ij}\left(x_j^2 - x_i^2\right) + b_i\left(x_0^2 - x_i^2\right) \qquad (9.5)$$

with pinning gains $b_i \geq 0$, and $b_i > 0$ for at least one i. Then, $b_i \neq 0$ if and only if there exist an arc from the control node to the i-th node in G. We refer to the nodes i for which $b_i \neq 0$ as the pinned or controlled nodes.

Note that (9.4) and (9.5) represent all the information that is available to any node i for control purposes on a distributed graph topology.

9.1.2 Sliding Mode Tracking Error

Define the consensus disagreement error vector

$$\delta = \begin{bmatrix} \delta^1 & \delta^2 \end{bmatrix}^T = \begin{bmatrix} x^1 - \underline{1}x_0^1 & x^2 - \underline{1}x_0^2 \end{bmatrix}^T \qquad (9.6)$$

with $\underline{1} \in R^N$ the vector of 1's. From (9.4), the global position error vector for network G is given by

$$e^1 = -\left(L + B\right)\left(x^1 - \underline{1}x_0^1\right) = -\left(L + B\right)\delta^1 \qquad (9.7)$$

and the global velocity error is

$$e^2 = -(L+B)\left(x^2 - \underline{1}x_0^2\right) = -(L+B)\delta^2 \qquad (9.8)$$

where, $B = diag\{b_i\}$ is the diagonal matrix of pinning gains, and $e^k = \begin{bmatrix} e_1^k & e_2^k & \cdots & e_N^k \end{bmatrix}^T \in R^N, k = 1, 2, \forall i$.

We take the communication digraph as strongly connected. Then, if $b_i \neq 0$ for at least one i then $(L+B)$ is an irreducibly diagonally dominant M-matrix and hence nonsingular [10]. It has all poles in the open right-half s-plane. A milder condition that guarantees $(L+B)$ is nonsingular is existence of a spanning tree with $b_i \neq 0$ for at least one root node i.

Then, the form of (9.7) and (9.8) directly yield the next result.

Lemma 9.1 *Let the graph be strongly connected and* $B \neq 0$. *Then*

$$\left\| \delta^k \right\| \leq \left\| e^k \right\| / \underline{\sigma}(L+B), \qquad k = 1, 2 \qquad (9.9)$$

with $\underline{\sigma}(L+B)$ *the minimum singular value of* $(L+B)$, *and* $e = 0$ *if and only if the nodes synchronize, that is*

$$x_i^k(t) = x_0^k(t), k = 1, 2, \forall i. \qquad (9.10)$$

This lemma shows that if the local neighborhood position and velocity tracking errors e^1, e^2 are small, then the consensus disagreement errors are small and synchronization is reached. Therefore, in the following we shall focus on making e^1, e^2 small.

9.1.3 Synchronization Control Design and Error Dynamics

Differentiating (9.7) and (9.8) yields

$$\dot{e}^1 = -(L+B)\left(x^2 - \underline{1}x_0^2\right) \qquad (9.11)$$

and

$$\dot{e}^2 = -(L+B)\left(\dot{x}^2 - \underline{1}f_0(x_0, t)\right) \qquad (9.12)$$

Note that $\dot{e}^1 = e^2$. Define a sliding mode error for node i as

$$r_i = e_i^2 + \lambda_i e_i^1 \qquad (9.13)$$

or written globally

$$r = e^2 + \Lambda e^1 \tag{9.14}$$

where $\Lambda = \text{diag}(\lambda_i) > 0$. The next result follows directly.

Lemma 9.2 *The velocity error is bounded according to*

$$\left\| e^2 \right\| \leq \left\| r \right\| + \bar{\sigma}(\Lambda) \left\| e^1 \right\| \tag{9.15}$$

Now differentiating r one obtains the error dynamics

$$
\begin{aligned}
\dot{r} &= -(L+B)\left(\dot{x}^2 - \underline{1} f_0(x_0,t) \right) - \Lambda(L+B)\left(x^2 - \underline{1} x_0^2 \right) \\
&= -(L+B)\left(f(x) + u + w \right) + (L+B)\underline{1} f_0(x_0,t) + \Lambda e^2
\end{aligned} \tag{9.16}
$$

Following the techniques in [5],[8],[3] we assume that the unknown nonlinearities in (9.1) are smooth and thus can be approximated on a compact set $S \in R$ by

$$f_i(x_i) = W_i^T \varphi_i(x_i) + \varepsilon_i \tag{9.17}$$

with $\varphi_i(x_i) \in R^{\eta_i}$ a suitable basis set of η_i functions at each node, with η_i the number of neurons and $W_i \in R^{\eta_i}$ a set of unknown coefficients. According to the Weierstrass higher-order approximation theorem [11], a polynomial basis set suffices to approximate $f_i(x_i)$ as well as its derivatives, when they exist. Mmoreover, the approximation error $\varepsilon_i \to 0$ uniformly as $\eta_i \to \infty$. According to the neural network (NN) approximation literature [5], a variety of basis sets can be selected, including sigmoids, gaussians, etc. There, $\varphi_i(x_i) \in R^{\eta_i}$ is known as the NN activation function vector and $W_i \in R^{\eta_i}$ as the NN weight matrix. Then it is shown that ε_i is bounded on a compact set. The ideal approximating weights $W_i \in R^{\eta_i}$ in (9.17) are assumed unknown. The intention is to select only a small number η_i of NN neurons at each node (see Simulations).

Here, to avoid distractions from the main issues being introduced, we assume a linear-in-the-parameters NN, i.e. the basis set of activation functions is fixed and only the output weights are tuned. It is straightforward to use a two-layer NN whereby the first and second-layer weights are tuned. Then one has a nonlinear-in-the-parameters NN and the basis set is automatically selected in the NN. Then, the below development can easily be modified as in [8].

To compensate for unknown nonlinearities, each node will maintain a local neural network to keep track of the current estimates for the nonlinearities. The idea is to use the information of the states from the neighbors of node i to evaluate the performance of the current control protocol along with the current

estimates of the nonlinear functions. Therefore, select the local node's approximation $\hat{f}_i(x_i)$ as

$$\hat{f}_i(x_i) = \hat{W}_i^T \varphi_i(x_i) \qquad (9.18)$$

where $\hat{W}_i \in R^{\eta_i}$ is a current estimate of the NN weights for node i, and η_i is the number of NN neurons maintained at each node i. It will be shown in Theorem 9.1 how to select the estimates of the parameters $\hat{W}_i \in R^{\eta_i}$ using the local neighborhood synchronization errors (9.4), (9.5).

The global node nonlinearity $f(x)$ for graph G is now written as

$$f(x) = W^T \varphi(x) + \varepsilon = \begin{bmatrix} W_1^T & & & \\ & W_2^T & & \\ & & \ddots & \\ & & & \ddots & \\ & & & & W_N^T \end{bmatrix} \underbrace{\begin{bmatrix} \varphi_1(x_1) \\ \varphi_2(x_2) \\ \vdots \\ \vdots \\ \varphi_N(x_N) \end{bmatrix}}_{\varphi(x)} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \vdots \\ \varepsilon_N \end{bmatrix}$$

$$(9.19)$$

and the global estimate $\hat{f}(x)$ as

$$\hat{f}(x) = \hat{W}^T \varphi(x) = \begin{bmatrix} \hat{W}_1^T & & & \\ & \hat{W}_2^T & & \\ & & \ddots & \\ & & & \ddots & \\ & & & & \hat{W}_N^T \end{bmatrix} \underbrace{\begin{bmatrix} \varphi_1(x_1) \\ \varphi_2(x_2) \\ \vdots \\ \vdots \\ \varphi_N(x_N) \end{bmatrix}}_{\varphi(x)} \qquad (9.20)$$

Consider the control protocol

$$u_i = -\hat{f}_i(x_i) + \mu_i(t) \qquad (9.21)$$

or in global form

$$u = -\hat{f}(x) + \mu(t) \qquad (9.22)$$

with $\mu_i(t)$ an auxiliary input for the ith node yet to be specified. Then using (9.22) the global error dynamics (9.16) becomes

$$\dot{r} = -(L+B)\left(\tilde{f}(x) + \mu(t) + w\right) + (L+B)\underline{1}f_0(x_0,t) + \Lambda e^2 \qquad (9.23)$$

where

$$\tilde{f}(x) = f(x) - \hat{f}(x) = \tilde{W}\varphi(x)$$

Is the function approximation error with the weight estimation errors defined as $\tilde{W} = \text{diag}\left(W_1 - \hat{W}_1, W_2 - \hat{W}_2, \cdots\cdots, W_N - \hat{W}_N\right)^T$.

9.2 Cooperative Adaptive Design and Distributed Tuning Laws

It is now shown how to select the auxiliary controls $\mu_i(t)$ and tuning laws for the NN weight parameters $\hat{W}_i \in R^{\eta_i}$ so as to guarantee that all nodes synchronize to the desired control node signal, i.e., $x_i(t) \to x_0(t)$, $\forall i$. Both the auxiliary controls and tuning laws are only allowed to depend on information about node i and its neighbors in the graph. It is assumed that the dynamics $f(x_0, t)$ of the control node (which could represent its motion) are unknown to any of the nodes in G. It is assumed further that the node nonlinearities $f_i(x_i)$ and disturbances $w_i(t)$ are unknown.

The Lyapunov analysis technique approach of [8],[9] is used here, though there are some complications arising from the fact that $\mu_i(t)$ and the NN weight tuning laws must be implemented as distributed protocols. This entails a careful selection of the Lyapunov function. The Lyapunov function must depend on the grah topology properties. This is closely connected to the selection of global performance indices that depend on the graph topology in Chapter 5. The basis for the selection of suitable graph-dependent Lyapunov functions was laid in the discussion on the graph Laplacian potential in Chapter 7.

The maximum and minimum singular values of a matrix M are denoted $\bar{\sigma}(M)$ and $\underline{\sigma}(M)$ respectively. The Frobenius norm is $\|M\|_F = \sqrt{tr\{M^T M\}}$ with $tr\{\cdot\}$ the trace. The Frobenius inner product of two matrices is $\langle M_1, M_2 \rangle_F = \sqrt{tr\{M_1^T M_2\}}$.

The following Fact gives two standard results used in neural adaptive control [8].

Fact 9.1 *Let the nonlinearities $f(x)$ in (9.2) be smooth on a compact set $\Omega \subset R^N$. Then:*

a. The NN estimation error $\varepsilon(x)$ is bounded by $\|\varepsilon\| \le \varepsilon_M$ on Ω, with ε_M a fixed bound [5],[8].

b. Weierstrass higher-order approximation theorem.   Select the activation functions $\varphi(x)$ as a complete independent basis (e.g. polynomials). Then the NN estimation error $\varepsilon(x)$ converges uniformly to zero on Ω as $\eta_i \to \infty, i = 1, N$. That is $\forall \xi > 0$ there exist $\bar{\eta}_i, i = 1, N$ such that $\eta_i > \bar{\eta}_i, \forall i$ implies $\sup_{x \in \Omega} \|\varepsilon(x)\| < \xi$ [11].

The following standard assumptions are required.   Although the bounds mentioned are assumed to exist, they are not used in the design and do not have to be known. They appear in the error bounds in the proof of Theorem 9.1.

Assumption 9.1

    a.   *The unknown disturbance w_i is bounded for all i . Thus the overall disturbance vector w is also bounded by $\|w\| \le w_M$ with w_M a fixed bound.*

    b.   *Unknown ideal NN weight matrix W is bounded by $\|W\|_F \le W_M$.*

    c.   *NN activation functions φ_i are bounded $\forall i$, so that one can write for the overall network that $\|\varphi\| \le \phi_M$.*

    d.   *The unknown consensus variable dynamics $f_0(x_0, t)$ are bounded so that $\|f_0(x_0, t)\| \le F_M, \forall t$ respectively.*

    e.   *The target trajectory is bounded so that $\|x_0^1(t)\| < X_0^1, \|x_0^2(t)\| < X_0^2, \forall t$.*

The definition for robust practical stability, or uniform ultimate boundedness, is standard and the following definitions extend it to multi-agent systems.

Definition 9.2 *Any vector time function $y(t)$ is said to be uniformly ultimately bounded (UUB) [8] if there exist a compact set $\Omega \subset R^N$ so that $\forall y(t_0) \in \Omega$ there exist a bound B_m and a time $t_f(B_m, y(t_0))$ such that $\|x(t) - y(t)\| \le B_m \ \forall t \ge t_0 + t_f$.*

Definition 9.3 *The control node state $x_0(t)$ is said to be cooperative uniformly ultimately bounded (CUUB) if there exist a compact set $\Omega \subset R$ so that $\forall (x_i(t_0) - x_0(t_0)) \in \Omega$ there exist a bound B_m and a time $t_f(B_m, x(t_0), x_0(t_0))$ such that $\|x_i(t) - x_0(t)\| \le B_m \ \forall i, \forall t \ge t_0 + t_f$.*

The next key constructive result is needed. An $M-$ matrix is a square matrix having nonpositive off-diagonal elements and all principal minors nonnegative

(see Chapter 2 and [10]). The Laplacian matrix L is irreducible if and only if the digraph is strongly connected. That is, L is not cogredient to a lower triangular matrix, i.e., there is no permutation matrix U such that

$$L = U \begin{bmatrix} * & 0 \\ * & * \end{bmatrix} U^T$$

Lemma 9.3 [10] *Let L be irreducible and B have at least one diagonal entry $b_i > 0$. Then $(L + B)$ is a nonsingular $M - matrix$. Define*

$$q = \begin{bmatrix} q_1 & q_2 & \cdots & q_N \end{bmatrix}^T = (L + B)^{-1} \underline{1} \qquad (9.24)$$

$$P = \mathrm{diag}\{p_i\} \equiv \mathrm{diag}\{1/q_i\} \qquad (9.25)$$

Then $P > 0$ and the symmetric matrix Q defined as

$$Q = P(L + B) + (L + B)^T P \qquad (9.26)$$

is positive definite.

In fact, this result holds if the digraph has a spanning tree and the pinning gain b_i is nonzero for at leat one root node i.

This result provides the main tool for selecting a Lyapunov function for cooperative control such that the resulting control protocols and the adaptive tuning laws are both distributed in the sense that they respect the graph topology and use only local information available from each agent and from its neighbors. This result extends the discusion in Chapter 7 about Lyapunov functions for cooperative control on graphs to the case of pinning control, that is, to the cooperative tracker problem. It will further be seen in the upcoming proof that it is important that the matrix P is diagonal.

The main result of the chapter is now presented.

Theorem 9.1 Distributed Adaptive Control Protocol for Synchronization [2].
Consider the networked systems given by (9.2) under the Assumption 1. Let the communication digraph be strongly connected. Select the auxiliary control signal $\mu_i(t)$ in(9.21) so that the local node control protocols are given by

$$u_i = cr_i - \hat{f}_i(x_i) + \frac{\lambda_i}{d_i + b_i} e_i^2 \qquad (9.27)$$

with $\lambda_i = \lambda > 0 \; \forall i$, control gains $c > 0$, and $r_i(t)$ defined in (9.13). Then

$$u = cr - \hat{W}^T \varphi(x) + \lambda (D + B)^{-1} e^2 \qquad (9.28)$$

Select the local node NN tuning laws

$$\dot{\hat{W}}_i = -F_i \varphi_i r_i^T p_i (d_i + b_i) - \kappa F_i \hat{W}_i \qquad (9.29)$$

with $F_i = \Pi_i I_{\eta_i}$, I_{η_i} *the* $\eta_i \times \eta_i$ *identity matrix,* $\Pi_i > 0$ *and* $\kappa > 0$ *scalar tuning gains, and* $p_i > 0$ *defined in Lemma 9.3. Select*

$$\lambda = \sqrt{\frac{\underline{\sigma}(D+B)}{\overline{\sigma}(P)\overline{\sigma}(A)}} \qquad (9.30)$$

and select the control gain c and NN tuning gain κ *so that*

$$
\begin{aligned}
&i. \quad c = \frac{2}{\underline{\sigma}(Q)}\left(\frac{1}{\sqrt{\lambda}} + \lambda\right) > 0 \\
&ii. \quad \frac{1}{2}\varphi_m \overline{\sigma}(P)\overline{\sigma}(A) \le \kappa \le \lambda - 1
\end{aligned}
\qquad (9.31)
$$

with $P > 0, Q > 0$ *define in Lemma 9.3 and* A *the graph adjacency matrix.*

Then there exist numbers of neurons $\overline{\eta}_i, i = 1, N$ *such that for* $\eta_i > \overline{\eta}_i, \forall i$ *the overall sliding mode cooperative error vector* $r(t)$, *the local cooperative error vectors* $e^1(t), e^2(t)$ *and the NN weight estimation errors* $\tilde{W}$ *are UUB, with practical bounds given by (9.52)-(9.54) respectively. Moreover the consensus variable* $x_0(t) = \begin{bmatrix} x_0^1 & x_0^2 \end{bmatrix}^T$ *is cooperative UUB and all nodes synchronize so that* $\left\| x_i^1(t) - x_0^1(t) \right\| \to 0$, $\left\| x_i^2(t) - x_0^2(t) \right\| \to 0$. *Moreover, the bounds (9.52)-(9.54) can be made small by manipulating the NN and control gain parameters.*

Proof. The proof has two parts.

Part A: We claim that for any fixed $\varepsilon_M > 0$, there exist numbers of neurons $\overline{\eta}_i, i = 1, N$ such that for $\eta_i > \overline{\eta}_i, \forall i$ the NN approximation error is bounded by $\|\varepsilon\| \le \varepsilon_M$. The claim is proven in Part b of the proof. Consider now the Lyapunov function candidate

$$V = \frac{1}{2} r^T P r + \frac{1}{2} \tilde{W}^T F^{-1} \tilde{W} + \frac{1}{2} \left(e^1\right)^T e^1 \qquad (9.32)$$

with $P = P^T > 0$ defined as in Lemma 9.3 and a constant matrix $F^{-1} = F^{-T} > 0$. Then

$$\dot{V} = r^T P \dot{r} + tr\left\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\right\} + \left(e^1\right)^T \dot{e}^1 \qquad (9.33)$$

Using (9.23) and (9.28)

$$\dot{r} = -(L+B)\left(\tilde{f}(x)+cr+w\right)+(L+B)\underline{1}f_0(x_0,t)+A(D+B)^{-1}\Lambda e^2 \quad (9.34)$$

Therefore

$$\dot{V} = -r^T P(L+B)\left(\tilde{W}^T\varphi(x)+\varepsilon+w+cr\right)+r^T P(L+B)\underline{1}f_0(x_0,t)+$$
$$r^T PA(D+B)^{-1}\Lambda e^2 +tr\left\{\tilde{W}^T F^{-1}\dot{\tilde{W}}\right\}+\left(e^1\right)^T e^2 \qquad (9.35)$$

$$\dot{V} = -r^T P(L+B)cr - r^T P(L+B)\tilde{W}^T\varphi(x) - r^T P(L+B)(\varepsilon+w)+$$
$$r^T P(L+B)\underline{1}f_0(x_0,t)+r^T PA(D+B)^{-1}\Lambda e^2 +tr\left\{\tilde{W}^T F^{-1}\dot{\tilde{W}}\right\}+\left(e^1\right)^T\left(r-\Lambda e^1\right) \qquad (9.36)$$

$$\dot{V} = -cr^T P(L+B)r - r^T P(L+B)\left\{\varepsilon+w-\underline{1}f_0(x_0,t)\right\}+$$
$$r^T PA(D+B)^{-1}\Lambda e^2 +tr\left[\tilde{W}^T\left(F^{-1}\dot{\tilde{W}}-\varphi(x)r^T P(L+B)\right)\right]+\left(e^1\right)^T r-\left(e^1\right)^T\Lambda e^1 \qquad (9.37)$$

$$\dot{V} = -cr^T P(L+B)r - r^T P(L+B)\left\{\varepsilon+w-\underline{1}f_0(x_0,t)\right\}+r^T PA(D+B)^{-1}\Lambda\left(r-\Lambda e^1\right)$$
$$+tr\left[\tilde{W}^T\left(F^{-1}\dot{\tilde{W}}-\varphi(x)r^T P(L+B)\right)\right]+\left(e^1\right)^T r-\left(e^1\right)^T\Lambda e^1 \qquad (9.38)$$

$$\dot{V} = -cr^T P(L+B)r - r^T P(L+B)\left\{\varepsilon+w-\underline{1}f_0(x_0,t)\right\}+$$
$$tr\left[\tilde{W}^T\left(F^{-1}\dot{\tilde{W}}-\varphi(x)r^T P(D+B)\right)\right]+tr\left[\tilde{W}^T\varphi(x)r^T PA\right]+ \qquad (9.39)$$
$$r^T PA(D+B)^{-1}\Lambda r - r^T PA(D+B)^{-1}\Lambda^2 e^1 +\left(e^1\right)^T r-\left(e^1\right)^T\Lambda e^1$$

Since L is irreducible and B has at least one diagonal entry $b_i > 0$, then $(L+B)$ is a nonsingular $M-$matrix. Thus, according to Lemma 9.3, one can write

$$\dot{V} = -\frac{1}{2}cr^T Q r - r^T P(L+B)\{\varepsilon + w - \underline{1}f_0(x_0,t)\} + tr\left[\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} - \varphi(x)r^T P(D+B)\right)\right] +$$

$$tr\left[\tilde{W}^T\varphi(x)r^T PA\right] + r^T PA(D+B)^{-1}\Lambda r - r^T PA(D+B)^{-1}\Lambda^2 e^1 + \left(e^1\right)^T r - \left(e^1\right)^T \Lambda e^1$$

$$(9.40)$$

with $Q = Q^T > 0$.

Adopt the NN weight tuning law $\dot{\hat{W}}_i = F_i\varphi_i r_i^T p_i(d_i + b_i) + \kappa F_i\hat{W}_i$. Taking norms on both sides in (9.40) one has

$$\dot{V} \le -\frac{1}{2}c\underline{\sigma}(Q)\|r\|^2 + \bar{\sigma}(P)\bar{\sigma}(L+B)B_M\|r\| + \kappa W_M\left\|\tilde{W}\right\|_F$$

$$-\kappa\left\|\tilde{W}\right\|_F^2 + \phi_M\bar{\sigma}(P)\bar{\sigma}(A)\left\|\tilde{W}\right\|_F\|r\| + \qquad (9.41)$$

$$\frac{\bar{\sigma}(P)\bar{\sigma}(A)\bar{\sigma}(\Lambda)}{\underline{\sigma}(D+B)}\|r\|^2 + \left(1 + \frac{\bar{\sigma}(P)\bar{\sigma}(A)\bar{\sigma}(\Lambda^2)}{\underline{\sigma}(D+B)}\right)\|r\|\|e^1\| - \underline{\sigma}(\Lambda)\|e^1\|^2$$

where $B_M = (\varepsilon_M + w_M + F_M)$. Then

$$\dot{V} \le -\begin{bmatrix}\|e^1\| & \|r\| & \|\tilde{W}\|_F\end{bmatrix}\begin{bmatrix} \underline{\sigma}(\Lambda) & \frac{1}{2}\left(1 + \frac{\bar{\sigma}(P)\bar{\sigma}(A)\bar{\sigma}(\Lambda^2)}{\underline{\sigma}(D+B)}\right) \\ \frac{1}{2}\left(1 + \frac{\bar{\sigma}(P)\bar{\sigma}(A)\bar{\sigma}(\Lambda^2)}{\underline{\sigma}(D+B)}\right) & \left(\frac{1}{2}c\underline{\sigma}(Q) - \frac{\bar{\sigma}(P)\bar{\sigma}(A)\bar{\sigma}(\Lambda)}{\underline{\sigma}(D+B)}\right) \\ 0 & \frac{1}{2}\phi_M\bar{\sigma}(P)\bar{\sigma}(A) \end{bmatrix} \cdots$$

$$\cdots \begin{matrix} 0 \\ \frac{1}{2}\phi_M\bar{\sigma}(P)\bar{\sigma}(A) \\ \kappa \end{matrix}\begin{bmatrix}\|e^1\| \\ \|r\| \\ \|\tilde{W}\|_F\end{bmatrix} + \begin{bmatrix}0 & \bar{\sigma}(P)\bar{\sigma}(L+B)B_M & \kappa W_M\end{bmatrix}\begin{bmatrix}\|e^1\| \\ \|r\| \\ \|\tilde{W}\|_F\end{bmatrix}$$

$$(9.42)$$

Write this as

$$\dot{V} \le -z^T H z + h^T z \qquad (9.43)$$

Clearly $\dot{V} \le 0$ iff $H \ge 0$ and

$$\|z\| > \frac{\|h\|}{\underline{\sigma}(H)} \qquad (9.44)$$

According to (9.32) this defines a level set of $V(z)$, so it is direct to show that $\dot{V} \le 0$ for V large enough such that (9.44) holds [6]. To show this, according to (9.32) one has

$$\tfrac{1}{2}\underline{\sigma}(P)\|e\|^2 + \frac{1}{2\Pi_{\max}}\|\tilde{W}\|_F^2 + \tfrac{1}{2}\|e^1\|^2 \le V \le \tfrac{1}{2}\bar{\sigma}(P)\|e\|^2 + \frac{1}{2\Pi_{\min}}\|\tilde{W}\|_F^2 + \tfrac{1}{2}\|e^1\|^2$$

$$(9.45)$$

$$\frac{1}{2}\begin{bmatrix}\|e^1\| & \|r\| & \|\tilde{W}\|_F\end{bmatrix}\underbrace{\begin{bmatrix}\underline{\sigma}(P) & & \\ & \frac{1}{\Pi_{\max}} & \\ & & 1\end{bmatrix}}_{S_1}\begin{bmatrix}\|e^1\| \\ \|r\| \\ \|\tilde{W}\|_F\end{bmatrix} \le V \le$$

$$\frac{1}{2}\begin{bmatrix}\|e^1\| & \|r\| & \|\tilde{W}\|_F\end{bmatrix}\underbrace{\begin{bmatrix}\bar{\sigma}(P) & & \\ & \frac{1}{\Pi_{\min}} & \\ & & 1\end{bmatrix}}_{S_2}\begin{bmatrix}\|e^1\| \\ \|r\| \\ \|\tilde{W}\|_F\end{bmatrix} \qquad (9.46)$$

with $\Pi_{\min}, \Pi_{\max}$ the minimum and maximum values of Π_i. Equation (9.46) is equivalent to

$$\tfrac{1}{2}z^T \underline{S}z \le V \le \tfrac{1}{2}z^T \bar{S}z \qquad (9.47)$$

where $\underline{S} = \underline{\sigma}(S_1)$ and $\bar{S} = \bar{\sigma}(S_2)$. Then

$$\tfrac{1}{2}\underline{\sigma}(\underline{S})\|z\|^2 \le V \le \tfrac{1}{2}\bar{\sigma}(\bar{S})\|z\|^2 \qquad (9.48)$$

Therefore,

$$V > \frac{1}{2}\frac{\bar{\sigma}(\bar{S})\|h\|^2}{\underline{\sigma}^2(H)} \qquad (9.49)$$

implies (9.44).

For a symmetric positive definite matrix, the singular values are equal to its eigenvalues. Define

$$\Lambda = \lambda I, \lambda = \sqrt{\frac{\underline{\sigma}(D+B)}{\bar{\sigma}(P)\bar{\sigma}(A)}} \ , \ c = \frac{2}{\underline{\sigma}(Q)}\left(\frac{1}{\sqrt{\lambda}}+\lambda\right) \text{ and } \gamma = \frac{1}{2}\varphi_m\bar{\sigma}(P)\bar{\sigma}(A). \text{ Then}$$

(9.42) can be written as

$$\dot{V} \leq -\begin{bmatrix} \|e^1\| & \|r\| & \|\tilde{W}\|_F \end{bmatrix} \underbrace{\begin{bmatrix} \lambda & 1 & 0 \\ 1 & \lambda & \gamma \\ 0 & \gamma & \kappa \end{bmatrix}}_{H} \begin{bmatrix} \|e^1\| \\ \|r\| \\ \|\tilde{W}\|_F \end{bmatrix}$$

$$+\begin{bmatrix} 0 & \bar{\sigma}(P)\bar{\sigma}(L+B)B_M & \kappa W_M \end{bmatrix} \begin{bmatrix} \|e^1\| \\ \|r\| \\ \|\tilde{W}\|_F \end{bmatrix} \tag{9.50}$$

The transformed H matrix is symmetric and positive definite under assumption given by (9.31). Therefore from Gershgorin circle's theorem

$$\underline{\sigma}(H) \geq \kappa - \gamma \tag{9.51}$$

with $0 < \gamma \leq \kappa \leq \lambda - 1$. Therefore $z(t)$ is UUB [6].

In view of the fact that, for any vector z, one has $\|z\|_1 \geq \|z\|_2 \geq \cdots \geq \|z\|_\infty$, sufficient conditions for (9.44) are:

$$\|r\| > \frac{B_M\bar{\sigma}(P)\bar{\sigma}(L+B)+\kappa W_M}{\kappa - \frac{1}{2}\varphi_m\bar{\sigma}(P)\bar{\sigma}(A)} \tag{9.52}$$

or

$$\|e^1\| > \frac{B_M\bar{\sigma}(P)\bar{\sigma}(L+B)+\kappa W_M}{\kappa - \frac{1}{2}\varphi_m\bar{\sigma}(P)\bar{\sigma}(A)} \tag{9.53}$$

or

$$\|\tilde{W}\| > \frac{B_M\bar{\sigma}(P)\bar{\sigma}(L+B)+\kappa W_M}{\kappa - \frac{1}{2}\varphi_m\bar{\sigma}(P)\bar{\sigma}(A)} \tag{9.54}$$

Note that this shows UUB of $r(t), e_1(t), \tilde{W}(t)$. Therefore from Lemma 9.2, the boundedness of r and e^1 implies bounded e^2. Now Lemma 9.1 shows that the consensus error vector $\delta(t)$ is UUB. Then $x_0(t)$ is cooperative UUB.

Part B: See [4].

According to (9.43) $\dot{V} \le -\underline{\sigma}(H)\|z\|^2 + \|h\|\|z\|$ and according to (9.48)

$$\dot{V} \le -\alpha V + \beta\sqrt{V} \tag{9.55}$$

with $\alpha \equiv 2\underline{\sigma}(H)/\bar{\sigma}(\bar{S})$, $\beta \equiv \sqrt{2}\|h\|/\sqrt{\underline{\sigma}(\underline{S})}$. Thence

$$\sqrt{V(t)} \le \sqrt{V(0)}e^{-\alpha t/2} + \tfrac{\beta}{\alpha}(1 - e^{-\alpha t/2}) \le \sqrt{V(0)} + \tfrac{\beta}{\alpha} \tag{9.56}$$

Using (9.48) one has

$$\|e^1(t)\| \le \|z(t)\| \le \sqrt{\frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})}} \sqrt{\|e(0)\|^2 + \|\tilde{W}(0)\|_F^2 + \|e^1(0)\|^2} + \frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})}\frac{\|h\|}{\underline{\sigma}(H)} \equiv \rho$$

and

$$\|r(t)\| \le \|z(t)\| \le \sqrt{\frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})}} \sqrt{\|e(0)\|^2 + \|\tilde{W}(0)\|_F^2 + \|e^1(0)\|^2} + \frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})}\frac{\|h\|}{\underline{\sigma}(H)} \equiv \rho$$

$$\tag{9.57}$$

Then from (9.7)

$$\|x^1(t)\| \le \frac{1}{\underline{\sigma}(L+B)}\|e^1(t)\| + \sqrt{N}\|x_0^1(t)\| \tag{9.58}$$

$$\|x^1(t)\| \le \frac{\rho}{\underline{\sigma}(L+B)} + \sqrt{N}X_0^1 \equiv h_0^1 \tag{9.59}$$

Similarly from (9.57) and using (9.15) in Lemma 9.2

$$\|e^2(t)\| \le \lambda\|e^1(t)\| + \|r(t)\| \tag{9.60}$$
$$\le \rho(1+\lambda)$$

This implies

$$\|x^2(t)\| \le \frac{\rho(1+\lambda)}{\underline{\sigma}(L+B)} + \sqrt{N}X_0^2 \equiv h_0^2 \tag{9.61}$$

where $X_0^2 = \|x_0^2(t)\|$ and $\|h\| \le B_M\bar{\sigma}(P)\bar{\sigma}(L+B) + \kappa W_M$. Therefore, the state is contained for all times $t \ge 0$ in a compact set $\Omega_0 = \{x(t) \mid \|x^1(t)\| < h_0^1, \|x^2(t)\| < h_0^2\}$. According to the Weierstrass approximation theorem (Fact 9.1), given any NN approximation error bound ε_M

there exist numbers of neurons $\bar{\eta}_i, i = 1, N$ such that $\eta_i > \bar{\eta}_i, \forall i$ implies $\sup_{x \in \Omega} \|\varepsilon(x)\| < \varepsilon_M$. ∎

Discussion on NN Adaptive Control Proofs for Cooperative Systems.

If any one of (9.52), (9.53) or (9.54) holds, the Lyapunov derivative is negative and V decreases. Therefore, these provide practical bounds for the neighborhood synchronization error and the NN weight estimation error.

The elements p_i of the positive definite matrix $P = diag\{p_i\}$ required in the NN tuning law (9.29) are computed as $P^{-1}\underline{1} = (L+B)^{-1}\underline{1}$ (see Lemma 9.3), which requires global knowledge of the graph structure unavailable to individual nodes. However, due to the presence of the arbitrary diagonal gain matrix $F_i > 0$ in (9.29), one can choose $p_i F_i > 0$ arbitrary without loss of generality.

It is important to select the Lyapunov function candidate V in (9.32) as a function of locally available variables, e.g. the local sliding mode error $r(t)$ and cooperative neighborhood error $e_1(t)$ in (9.14) and (9.7) respectively. This means that any local auxiliary control signals $\mu_i(t)$ and NN tuning laws developed in the proof are distributed and hence implementable at each node. The use of the Frobenius norm in the Lyapunov function is also instrumental, since it gives rise to Frobenius inner products in the proof that only depend on trace terms, where only the diagonal terms are important. In fact, the Frobenius norm is ideally suited for the design of distributed protocols. Finally, it is important in developing distributed control protocols that the matrix P of Lemma 9.3 is diagonal.

9.3 Simulation Example

For this set of simulations, consider the 5-node strongly connected digraph structure in Fig. 9.1 with a leader node connected to node 3. The edge weights and the pinning gain in (9.4) were taken equal to 1.

Consider the node dynamics for node i given by the second-order Lagrange form dynamics

$$\begin{aligned} \dot{q}_{1_i} &= q_{2_i} \\ \dot{q}_{2_i} &= J_i^{-1}\left[u_i - B_i^r q_{2_i} - M_i g l_i \sin(q_{1_i}) \right] \end{aligned} \qquad (9.62)$$

where $q_i = \begin{bmatrix} q_{1_i}, q_{2_i} \end{bmatrix}^T \in R^2$ is the state vector, J_i is the total inertia of the link and the motor, B_i^r is overall damping coefficient, M_i is total mass, g is

gravitational acceleration and l_i is the distance from the joint axis to the link center of mass for node. J_i, B_i^r, M_i and l_i are considered unknown and may be different for each node.
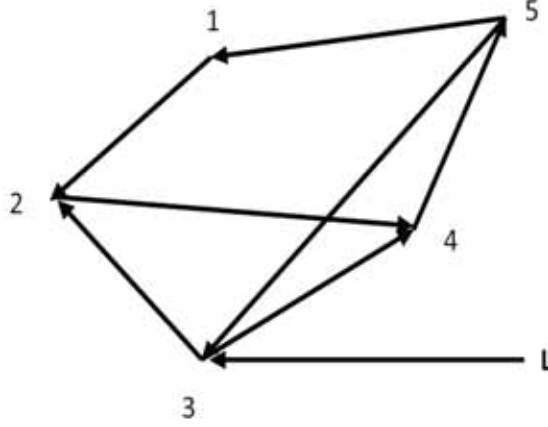

Fig. 9.1 Five-node digraph with one leader node

The desired target node dynamics is taken as the inertial system

$$m_0 \ddot{q}_0 + d_0 \dot{q}_0 + k_o q_0 = u_0 \qquad (9.63)$$

Select the feedback linearization input

$$u_0 = -\left[K_1 \left(q_0 - \sin(\beta t) \right) + K_2 \left(\dot{q}_0 - \beta \cos(\beta t) \right) \right] + d_0 \dot{q}_0 + k_0 q_0 + \beta^2 m_0 \sin(\beta t)$$

$$(9.64)$$

for a constant $\beta > 0$. Then, the target motion $q_0(t)$ tracks the desired reference trajectory $\sin(\beta t)$.

The cooperative adaptive control protocols of Theorem 9.1 were implemented at each node. As is standard in neural and adaptive control systems, reasonable positive values were initially chosen for all control and NN tuning parameters. Generally, the simulation results are not too dependent on the specific choice of parameters as long as they are selected as detailed in the Theorem 9.1 and assumptions, e.g. positive. The number of NN hidden layer units can be fairly small with good performance resulting, normally in the range of 5-10. As in most control systems, however, the performance can be improved by trying a few simulation runs and adjusting the parameters to obtain good behavior. In on-line implementations, the parameters can be adjusted after some simulation studies to obtain better performance. For NN activation functions we use log-sigmoid of the

form $\dfrac{1}{1+e^{-kt}}$ with positive slope parameter k. The number of neurons used at each node was finally selected as 3. So, $\varphi_m \approx 1$ and $\gamma \approx 0.04$. NN gain parameter was selected as $\kappa = 1.5$. According to (9.31) the pinning gain should be selected large and it was taken as $c = 1000$.

In the simulation plots we show tracking performance of positions (q_{1_i}) and velocities (q_{2_i}) for all i. At steady state all q_{1_i} and q_{2_i} are synchronized and follow the second order single link leader trajectory given by q_0 and $\dot{q}_0$ respectively.
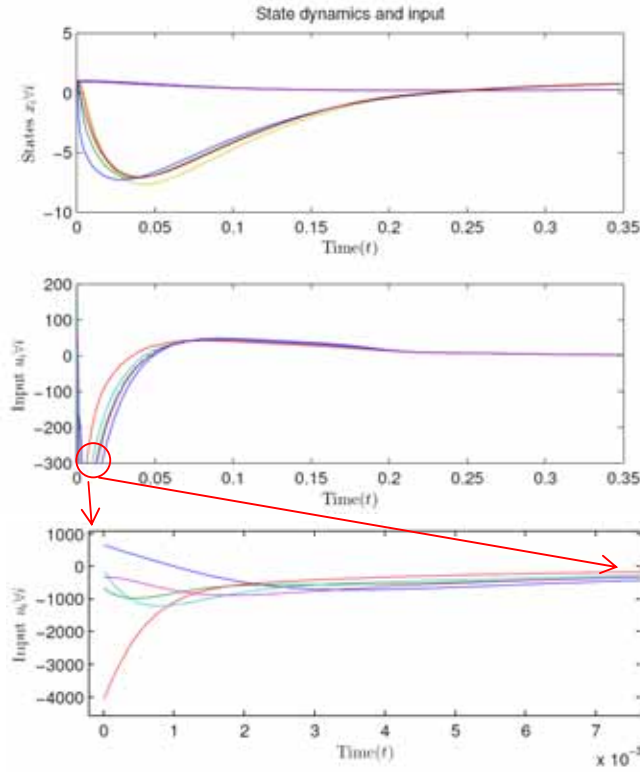
Fig. 9.2 Tracking performance (position and velocity) and control input

Fig. 9.2 shows the tracking performance of the system. One can see from the figure that positions and velocities of all the five nodes are synchronized in two different final values which are given by the final values of $\left[q_0, \dot{q}_0\right]^T$. More

specifically q_{1_i}'s are synchronized with q_0 and q_{2_i}'s are synchronized with $\dot{q}_0$ at steady state. The figure also shows the inputs u_i for all agents while tracking.

Fig. 9.3 describes the position and velocity consensus disagreement error vectors namely $\left(q_{1_i} - q_0, q_{2_i} - \dot{q}_0\right)^T, \forall i$ and also the NN estimation error in terms of $\left[f_i(x) - \hat{f}_i(x)\right]^T \forall i$. One can see that all the errors are minimized almost to zero.
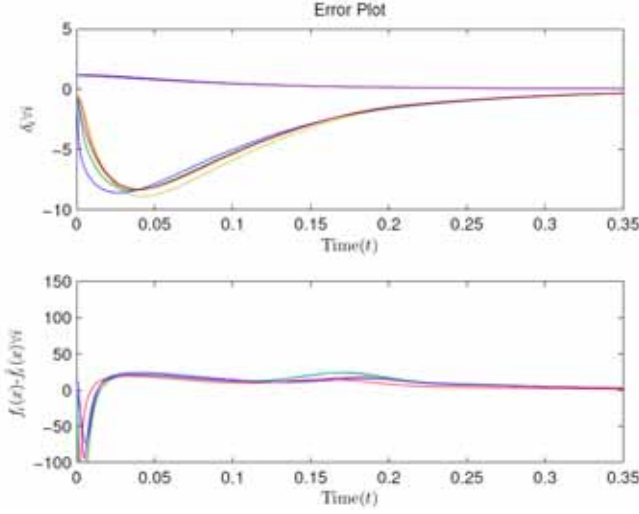

Fig. 9.3 Disagreement vector and NN estimation error vector

Fig. 9.4 describes the comparison of unknown dynamics $f_i(x)$ with estimated dynamics $\hat{f}_i(x)$. The figure also shows that the steady state values of $f_i(x)$ and $\hat{f}_i(x)$ are almost equal. Fig. 9.5 describes the dynamics of all the NN weights over time which also describes the effectiveness of the tuning law. As stated in Chapter 8, please note that the simulation response can be varied for the choice of number of neurons. In this example the number of neurons = 3 for each node.
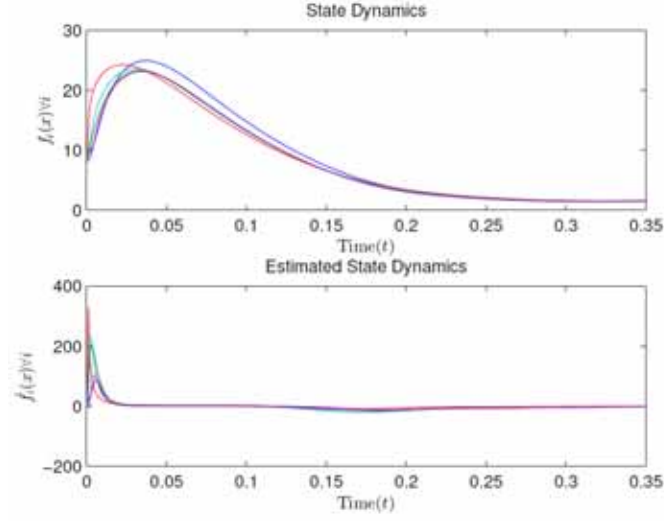
Fig. 9.4 Comparison of $f(x)$ and $\hat{f}(x)$

Fig. 9.6 is the phase plane plot of all agents, i.e. the plot of $q_{1_i} \forall i$ (along the x-axis) and $q_{2_i} \forall i$ (along the $y-$axis). At steady state the Lissajous pattern formed by all five nodes is the target node's phase plane trajectory.
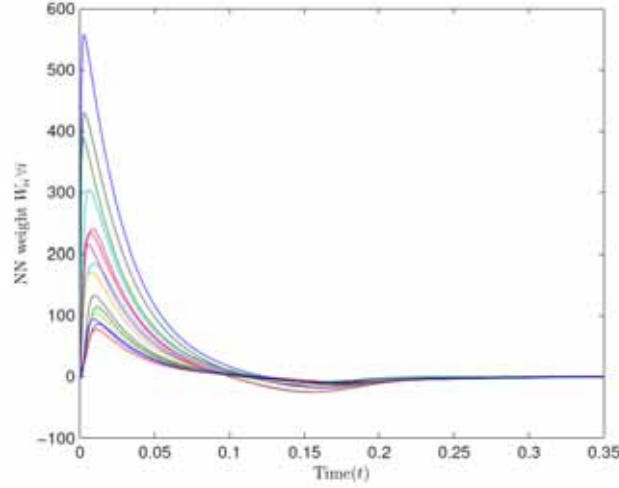

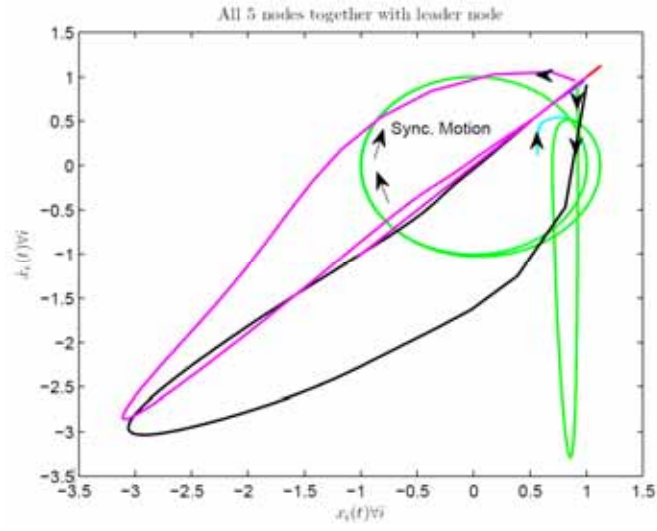**Fig. 9.5** Stable NN weight dynamics for all nodes

Fig. 9.6 Phase plane plot for all the nodes

References

[1]   A. Das and F. L. Lewis, "Distributed Adaptive Control for Synchronization of Unknown Nonlinear Networked Systems," *Automatica*, vol. 46, no. 12, pp. 2014–2021, Aug. 2010.

[2]   A. Das and F. L. Lewis, "Cooperative adaptive control for synchronization of second-order systems with unknown nonlinearities," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 13, pp. 1509–1524, Sep. 2011.

[3]   S. S. Ge, C. C. Hang, and T. Zhang, *Stable Adaptive Neural Network control*. Berlin: Springer, 1998.

[4]   S. S. Ge and C. Wang, "Adaptive neural control of uncertain MIMO nonlinear systems," *IEEE Trans. Neural Networks*, vol. 15, no. 3, pp. 674–692, May 2004.

[5]   K. Hornik, M. Stinchombe, and H. White, "Multilayer Feedforward Networks are Universal Approximations," *Neural Networks*, vol. 20, pp. 359–366, 1989.

[6]   H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, N.J: Prentice Hall, 2002.

[7]   S. Khoo, L. Xie, and Z. Man, "Robust Finite-Time Consensus Tracking Algorithm for Multirobot Systems," *IEEE Transaction on Mechatronics*, vol. 14, no. 2, pp. 219–228.

[8]   F. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London: Taylor and Francis, 1999.

[9]   F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 7, no. 2, pp. 388–399, 1996.

[10]   Z. Qu, *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. New York: Springer-Verlag, 2009.

[11]   M. H. Stone, "The Generalized Weierstrass Approximation Theorem," *Mathematics Magazine*, vol. 21, no. 4,5, pp. 167–184,237–254, 1948.

10

Cooperative Adaptive Control for Higher-Order Nonlinear Systems

Cooperative control on communication graphs for agents that have unknown nonlinear dynamics that are not the same is a challenge. The interaction of the communication graph topology with the agent's system dynamics is not easy to investigate if the dynamics of the agents are heterogeneous, that is, not identical, since the Kronecker product cannot be used to simplify the analysis. Therefore, the intertwining of the graph structure with the local control design is more severe and makes the design of guaranteed synchronizing controls very difficult. That is to say, the communication graph structure imposes more severe limitations on the design of controllers for systems that have nonlinear and non-identical dynamics, making it more challenging to guarantee synchronization of all agents in the network.

In this chapter, we continue to study adaptive control for the cooperative tracking problem, where it is desired for all agents to synchronize to the state of a target or command generator node. For implementation using local neighborhood information on a communication graph, it is required not only for the control law to be distributed, but also for the adaptive parameter tuning laws to be distributed. We have seen in Chapters 8 and 9 how to design adaptive controllers for multi-agent systems with first-order [2] and second-order [3] nonlinear dynamics on general directed graphs. However, many practical systems have higher-order dynamics. Therefore, in this chapter we study synchronizing controls design for nonlinear systems in chained or Brunovsky form, where all agents have unknown dynamics that may not be the same for each agent.

The extension of adaptive control to higher-order dynamics in this chapter is not straightforward. In cooperative adaptive controller for multi-agent systems on graphs, it is required not only for the control law to be distributed, but also for the adaptive tuning laws to be distributed. This requires the careful crafting of a suitable Lyapunov function that automatically yields a distributed adaptive controller depending only on local neighborhood information. In Chapter 7 were given some techniques for Lyapunov design for cooperative control. It was seen that for undirected graphs, and even balanced graphs, it is not difficult to extend Lyapunov

techniques to multi-agent systems on graphs. However, for the case of general directed graphs it was seen that the Lyapunov function must be carefully constructed, and must generally contain information about the graph topology, e.g. in the form of the elements of the first left eigenvector of the graph Laplacian matrix.

We show in this chapter how to craft a Lyapunov function suitable for adaptive control on graphs of higher-order multi-agent systems with heterogeneous nonlinear dynamics. We use two Lyapunov equations, one that captures the local system dynamics of each agent, and one that captures the graph topology. The two Lyapunov equations are solved independently, so that the local controls design is decoupled from the detailed communication graph topology. Yet the interplay of these two design equations in the Lyapunov proof yields cooperative adaptive control laws where both the control protocols and the tuning laws depend only on distributed neighbor information.

In Section 10.1 we formulate the adaptive cooperative tracker problem for higher-order multi-agent systems on graphs. An error dynamics is derived and a Lyapunov equation is defined that depends on the graph topology. In Section 10.2 a distributed control protocol is developed based on a suitable sliding mode error. A second Lyapunov equation is introduced to capture the stability properties of this sliding error dynamics. Local function approximator structures are introduced to estimate the unknown nonlinear dynamics. In Section 10.3 we develop cooperative control adaptive tuning laws that only depend on distributed information locally available in the graph. The Lyapunov function we use to derive the tuning laws involves both the Lyapunov equations- one depending on the graph topology and one on the sliding error dynamics. A simulation example in Section 10.4 verifies proper performance of the cooperative adaptive controller.

The results of this chapter were first published in [27].

10.1 Sliding Variable Control Formulation and Error Dynamics

In this section we lay the foundation for design of adaptive cooperative controllers for a class of nonlinear agent dynamics in Brunovsky form. Basic definitions are given and the local neighborhood error dynamics are introduced. Some requirements on the graph topology are laid out, and a Lyapunov equation is given that contains information about the graph topological structure.

10.1.1 Synchronization for Nonlinear Higher-Order Cooperative Systems

We consider the case of multi-agent control on a communication graph where each agent has higher-order unknown nonlinear dynamics in Brunovsky form. The agent dynamics need not be identical. This class of systems is important for several reasons. First, most existing works on multi-agent systems study the first- or second-order systems with linear dynamics. However, in engineering, many systems are modeled by higher-order dynamics. For example, a single link, flexible joint manipulator is well modeled by a fourth-order nonlinear system [10]. The jerk (i.e. derivative of acceleration) systems, described by third-order differential equations, are of particular interest in mechanical engineering. An interesting property of nonlinear jerk systems is its possible chaotic behavior. Due to the challenges in designing cooperative controls for systems distributed on communication networks, it is not straightforward to extend results for first- and second-order systems to that with higher-order dynamics, and a great deal of study has been devoted to higher-order linear multi-agent systems ([9],[20],[24],[28], etc.).

The communication network considered in this chapter is a directed graph with a fixed topology, denoted by $\mathcal{G} = (\mathcal{V}, E)$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is a nonempty set of nodes/agents and $E \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges/arcs. Its associated adjacency matrix is $A = [a_{ij}] \in \mathbb{R}^{N \times N}$, where $a_{ij} > 0$ if $(v_j, v_i) \in E$, i.e. there is an edge from node j to node i; otherwise $a_{ij} = 0$. Define $d_i = \sum_{j=1}^{N} a_{ij}$ as the weighted in-degree of node i and $D = diag\{d_i\} \in \mathbb{R}^{N \times N}$ as the in-degree matrix. The graph Laplacian matrix is $L = [l_{ij}] = D - A$. Let $\underline{1} = [1, \ldots, 1]^T$ with appropriate dimension, which satisfies $L\underline{1} = 0$. Define $\mathcal{N} = \{1, \ldots, N\}$.

Consider a group of N ($N \geq 2$) agents with non-identical dynamics distributed on a directed communication network $\mathcal{G}$. The dynamics of the i-th agent is described in the nonlinear Brunovsky form

$$
\begin{aligned}
\dot{x}_{i,1} &= x_{i,2} \\
\dot{x}_{i,2} &= x_{i,3} \\
&\vdots \\
\dot{x}_{i,M} &= f_i(x_i) + u_i + \zeta_i
\end{aligned}
\tag{10.1}
$$

where $x_{i,m} \in \mathbb{R}$ is the m-th state of node i; $x_i = [x_{i,1}, \ldots, x_{i,M}]^T \in \mathbb{R}^M$ is the state vector of node i; $f_i(x_i): \mathbb{R}^M \to \mathbb{R}$ is locally Lipschitz with $f_i(0) = 0$, and it is assumed to be unknown; $u_i \in \mathbb{R}$ is the control input/protocol; $\zeta_i \in \mathbb{R}$ is an external disturb-

ance, which is unknown but bounded. The development can be extended to the vector case $x_{i,m} \in \mathbb{R}^n$ by using the Kronecker product.

Equation (10.1) can be written collectively in global form for all the agents as

$$
\begin{aligned}
\dot{x}^1 &= x^2 \\
\dot{x}^2 &= x^3 \\
&\vdots \\
\dot{x}^M &= f(x) + u + \zeta,
\end{aligned}
$$

(10.2)

where $\quad x^m = [x_{1,m}, \ldots, x_{N,m}]^T \in \mathbb{R}^N \quad ; \quad f(x) = [f_1(x_1), \ldots, f_N(x_N)]^T \in \mathbb{R}^N \quad ;$ $u = [u_1, \ldots, u_N]^T \in \mathbb{R}^N$ and $\zeta = [\zeta_1, \ldots, \zeta_N]^T \in \mathbb{R}^N$. Specifically, for example, when $M = 3$, x^1, x^2 and x^3 represent the global position, velocity and acceleration vectors, respectively.

The time-varying dynamics of a leader or target node, labeled *0*, is described by

$$
\begin{aligned}
\dot{x}_{0,1} &= x_{0,2} \\
\dot{x}_{0,2} &= x_{0,3} \\
&\vdots \\
\dot{x}_{0,M} &= f_0(t, x_0),
\end{aligned}
$$

(10.3)

where $x_{0,m} \in \mathbb{R}$ is the *m*-th state variable and $x_0 = [x_{0,1}, \ldots, x_{0,M}]^T \in \mathbb{R}^M$ is the state vector of the leader node; $f_0(t, x_0) : [0, \infty) \times \mathbb{R}^M \to \mathbb{R}$ is piecewise continuous in *t* and locally Lipschitz in x_0 with $f_0(t, 0) = 0$ for all $t \geq 0$ and all $x_0 \in \mathbb{R}^M$. The leader dynamics $f_0(t, x_0)$ is unknown to all follower nodes (10.1). System (10.3) is assumed to be forward complete, i.e., for every initial condition, the solution $x_0(t)$ exists for all $t \geq 0$.

The leader node dynamics (10.3) can be considered as an exosystem that generates a desired command trajectory. As such, the leader is a command generator.

Define the *m*-th order disagreement variable for node *i*, disagreeing from the leader node 0, as $\delta_{i,m} = x_{i,m} - x_{0,m}$. Then, the global *m*-th order disagreement variable is

$$
\delta^m = x^m - \underline{x}_{0,m},
$$

where $\delta^m = [\delta_{1,m}, \ldots, \delta_{N,m}]^T \in \mathbb{R}^N$ and $\underline{x}_{0,m} = [x_{0,m}, \ldots, x_{0,m}]^T \in \mathbb{R}^N$. The perfect cooperative tracking problem, also known as synchronization problem, is described as follows.

Problem 10.1. Cooperative Tracking Problem.

Design controllers u_i for all the nodes in graph $\mathcal{G}$, such that $x_{i,m}(t) \to x_{0,m}(t)$ as $t \to \infty$ for all $i \in \mathcal{N}$ and $m = 1, \ldots, M$, namely, $\lim_{t \to \infty} \delta^m = 0$ for all $m = 1, \ldots, M$.

Here it should be noted that due to the existence of unknown nonlinearities f_i and disturbances ζ_i, perfect tracking (or synchronization) cannot be achieved. Therefore, the objective is to relax the requirement for the tracking errors δ^m to be within a small neighborhood of zero. This is illustrated by the following definition, which extends the standard concept of uniform ultimate boundedness ([10],[14]) to cooperative control systems with higher-order dynamics.

Definition 10.1. Cooperative Uniform Ultimate Boundedness.

For any $m = 1, \ldots, M$, the tracking error δ^m is said to be cooperatively uniformly ultimately bounded (CUUB) if there exists a compact set $\Omega^m \subset \mathbb{R}^N$ with the property that $\{0\} \subset \Omega^m$, so that for any $\delta^m(t_0) \in \Omega^m$, there exist a bound B^m and a finite time instant $T_m(B^m, \delta^1(t_0), \ldots, \delta^M(t_0))$, both independent of t_0, such that $\left\| \delta^m(t) \right\| \le B^m$, $\forall t \ge t_0 + T_m$.

If the tracking error δ^m is CUUB, then $x_{i,m}(t)$ is bounded within a neighborhood of $x_{0,m}(t)$, for all $i \in \mathcal{N}$ and $t \ge t_0 + T_m$. This describes a practical scenario of "close enough" synchronization.

10.1.2 Local Neighborhood Error Dynamics

In this chapter, it is assumed that only *local distributed state information* on the communication graph is available for controller design. More precisely, for the *i*-th node, the only obtainable information is its local neighborhood synchronization errors [11]

$$e_{i,m} = \sum_{j \in N_i} a_{ij}(x_{j,m} - x_{i,m}) + b_i(x_{0,m} - x_{i,m}), \quad m = 1, \ldots, M, \tag{10.4}$$

where $b_i \ge 0$ is the weight of the edge from the leader node *0* to node *i* ($i \in \mathcal{N}$). Then $b_i > 0$ if and only if there is an edge from the leader node to node *i*, which implies that the leader node can send commands to node *i*. Only a few nodes may be connected to the leader.

Define $e^m = [e_{1,m}, \ldots, e_{N,m}]^T$ and $B = diag\{b_i\} \in \mathbb{R}^{N \times N}$. Noticing that $L\underline{1} = 0$, a straightforward computation of (10.4) gives the global form $e^m = -(L+B)(x^m - \underline{x}_{0,m})$, as detailed below.

$$
e^m = \begin{bmatrix} e_{1,m} \\ e_{2,m} \\ \vdots \\ e_{N,m} \end{bmatrix} = \begin{bmatrix} \sum_{j \in N_1} a_{1j}(x_{j,m} - x_{1,m}) + b_1(x_{0,m} - x_{1,m}) \\ \sum_{j \in N_2} a_{2j}(x_{j,m} - x_{2,m}) + b_2(x_{0,m} - x_{2,m}) \\ \vdots \\ \sum_{j \in N_N} a_{Nj}(x_{j,m} - x_{N,m}) + b_N(x_{0,m} - x_{N,m}) \end{bmatrix}
$$

$$
= \begin{bmatrix} \sum_{j \in N_1} a_{1j} x_{j,m} \\ \sum_{j \in N_2} a_{2j} x_{j,m} \\ \vdots \\ \sum_{j \in N_N} a_{Nj} x_{j,m} \end{bmatrix} - \begin{bmatrix} \sum_{j \in N_1} a_{1j} x_{1,m} \\ \sum_{j \in N_2} a_{2j} x_{2,m} \\ \vdots \\ \sum_{j \in N_N} a_{Nj} x_{N,m} \end{bmatrix} + \begin{bmatrix} b_1 & & & \\ & b_2 & & \\ & & \ddots & \\ & & & b_N \end{bmatrix} (\underline{x}_{0,m} - x^m)
$$

$$
= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} x_{1,m} \\ x_{2,m} \\ \vdots \\ x_{N,m} \end{bmatrix} - \begin{bmatrix} \sum_{j \in N_1} a_{1j} & & & \\ & \sum_{j \in N_2} a_{2j} & & \\ & & \ddots & \\ & & & \sum_{j \in N_N} a_{Nj} \end{bmatrix} \begin{bmatrix} x_{1,m} \\ x_{2,m} \\ \vdots \\ x_{N,m} \end{bmatrix} + B(\underline{x}_{0,m} - x^m)
$$

$$
= Ax^m - Dx^m + B(\underline{x}_{0,m} - x^m) = -Lx^m + B(\underline{x}_{0,m} - x^m) + L\underline{x}_{0,m}
$$

$$
= -(L+B)x^m + (L+B)\underline{x}_{0,m} = -(L+B)(x^m - \underline{x}_{0,m}).
$$

Thus one has the error dynamics in global form

$$
\begin{aligned}
\dot{e}^1 &= e^2 \\
\dot{e}^2 &= e^3 \\
&\vdots \\
\dot{e}^M &= -(L+B)(f(x) + u + \zeta - \underline{f}_0),
\end{aligned}
\tag{10.5}
$$

where $\underline{f}_0 = [f_0(t, x_0), \ldots, f_0(t, x_0)]^T \in \mathbb{R}^N$.

10.1.3 Communication Graph Structure and the Graph Lyapunov Equation

Define the augmented graph as $\bar{\mathcal{G}} = \{\bar{\mathcal{V}}, \bar{E}\}$, where $\bar{\mathcal{V}} = \{v_0, v_1, \ldots, v_N\}$ is the node set and $\bar{E} \subseteq \bar{\mathcal{V}} \times \bar{\mathcal{V}}$ is the edge set. The following assumption holds throughout this chapter.

Assumption 10.1 Communication Graph Topology.

The augmented graph $\bar{\mathcal{G}}$ contains a spanning tree with the root node being the leader node 0.

Assumption 10.1 is a necessary condition for solving the cooperative tracking problem.

The following two technical lemmas play important roles in the proof of Theorem 10.1, the main result of this chapter. In particular, Lemma 10.1 constructs a Lyapunov equation that captures the graph structure, which we call the *graph Lyapunov equation*.

Lemma 10.1. Graph Lyapunov Equation [17].
Define

$$q = [q_1, \ldots, q_N]^T = (L+B)^{-1}\underline{1}, \tag{10.6}$$

$$P = diag\{p_i\} = diag\{1/q_i\}, \tag{10.7}$$

$$Q = P(L+B) + (L+B)^T P. \tag{10.8}$$

Then $P > 0$ and $Q > 0$.

Proof. First we show that matrix $L+B$ is nonsingular. Under Assumption 10.1, at least one node in $\mathcal{G}$ can get information directly from the leader node, i.e. $b_i > 0$ for at least one $i \in \mathcal{N}$. Without loss of generality, it is assumed that there are exactly two nodes r_1 and r_2 such that $b_{r_1} > 0$ and $b_{r_2} > 0$. Then, for matrix $L+B$, $|l_{ii} + b_i| \geq \sum_{j=1, j\neq i}^{N} l_{ij}$ for all $i \in \mathcal{N}$ and the strict inequality holds for $i \in \{r_1, r_2\}$. Assumption 10.1 implies that, for any other node i which does not have direct access to the leader node (i.e. $i \in \mathcal{N} \setminus \{r_1, r_2\}$), there must be a directed path either originated from node r_1 or node r_2. According to [22], $L+B$ is nonsingular. Then

$L + B$ is a nonsingular M-matrix and Lemma 10.1 follows from the same development as in ([17] Theorem 4.25). ∎

It is noted that the reciprocals of the elements of vector $q = [q_1, \ldots, q_N]^T = (L+B)^{-1}\underline{1}$ contain information related to that contained in the elements of the left eigenvector p of the graph Laplacian matrix., i.e. $p^T L = 0$.

Lemma 10.2. *One has*

$$\|\delta^m\| \le \|e^m\| / \underline{\sigma}(L+B), \quad m = 1, \ldots, M. \tag{10.9}$$

where $\underline{\sigma}(L+B)$ is the minimum singular value of the matrix $L+B$.

Proof. $L + B$ is nonsingular under Assumption 10.1. Noticing $e^m = -(L+B)\delta^m$, it follows that $\delta^m = -(L+B)^{-1}e^m$. Therefore

$$\|\delta^m\| = \|(L+B)^{-1}e^m\| \le \|e^m\| / \underline{\sigma}(L+B) . \; i \qquad ∎$$

Remark 10.1. Lemma 10.2 implies that, if the graph has a spanning tree, then as long as $\|e^m\|$ is bounded, $\|\delta^m\|$ is bounded. Moreover, $e^m \to 0$ implies $\delta^m \to 0$, i.e. $x_{i,m} \to x_{0,m}$, and synchronization to the leader is achieved. Therefore we focus henceforth on designing a controller such that local neighborhood errors $\|e^m\|$ is bounded.

10.2 Distributed Control Structure

In this section, we design a control protocol structure that is distributed in the sense that it depends only on local neighbor information. As such, it is allowable for multi-agent control in that it obeys the restrictions imposed by the communication graph structure. Sliding mode error variables are introduced to streamline the controller design. A Performance Lyapunov Equation is given that is complementary to the Graph Lyapunov Equation (10.8). Neural network approximators are introduced at each node to estimate the unknown agent nonlinearities.

10.2.1. Sliding Mode Error Variables and Performance Lyapunov Equation

Introduce sliding mode error variable $i \in N$ for each follower node $i = 1, \ldots, N$ as

$$r_i = \lambda_1 e_{i,1} + \lambda_2 e_{i,2} + \cdots + \lambda_{M-1} e_{i,M-1} + e_{i,M}, \quad i \in \mathcal{N}. \tag{10.10}$$

The design gain parameters λ_i are chosen such that the polynomial $s^{M-1} + \lambda_{M-1} s^{M-2} + \cdots + \lambda_1$ is Hurwitz. To ease the design, one can obtain such λ_i by solving the equation $s^{M-1} + \lambda_{M-1} s^{M-2} + \ldots + \lambda_1 = \Pi_{i=1}^{M-1}(s - \alpha_i)$ for all $s \in \mathbb{R}$, where α_i are arbitrary positive real numbers.

The cooperative tracking control objective is to keep the sliding mode errors r_i on or near the sliding surface $r_i = 0$. If $r_i = 0$, then $e_i \to 0$ exponentially as $t \to \infty$. If r_i is bounded, then, e_i will be bounded. Define the global sliding mode error $r = [r_1, \ldots, r_N]^T$. Then

$$r = \lambda_1 e^1 + \cdots + \lambda_{M-1} e^{M-1} + e^M.$$

Define

$$E_1 = [e^1, \ldots, e^{M-1}] \in \mathbb{R}^{N \times (M-1)},$$

$$E_2 = \dot{E}_1 = [e^2, \ldots, e^M] \in \mathbb{R}^{N \times (M-1)},$$

$$l = [0, \ldots, 0, 1]^T \in \mathbb{R}^{M-1},$$

and

$$\Lambda = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\lambda_1 & -\lambda_2 & -\lambda_3 & \cdots & -\lambda_{M-1} \end{bmatrix} \in \mathbb{R}^{(M-1) \times (M-1)}.$$

Then one has

$$E_2 = E_1 \Lambda^T + rl^T. \tag{10.11}$$

Since Λ is Hurwitz, given any positive number β, there exists a matrix $P_1 > 0$, such that the following *performance-related Lyapunov equation* holds.

$$\Lambda^T P_1 + P_1 \Lambda = -\beta I, \tag{10.12}$$

where $I \in \mathbb{R}^{(M-1) \times (M-1)}$ is the identity matrix.

Remark 10.2. **Graph Lyapunov Equation and Performance Lyapunov Equation.** It is emphasized that the Performance Lyapunov Equation (10.12) captures information about the local control design, and the Graph Lyapunov Equation (10.8) captures information about the communication graph topology. Both are needed in the Lyapunov proof of the main result in Section 10.3.

The dynamics of the sliding mode error r is

$$
\begin{aligned}
\dot{r} &= \lambda_1 \dot{e}^1 + \lambda_2 \dot{e}^2 + \cdots + \lambda_{M-1} \dot{e}^{M-1} + \dot{e}^M \\
&= \lambda_1 e^2 + \lambda_2 e^3 + \cdots + \lambda_{M-1} e^M - (L+B)(f(x) + u + \zeta - \underline{f_0}) \\
&= \rho - (L+B)(f(x) + u + \zeta - \underline{f}_0),
\end{aligned}
\tag{10.13}
$$

where

$$
\rho = \lambda_1 e^2 + \lambda_2 e^3 + \cdots + \lambda_{M-1} e^M = E_2 \bar{\lambda}
\tag{10.14}
$$

with $\bar{\lambda} = [\lambda_1, \ldots, \lambda_{M-1}]^T$.

The next lemma shows that (ultimate) boundedness of r_i implies the (ultimate) boundedness of e_i, $\forall i = 1, \ldots, N$.

Lemma 10.3. *For all* $i = 1, \ldots, N$, *suppose*

$$
\begin{aligned}
|r_i(t)| &\le \psi_i, \quad \forall t \ge t_0, \\
|r_i(t)| &\le \xi_i, \quad \forall t \ge T_i,
\end{aligned}
$$

for some bounds $\psi_i > 0$, $\xi_i > 0$, *and time* $T_i > t_0$. *Then there exist bounds* $\Psi_i > 0$, $\Xi_i > 0$ *and time* $\Delta_i > t_0$, *such that*

$$
\begin{aligned}
\|e_i(t)\| &\le \Psi_i, \quad \forall t \ge t_0, \\
\|e_i(t)\| &\le \Xi_i, \quad \forall t \ge \Delta_i.
\end{aligned}
$$

Proof. Let $\varrho_i = [e_{i,1}, \ldots, e_{i,M-1}]^T$. Then equation (10.10) implies

$$
\dot{\varrho}_i = \Lambda \varrho_i + l r_i.
\tag{10.15}
$$

Using a similar development as in [6] Lemma 2.1, it is straightforward to show that if $r_i(t)$ is bounded, i.e. $|r_i(t)| \leq \psi_i$ for $t \geq t_0$, then $\varrho_i(t)$ is bounded. Equation (10.15) further implies $\dot{\varrho}_i(t)$ is bounded, for $t \geq t_0$. Therefore, $e_i(t)$ is bounded, i.e. $\|e_i\| \leq \Psi_i, \forall t \geq t_0$ for some $\Psi_i > 0$. Similarly, we can show that if $r_i(t)$ is ultimately bounded, $e_i(t)$ is ultimately bounded. ∎

10.2.2. Local Neural Network Approximators for Unknown Nonlinearities

Their function approximation property makes neural networks (NN) a useful tool for solving control problems [13]. It is well known that a two-layer neural network is a universal approximator [7]. A two-layer neural network with v neurons in the hidden layer is depicted in Fig.10.1 with the mathematic expression be

$$z = \phi(\bar{W}^T \sigma(V^T y)), \tag{10.16}$$

where $y = [y_1, \ldots, y_n]^T$ is the input vector; $\sigma = [\sigma_1, \ldots, \sigma_v]^T$ is the activation function vector for the hidden layer; $V \in \mathbb{R}^{n \times v}$ is the input-layer weights matrix; $\bar{W} \in \mathbb{R}^v$ is the output-layer weights vector; ϕ is the activation function for the output layer. According to the NN approximation literature [7], for different applications, a variety of NN activation functions can be selected, such as sigmoids, and Gaussians/radial basis function (RBF). A list of frequently used activation functions can be found in Fig.1.1.3 in [13].
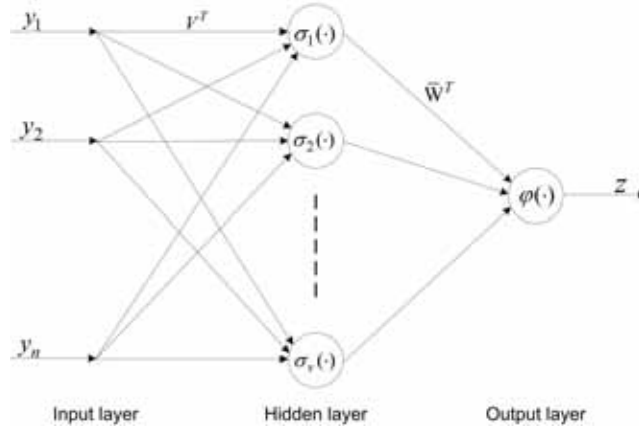

Fig. 10.1 A two-layer neural network

To avoid distraction from the main issues being introduced, in this chapter, we adopt a linear-in-parameters (LIP) NN, i.e. the hidden layer activation functions $\sigma_i(\cdot)$ and the input-layer weights matrix V are fixed, the output layer activation function is simply a summation function, and only the output weights $\bar{W}$ are tuned. The following development can be extended to a two-layer nonlinear-in-parameters NN as in [13].

It is assumed that the unknown nonlinearity $f_i(x_i)$ in system (10.1) can be expressed on a prescribed compact set $\Omega \subset \mathbb{R}^M$ by

$$f_i(x_i) = W_i^T \varphi_i(x_i) + \epsilon_i, \qquad (10.17)$$

where $\varphi_i(x_i) \in \mathbb{R}^{v_i}$ is a suitable set of v_i basis functions, $W_i \in \mathbb{R}^{v_i}$ is the ideal neural network weights, and ϵ_i is the NN approximation error. This means that each agent keeps a NN to approximate its own unknown nonlinearities.

Remark 10.3. By the Stone-Weierstrass approximation theorem [23], in the compact set Ω, given any positive number ε_i, there exists a large enough positive integer v_i^*, such that for any $v_i \geq v_i^*$, one can always find an ideal weight vector W_i and a suitable basis set $\varphi_i(\cdot)$, which make Eq. (10.17) satisfy the property that $max_{x_i \in \Omega} |\epsilon_i| \leq \varepsilon_i$.

Assume that the ideal weights vector W_i in Eq. (10.17) is unknown. Then let each node maintain a neural network locally to keep track of the current estimates for its unknown nonlinearities. Since the neural networks are maintained locally at each node, they only need to approximate the local nonlinearities in the dynamics of that node. Therefore, the number of neurons needed at each node is reduced compared to centralized neural adaptive control.

Define the approximation of $f_i(x_i)$ as

$$\hat{f}_i(x_i) = \hat{W}_i^T(t) \varphi_i(x_i), \qquad (10.18)$$

where $\hat{W}_i(t) \in \mathbb{R}^{v_i}$ (or $\hat{W}_i$ for convenience) is the actual value of the NN weights vector for the i-th node and must be determined using only the local information available to node i in the graph (in fact, by the distributed NN weights tuning law (10.23)). Note that $\hat{f}_i(x_i)$ is the actual output of the NN maintained by node i.

Define $\epsilon = [\epsilon_1, ..., \epsilon_N]^T \in \mathbb{R}^N$, $\varphi(x) = [\varphi_1^T(x_1), ..., \varphi_N^T(x_N)]^T \in \mathbb{R}^{\sum_{i=1}^{N} v_i}$, and

$$W = \begin{bmatrix} W_1 & & & \\ & W_2 & & \\ & & \ddots & \\ & & & W_N \end{bmatrix} \in \mathbb{R}^{(\sum_{i=1}^N v_i) \times N},$$

$$\hat{W} = \begin{bmatrix} \hat{W}_1 & & & \\ & \hat{W}_2 & & \\ & & \ddots & \\ & & & \hat{W}_N \end{bmatrix} \in \mathbb{R}^{(\sum_{i=1}^N v_i) \times N}.$$

Then the global nonlinearity $f(x)$ can be written as

$$f(x) = W^T \varphi(x) + \epsilon, \tag{10.19}$$

and its approximation is

$$\hat{f}(x) = \hat{W}^T \varphi(x). \tag{10.20}$$

The error of the NN weights vector is defined as

$$\tilde{W} = W - \hat{W}.$$

It is important to note that matrices W, $\hat{W}$ and $\tilde{W}$ are all block diagonal, a fact that is used later in the proof of Theorem 10.1 to obtain cooperative tuning laws of distributed form.

Remark 10.4. Denote $\phi_{iM} = max_{x_i \in \Omega} \| \phi_i(x_i) \|$ and $W_{iM} = \| W_i \|$. According to the definitions of ϕ, W and ϵ, it is observed that there exist positive numbers Φ_M, W_M and ϵ_M, such that $\| \phi \| \le \Phi_M$, $\| W \|_F \le W_M$ and $\| \epsilon \| \le \epsilon_M$, where $\| \cdot \|_F$ is the Frobenius norm.

10.2.3. Distributed Control Law Structure

This section presents the structure of the distributed adaptive control protocols u_i in (10.1). These must be distributed in the sense that they only depend on the local neighbor information available to each agent. Then, they are allowable in the sense that they obey the restrictions imposed by the communication graph structure.

To proceed, we require the following standard assumptions.

Assumption 10.2

a) *There exists a positive number $X_M > 0$ such that $\|x_0(t)\| \le X_M$, $\forall t \ge t_0$.*

b) *There exists a continuous function $g(\cdot) : \mathbb{R}^M \to \mathbb{R}$, such that $|f_0(t, x_0)| \le |g(x_0)|$, $\forall x_0 \in \mathbb{R}^M$, $\forall t \ge t_0$.*

c) *For each node i, the disturbance ζ_i is unknown, but bounded. Thus the overall global disturbance vector ζ is also bounded by $\|\zeta\| \le \zeta_M$ where ζ_M can be unknown.*

Remark 10.5. Since node 0 acts as a command generator, Assumption 10.2 a) is reasonable in practical applications. According to Assumption 10.2 b), it is straightforward to show that there exists a positive number F_M such that $|f_0(t, x_0)| \le F_M$, $\forall x_0 \in \Omega_0$ and $\forall t \ge t_0$, where $\Omega_0 = \{x_0 \in \mathbb{R}^M \mid \|x_0\| \le X_M\}$.

Remark 10.6. The bounds X_M, ζ_M, F_M, W_M and ϵ_M (as in Assumption 10.2 a), c), Remark 10.5 and Remark 10.4, respectively) are actually not used in the controller design. Thus they do not have to be known. They are only required for the stability analysis. The bound Φ_M (in Remark 10.4) is needed to choose the design parameter c as in condition (10.25). Since generally we can choose some squashing functions as the basis set, such as sigmoids, Gaussians, and hyperbolic tangents, Φ_M can be explicitly computed.

According to Remark 10.3, v_i neurons are used at each node i. Design the distributed control law for each node i as

$$u_i = \frac{1}{d_i + b_i}(\lambda_1 e_{i,2} + \cdots + \lambda_{M-1} e_{i,M}) - \hat{W}_i^T(t)\phi_i(x_i) + cr_i, \tag{10.21}$$

where c is the control gain. This is a distributed protocol that only depends on local information available in the graph about node i and its neighbors. This can be written collectively for all the agents as

$$u = (D + B)^{-1}\rho - \hat{W}^T\phi(x) + cr. \tag{10.22}$$

Remark 10.7. When $M = 1$ and $M = 2$, control law (10.21) reduces to the control law for the first-order nonlinear system and the second-order nonlinear system, as

presented in Chapter 8 and Chapter 9, respectively.

10.3 Distributed Tuning Laws for Cooperative Adaptive Control

For cooperative adaptive control, both the control law structure and the adaptive parameter tuning laws must be distributed, that is, they are only allowed to depend on local neighbor information. A suitable distributed control protocol structure was developed in the previous section. Developing distributed adaptive tuning laws for multi-agent system is a challenge and depends on constructing a suitable Lyapunov function that captures both the controller design at each node and the communication graph topology. This is achieved in the proof of the upcoming main theorem by using both the Graph Lyapunov Equation (10.8) and the Performance Lyapunov Equation (10.12).

Let the NN parameter adaptive tuning laws be taken as

$$\dot{\hat{W}}_i = -F_i \varphi_i r_i p_i (d_i + b_i) - \kappa F_i \hat{W}_i, \tag{10.23}$$

or collectively

$$\dot{\hat{W}} = -F \varphi r P (D + B) - \kappa F \hat{W}, \tag{10.24}$$

where design parameters $F_i = F_i^T \in \mathbb{R}^{v_i \times v_i}$ are arbitrary positive definite matrices and $F = diag\{F_1, ..., F_N\}$; κ is a positive scalar tuning gain; P is defined in Lemma 1. Note the control law (10.21) and NN tuning law (10.23) are implemented using only local information available for node i. As such, they are allowable in terms of obeying the communication restrictions impose by the graph topology.

The main result of this chapter is given by the following theorem. It shows that under the proposed cooperative control laws and NN tuning laws, the tracking errors δ^m are cooperatively uniformly ultimately bounded, thereby showing synchronization in the sense of CUUB for the whole graph $\mathcal{G}$. Then, all agents synchronize to the state of the control node dynamics (10.3).

The result depends on Assumption 10.1 and Assumption 10.2. Interestingly, the former concerns the communication graph topology, while the latter concerns the local agent dynamics.

Theorem 10.1. *Consider the distributed systems (10.1) and the leader node dynamics (10.3). Suppose Assumption 10.1 and Assumption 10.2 hold. Let the distributed control laws be (10.21) and the distributed NN tuning laws be (10.23). Let the control gain c satisfy*

$$c > \frac{2}{\underline{\sigma}(Q)}\left(\frac{\gamma^2}{\kappa} + \frac{2}{\beta}g^2 + h\right), \tag{10.25}$$

with $\gamma = -\frac{1}{2}\Phi_M\bar{\sigma}(P)\bar{\sigma}(A)$, $g = -\frac{1}{2}\left(\frac{\bar{\sigma}(P)\bar{\sigma}(A)}{\underline{\sigma}(D+B)}\|\Lambda\|_F\|\bar{\lambda}\| + \bar{\sigma}(P_1)\right)$ *and* $h = \frac{\bar{\sigma}(P)\bar{\sigma}(A)}{\underline{\sigma}(D+B)}\|\bar{\lambda}\|$ *where* P_1 *is defined in (10.12) for any* $\beta > 0$ *and* Q *is defined as in Lemma 10.1.* *Then:*

1) *the tracking errors* δ^m , *for all* $m = 1, ..., M$, *are cooperatively uniformly ultimately bounded, which implies that all nodes in graph* $\mathcal{G}$ *synchronize to the leader node 0 with bounded residual errors.*

2) *the states* $x_i(t)$ *are bounded* $\forall t \ge t_0$ *for all* $i \in \mathcal{N}$.

Proof.

1) For all $i = 1, ..., N$, given fixed positive numbers ε_i , there exist numbers of neurons v_i^* , such that when $v_i \ge v_i^*$, the NN approximation errors satisfy $\|\epsilon_i\| \le \varepsilon_i$. Then by Remark 10.4, one has $\|W\|_F \le W_M$, $\|\phi\| \le \Phi_M$ and $\|\epsilon\| \le \epsilon_M$.

Consider the Lyapunov function candidate

$$V = V_1 + V_2 + V_3, \tag{10.26}$$

where $V_1 = \frac{1}{2}r^T Pr$, $V_2 = \frac{1}{2}tr\{\tilde{W}^T F^{-1}\tilde{W}\}$ and $V_3 = \frac{1}{2}tr\{E_1 P_1 (E_1)^T\}$.

First we compute the derivative of V_1 .

$$\dot{V}_1 = r^T P\dot{r} = r^T P[\rho - (L+B)(f(x) + u + \zeta - \underline{f_0})] \tag{10.27}$$

For notational simplicity, denote $f(x)$ as f , $\hat{f}(x)$ as $\hat{f}$ and $\phi(x)$ as ϕ in the sequel. Substituting (10.22) into (10.27), and considering (10.19) and (10.20), and $L = D - A$, one has

$$\dot{V}_1 = r^T P[\rho - (L+B)(f + \zeta - \underline{f_0} + (D+B)^{-1}\rho - \hat{f} + cr)]$$
$$= r^T P[-(L+B)(f + \zeta - \underline{f_0} - \hat{f} + cr) + A(D+B)^{-1}\rho]$$
$$= r^T P[-(L+B)(\tilde{W}^T\phi + \epsilon + \zeta - \underline{f_0} + cr) + A(D+B)^{-1}\rho]$$

$$\dot{V}_1 = -r^T P(L+B)\tilde{W}\phi - r^T P(L+B)(\epsilon + \zeta - \underline{f_0})$$
$$- cr^T P(L+B)r + r^T PA(D+B)^{-1}\rho$$
$$= -r^T P(L+B)(\epsilon + \zeta - \underline{f_0}) - cr^T P(L+B)r$$
$$- r^T P(D+B)\tilde{W}^T \phi + r^T PA\tilde{W}^T \phi + r^T PA(D+B)^{-1}\rho. \tag{10.28}$$

Noting the fact that $x^T y = tr\{yx^T\}$, $\forall x, y \in \mathbb{R}^N$, and considering Eq. (10.8), one can write $\dot{V}_1$ as

$$\dot{V}_1 = -r^T P(L+B)(\epsilon + \zeta - \underline{f_0})$$
$$- \frac{1}{2} cr^T Qr - tr\{\tilde{W}^T \phi r^T P(D+B)\}$$
$$+ tr\{\tilde{W}^T \phi r^T PA\} + r^T PA(D+B)^{-1}\rho. \tag{10.29}$$

Since $\dot{\tilde{W}} = \dot{W} - \dot{\hat{W}} = -\dot{\hat{W}}$, considering (10.24), one has

$$\dot{V}_1 + \dot{V}_2 = -r^T P(L+B)(\epsilon + \zeta - \underline{f_0}) - \frac{1}{2} cr^T Qr$$
$$- tr\{\tilde{W}^T \phi r^T P(D+B)\} + tr\{\tilde{W}^T \phi r^T PA\}$$
$$+ r^T PA(D+B)^{-1}\rho - tr\{\tilde{W}^T F^{-1}\dot{\hat{W}}\}$$
$$= -\frac{1}{2} cr^T Qr - r^T P(L+B)(\epsilon + \zeta - \underline{f_0})$$
$$+ \kappa tr\{\tilde{W}^T \hat{W}\} + tr\{\tilde{W}^T \phi r^T PA\} + r^T PA(D+B)^{-1}\rho. \tag{10.30}$$

Considering (10.14) and (10.11), one can further write Eq. (10.30) as

$$\dot{V}_1 + \dot{V}_2 = -\frac{1}{2} cr^T Qr - r^T P(L+B)(\epsilon + \zeta - \underline{f_0})$$
$$+ \kappa tr\{\tilde{W}^T W\} - \kappa \|\tilde{W}\|_F^2 + tr\{\tilde{W}^T \phi r^T PA\}$$
$$+ r^T PA(D+B)^{-1} E_1 \Lambda^T \overline{\lambda} + r^T PA(D+B)^{-1} rl^T \overline{\lambda}. \tag{10.31}$$

Let $T_M = \epsilon_M + \zeta_M + F_M$. Then

$$\dot{V}_1 + \dot{V}_2$$
$$\leq -\frac{1}{2} c\underline{\sigma}(Q)\|r\|^2 + \overline{\sigma}(P)\overline{\sigma}(L+B)T_M \|r\| - \kappa \|\tilde{W}\|_F^2$$
$$+ \Phi_M \overline{\sigma}(P)\overline{\sigma}(A)\|\tilde{W}\|_F \|r\| + \frac{\overline{\sigma}(P)\overline{\sigma}(A)}{\underline{\sigma}(D+B)} \|r\|^2 \|l\| \|\overline{\lambda}\|$$
$$+ \|r\| \frac{\overline{\sigma}(P)\overline{\sigma}(A)}{\underline{\sigma}(D+B)} \|E_1\|_F \|\Lambda\|_F \|\overline{\lambda}\| + \kappa W_M \|\tilde{W}\|_F$$

$$
\begin{aligned}
= & -\left(\frac{1}{2}c\underline{\sigma}(Q) - \frac{\overline{\sigma}(P)\overline{\sigma}(A)}{\underline{\sigma}(D+B)}\|\bar{\lambda}\|\right)\|r\|^2 - \kappa\|\tilde{W}\|_F^2 \\
& + \Phi_M\overline{\sigma}(P)\overline{\sigma}(A)\|\tilde{W}\|_F\|r\| + \kappa W_M\|\tilde{W}\|_F + \overline{\sigma}(P)\overline{\sigma}(L+B)T_M\|r\| \\
& + \frac{\overline{\sigma}(P)\overline{\sigma}(A)}{\underline{\sigma}(D+B)}\|\Lambda\|_F\|\bar{\lambda}\|\|E_1\|_F\|r\|.
\end{aligned}
\tag{10.32}
$$

Next we compute the derivative of V_3.

$$
\dot{V}_3 = \frac{1}{2}tr\{\dot{E}_1 P_1(E_1)^T + E_1 P_1(\dot{E}_1)^T\} = tr\{E_2 P_1(E_1)^T\}.
\tag{10.33}
$$

Substituting (10.11) into (10.33) and considering (10.12) gives

$$
\begin{aligned}
\dot{V}_3 &= tr\{E_1\Lambda^T P_1(E_1)^T\} + tr\{rl^T P_1(E_1)^T\} \\
&= \frac{1}{2}tr\{E_1(\Lambda^T P_1 + P_1\Lambda)(E_1)^T\} + tr\{rl^T P_1(E_1)^T\} \\
&= -\frac{\beta}{2}tr\{E_1(E_1)^T\} + tr\{rl^T P_1(E_1)^T\}.
\end{aligned}
\tag{10.34}
$$

Thus, noting $\|l\|=1$, one has

$$
\begin{aligned}
\dot{V}_3 &\leq -\frac{\beta}{2}\|E_1\|_F^2 + \overline{\sigma}(P_1)\|l\|\|r\|\|E_1\|_F \\
&= -\frac{\beta}{2}\|E_1\|_F^2 + \overline{\sigma}(P_1)\|r\|\|E_1\|_F.
\end{aligned}
\tag{10.35}
$$

Therefore,

$$
\begin{aligned}
\dot{V} \leq & -\left(\frac{1}{2}c\underline{\sigma}(Q) - \frac{\overline{\sigma}(P)\overline{\sigma}(A)}{\underline{\sigma}(D+B)}\|\bar{\lambda}\|\right)\|r\|^2 - \kappa\|\tilde{W}\|_F^2 \\
& -\frac{\beta}{2}\|E_1\|_F^2 + \Phi_M\overline{\sigma}(P)\overline{\sigma}(A)\|\tilde{W}\|_F\|r\| \\
& +\left(\frac{\overline{\sigma}(P)\overline{\sigma}(A)}{\underline{\sigma}(D+B)}\|\Lambda\|_F\|\bar{\lambda}\| + \overline{\sigma}(P_1)\right)\|r\|\|E_1\|_F \\
& +\overline{\sigma}(P)\overline{\sigma}(L+B)T_M\|r\| + \kappa W_M\|\tilde{W}\|_F.
\end{aligned}
\tag{10.36}
$$

Let $\gamma = -\frac{1}{2}\Phi_M\overline{\sigma}(P)\overline{\sigma}(A)$, $g = -\frac{1}{2}\left(\frac{\overline{\sigma}(P)\overline{\sigma}(A)}{\underline{\sigma}(D+B)}\|\Lambda\|_F\|\bar{\lambda}\| + \overline{\sigma}(P_1)\right)$ and $h = \frac{\overline{\sigma}(P)\overline{\sigma}(A)}{\underline{\sigma}(D+B)}\|\bar{\lambda}\|$. Then (10.36) is written as

$$
\dot{V} \leq -z^T K z + \omega^T z = -V_z(z),
\tag{10.37}
$$

where

$$z = \left[\| E_1 \|_F, \| \tilde{W} \|_F, \| r \| \right]^T, \tag{10.38}$$

$$K = \begin{bmatrix} \frac{\beta}{2} & 0 & g \\ 0 & \kappa & \gamma \\ g & \gamma & \mu \end{bmatrix}, \quad \mu = \tfrac{1}{2} c \underline{\sigma}(Q) - h \quad \text{and} \quad \omega = [0, \kappa W_M, \bar{\sigma}(P)\bar{\sigma}(L+B)T_M]^T.$$

$V_z(z)$ is positive definite if the following conditions C1 and C2 hold:

C1) K is positive definite.

C2) $\| z \| > \frac{\| \omega \|}{\underline{\sigma}(K)}$.

According to Sylvester's criterion, $K > 0$ if

$$\beta > 0, \tag{10.39}$$

$$\beta \kappa > 0, \tag{10.40}$$

$$\kappa(\beta\mu - 2g^2) - \beta\gamma^2 > 0. \tag{10.41}$$

Solving the above system of equations gives the equivalent condition (10.25). Since $\| \omega \|_1 > \| \omega \|$, condition C2) holds if $\| z \| \geq Bd$, where

$$Bd = \frac{\| \omega \|_1}{\underline{\sigma}(K)} = \frac{\bar{\sigma}(P)\bar{\sigma}(L+B)T_M + \kappa W_M}{\underline{\sigma}(K)}. \tag{10.42}$$

Therefore, under conditions (10.25), one has

$$\dot{V} \leq -V_z(z), \quad \forall \| z \| \geq Bd, \tag{10.43}$$

with $V_z(z)$ being a continuous positive definite function.

Straightforward computation of (10.26) implies

$$\underline{\sigma}(\Gamma) \| z \|^2 \leq V \leq \bar{\sigma}(T) \| z \|^2, \tag{10.44}$$

where $\Gamma = diag\left(\frac{\underline{\sigma}(P_1)}{2}, \frac{1}{2\bar{\sigma}(F)}, \frac{\underline{\sigma}(P)}{2} \right)$ and $T = diag\left(\frac{\bar{\sigma}(P_1)}{2}, \frac{1}{2\underline{\sigma}(F)}, \frac{\bar{\sigma}(P)}{2} \right)$. Then following Theorem 4.18 in [10], one can draw the conclusion that for any initial value $z(t_0)$ (or equivalently $V(t_0)$), there exists a time T_0, such that

$$\| z(t) \| \leq \sqrt{\frac{\bar{\sigma}(T)}{\underline{\sigma}(\Gamma)}} Bd, \quad \forall t \geq t_0 + T_0 \tag{10.45}$$

Let $k = min_{\|z\| \geq Bd} \dot{V}_z(z)$. Then using essentially the same development as in [10], it can be shown that

$$T_0 = \frac{V(t_0) - \bar{\sigma}(T)Bd^2}{k}.$$ (10.46)

By definition of z, (10.45) implies that $r(t)$ is ultimately bounded. Hence $r_i(t)$ ($\forall i \in \mathcal{N}$) is ultimately bounded. Then by Lemma 10.3, $e_i(t)$ is ultimately bounded, which implies $e^m(t)$ ($\forall m = 1, ..., M$) is ultimately bounded. Then, following Lemma 10.2, the synchronization errors $\delta^m(t)$ are CUUB and all nodes in graph $\mathcal{G}$ synchronize, in the sense of Definition 10.1, to the trajectory $x_0(t)$ of the leader node.

2) It can be shown that the states $x_i(t)$ are bounded $\forall t \geq t_0$ for all $i \in \mathcal{N}$. In fact, inequality (10.37) implies

$$\dot{V} \leq -\underline{\sigma}(K)\|z\|^2 + \|\omega\|\|z\|.$$ (10.47)

Then combination of (10.44) and (10.47) gives

$$\frac{d}{dt}(\sqrt{V}) \leq -\frac{\underline{\sigma}(K)}{2\bar{\sigma}(T)}\sqrt{V} + \frac{\|\omega\|}{2\sqrt{\underline{\sigma}(\Gamma)}}.$$ (10.48)

Thus $V(t)$ is bounded $\forall t \geq t_0$ by Corollary 1.1 in [5]. Since (10.26) implies $\|r\|^2 \leq \frac{2V(t)}{\underline{\sigma}(P)}$, there follows the boundedness of $r(t)$, $\forall t \geq t_0$. By Lemma 10.2 and Lemma 10.3, it is straightforward that $\delta^m(t)$ is also bounded. Noting $\delta^m = x^m - \underline{x}_{0,m}$ and considering Assumption 10.2 a), one can see that $x^m(t)$, $\forall m = 1, ..., M$, is bounded all the time, i.e. $x_i(t)$ is bounded, $\forall i \in \mathcal{N}$. In other words, $x_i(t)$ is contained in compact sets Ω_i for all $t \geq t_0$, $\forall i \in \mathcal{N}$. ∎

For higher-order synchronization of cooperative systems, the interplay between node dynamics and graph structure is more complicated than that for the first- and second-order cases. This is reflected in the fact that two Lyapunov equations are involved in the stability analysis. One is for the individual node controller design (Eq. (10.12)), and the other for the graph structure (Eq. (10.8)).

Remark 10.8. Finite-time convergence. Inequality (10.45) and Eq. (10.38) imply that the sliding mode error r_i reaches a bounded neighborhood of the origin in fi-

nite time T_0 given in (10.46). With the help of Lemma 10.2 and Lemma 10.3, it can be shown that the tracking error δ_i reaches the final bound in finite time.

Remark 10.9. **Design issues.** The parameters $p_i > 0$ in adaptive tuning laws (10.23) depend on the graph topology through equations (10.6), (10.7) which require global information. However, the design parameter F_i is an arbitrary positive definite matrix, which implies $p_i F_i$ is an arbitrary positive definite matrix. This finally implies that one can pick an arbitrary positive number $p_i > 0$ for the NN tuning law. Therefore, the proposed controller u_i is completely distributed, as one can see from (10.21) and (10.23). It is also worth noting that although Eq. (10.42) is the worst case for the ultimate error bound, it provides a practical design tool for adaptive higher-order synchronization. This bound shows relations between the graph topology and the control design parameters in reducing the error bound.

Remark 10.10. **Graph Lyapunov Equation and Performance Lyapunov Equation.** This proof verifies the performance of the cooperative control laws and the adaptive tuning laws, both of which are distributed in the sense of depending only on local neighborhood information. This is accomplished by carefully constructing the Lyapunov function in the proof to depend on both the Graph Lyapunov Equation (10.8) and the Performance Lyapunov Equation (10.12). The former appears in Lyapunov function V_1 and the latter in V_3.

Remark 10.11. It is well known in the literature of neural network control ([5], [13]) that the size of the compact set Ω in Remark 10.3 is not needed for the controller design. It is commonly assumed to be as large as desired in practical applications, such that state x_i will stay inside Ω, i.e. $\Omega_i \subset \Omega$, $\forall t \geq t_0$. In this sense, the results obtained in this chapter is semi-global. In the case where Remark 10.3 and Remark 10.4 hold for all $x_i \in \mathbb{R}^M$, the results will be global.

10.4 Simulation Example

This simulation example shows the effectiveness of the proposed cooperative adaptive controller. Consider a 5-node digraph $\mathcal{G}$ and a leader node 0 as depicted in Fig.10.2. Note that this topology satisfies Assumption 10.1. Each node is modeled by a third order nonlinear dynamics. Let the dynamics of the leader node 0 be

$$\dot{x}_{0,1} = x_{0,2}$$
$$\dot{x}_{0,1} = x_{0,2}$$
$$\dot{x}_{0,3} = -x_{0,2} - 2x_{0,3} + 1 + 3sin(2t) + 6cos(2t) \tag{10.49}$$
$$-\frac{1}{3}(x_{0,1} + x_{0,2} - 1)^2(x_{0,1} + 4x_{0,2} + 3x_{0,3} - 1).$$

This is modified from the FitzHugh-Nagumo model [21], and selected to illustrate the higher-order feature of our algorithm. These dynamics satisfy Assumption 10.2. Its state trajectory is shown in Fig.10.3 with initial condition $x_0 = [3, 4, 1]^T$.
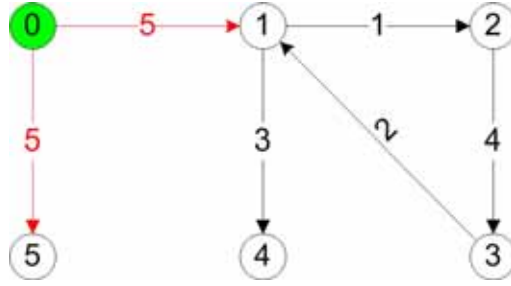

Fig. 10.2 Topology of the augmented graph $\overline{\mathcal{G}}$

The follower nodes are described by third order nonlinear systems

$$\dot{x}_{i,1} = x_{i,2}$$
$$\dot{x}_{i,2} = x_{i,3}$$
$$\dot{x}_{i,3} = f_i(x_i) + u_i + \zeta_i, \quad i = 1, \dots, 5,$$

with

$$\dot{x}_{1,3} = x_{1,2}sin(x_{1,1}) + cos(x_{1,3})^2 + u_1 + \zeta_1,$$
$$\dot{x}_{2,3} = -(x_{2,1})^2 x_{2,2} + 0.01x_{2,1} - 0.01(x_{2,1})^3 + u_2 + \zeta_2,$$
$$\dot{x}_{3,3} = x_{3,2} + sin(x_{3,3}) + u_3 + \zeta_3,$$
$$\dot{x}_{4,3} = -3(x_{4,1} + x_{4,2} - 1)^2(x_{4,1} + x_{4,2} + x_{4,3} - 1) \tag{10.50}$$
$$\quad - x_{4,2} - x_{4,3} + 0.5sin(2t) + cos(2t) + u_4 + \zeta_4,$$
$$\dot{x}_{5,3} = cos(x_{5,1}) + u_5 + \zeta_5.$$

The disturbances ζ_i are taken as random and bounded by $|\zeta_i| \le 2$. Note that the open-loop system of node 2 (without disturbance) exhibits a chaotic behavior for the initial value $x_2 = [0.325, 0.5, 0]^T$, as shown in Fig.10.4; the open-loop system of

node 3 has a finite escape time; the open-loop system of node 4 has the same structure as the leader node dynamics but with different parameters.

Let $x_0(0) = [3, 4, 1]^T$, $x_1(0) = [4, 1.6, 0.9]^T$, $x_2(0) = [0.325, 0.5, 0]^T$, $x_3(0) = [0.4, 0, -1]^T$, $x_4(0) = [1, 1, 0]^T$, and $x_5(0) = [5, -3, 2]^T$. In the simulation, it is observed that NNs with only a small number of neurons give good performance. In this example, 6 neurons are used for each NN. The NN weights are initialized to be $\hat{W}_i(0) = [0, 0, 0, 0, 0, 0]^T$, $\forall i$. Sigmoid basis functions are used. Choose design parameter as $\lambda_1 = 100$, $\lambda_2 = 20$, $c = 600$, $\kappa = 0.01$, $F_i = 2000I$, and choose p_i in the NN adaptive tuning law (10.23) arbitrarily, as long as it is positive.

The control signals are shown in Fig.10.5. As one can see, the control force is very large initially. This is reasonable, because the neural network maintained by each follower node has no prior knowledge of the nonlinear systems and it takes time to learn the unknown nonlinearities. The state trajectories for each node are shown in Figs.10.6–10.8. Fast tracking ability can be observed. Also note that the tracking errors δ^1 , δ^2 , and δ^3 do not go to zero, but are bounded by small residual errors. These are shown in Figs. 10.9–10.11. These figures demonstrate the efficiency of the proposed algorithm in guaranteeing synchronization despite the presence of complex unknown dynamics.
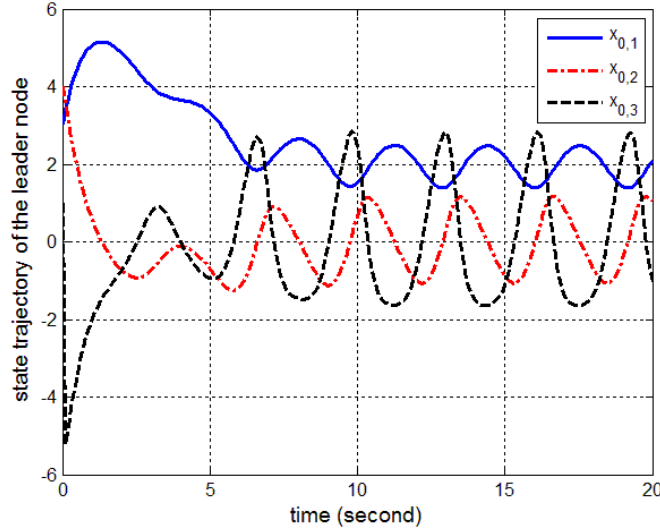

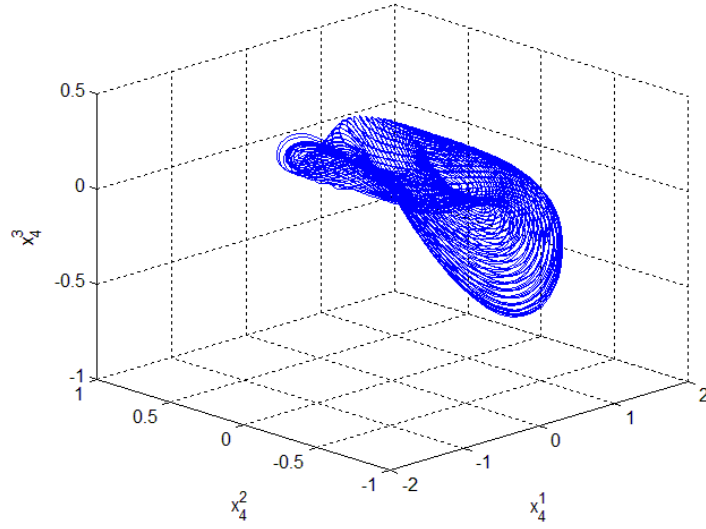**Fig. 10.3** State trajectory of the leader node 0

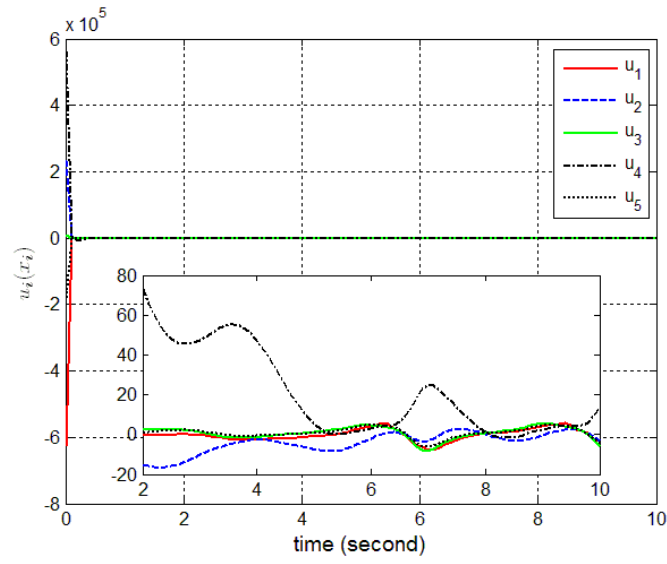**Fig. 10.4** 3-D plot of node 2 in isolation with initial value $x_2 = [0.325, 0.5, 0]^T$
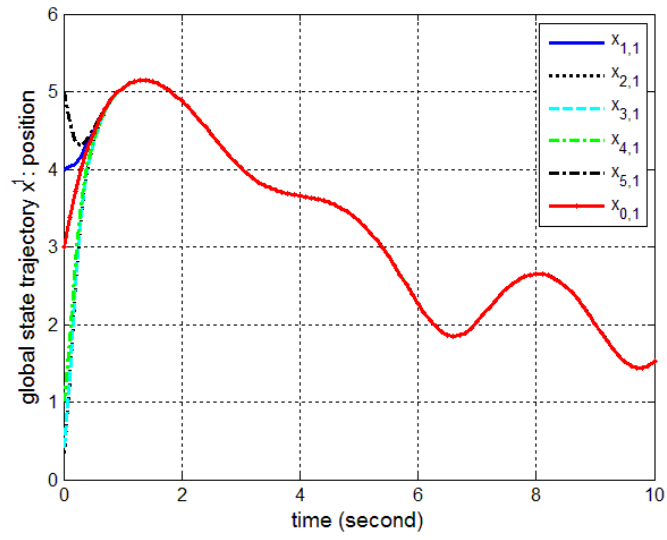

Fig. 10.5 Control signals u_i
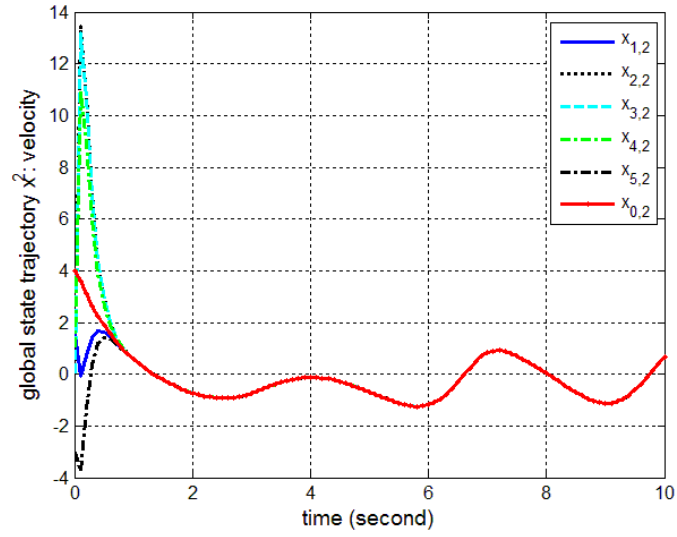
Fig. 10.6 Node positions in vector x^1

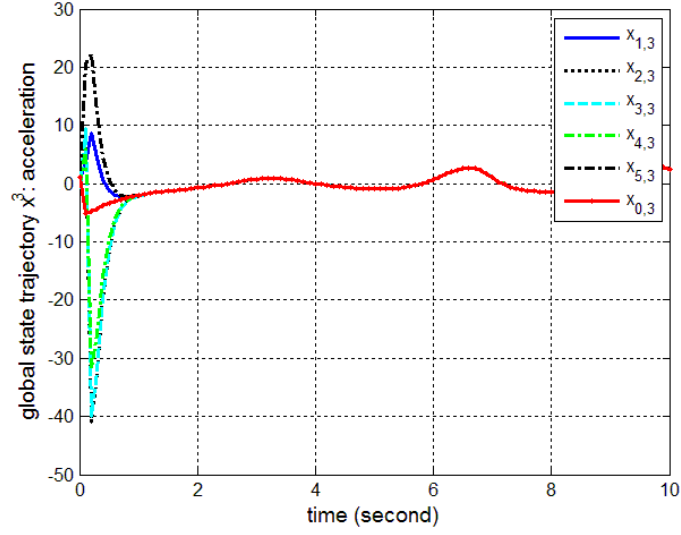**Fig. 10.7** Node velocities in vector x^2
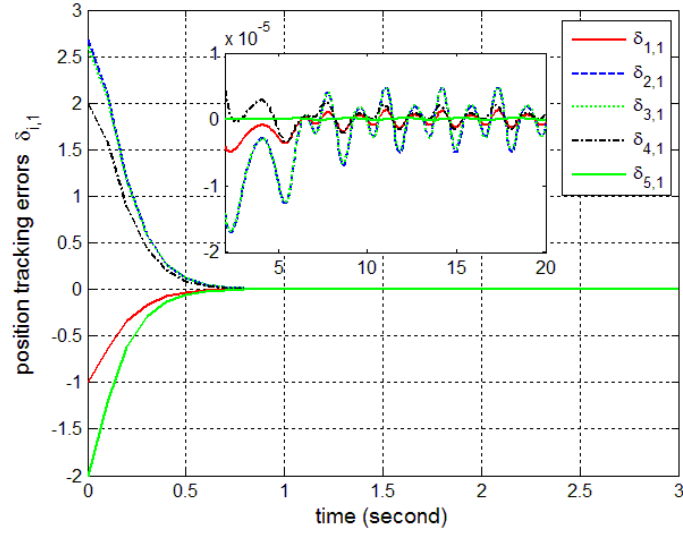
Fig. 10.8 Node accelerations in vector x^3

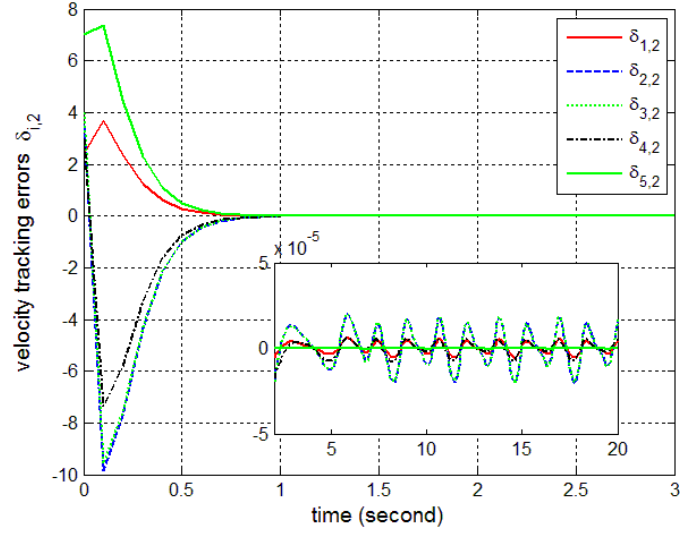**Fig. 10.9** Error components of global position disagreement vector δ^1

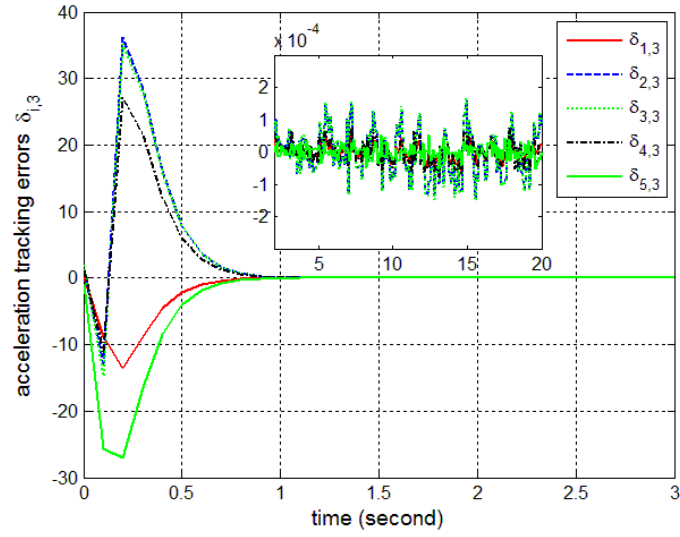**Fig. 10.10** components of global velocity disagreement vector δ^2

Fig. 10.11 Components of global acceleration disagreement vector δ^3

References

[1] G. Chen, F.L. Lewis, and L. Xie. Finite-time distributed consensus via binary control protocols. Automatica, 47(9),1962-1968, 2011.

[2] A. Das and F.L. Lewis. Distributed adaptive control for synchronization of unknown nonlinear networked systems. Automatica, 46(12):2014–2021, 2010.

[3] A. Das and F.L. Lewis. Cooperative adaptive control for synchronization of second-order systems with unknown nonlinearities. Int. J. Robust. Nonlinear Control, 21(3), 1509-1524, 2011.

[4] H. Du, S. Li, and C. Qian. Finite-time attitude tracking control of spacecraft with application to attitude synchronization. IEEE Trans. Autom. Control, 2011. to be published. DOI: 10.1109/TAC.2011.2159419.

[5] S.S. Ge and C. Wang. Adaptive neural control of uncertain MIMO nonlinear systems. IEEE Trans. Neural Netw., 15(3):674–692, 2004.

[6] S.S. Ge and J. Zhang. Neural network control of non-affine nonlinear systems with zero dynamics by state and output feedback. IEEE Trans. Neural Netw., 14(4):900–918, 2003.

[7] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. Neural Networks, 2(5):359–366, 1989.

[8] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans. Autom. Control, 48(6):988–1001, Jun. 2003.

[9] F. Jiang, L. Wang, and Y. Jia. Consensus in leaderless networks of high-order-integrator agents. In Proc. Amer. Control Conf., pages 4458–4463, 2009.

[10] H.K. Khalil. Nonlinear Systems. Prentice Hall, Upper Saddle River, NJ, 3rd edition, 2002.

[11] S. Khoo, L. Xie, and Z. Man. Robust finite-time consensus tracking algorithm for multi-robot systems. IEEE Trans. Mechatron., 14(2):219–228, Apr. 2009.

[12] T. Lee and H.-S. Ahn. Consensus of nonlinear systems using feedback linearization. In Proc. IEEE/ASME Conf. MESA, pages 26–31, 2010.

[13] F.L. Lewis, S. Jagannathan, and A. Yesildirek. Neural Network Control of Robot Manipulators and Nonlinear Systems. Taylor and Francis, London, 1999.

[14] F.L. Lewis, A. Yesildirek, and K. Liu. Multilayer neural net robot controller with guaranteed tracking performance. IEEE Trans. Neural Netw., 7(2):388–399, 1996.

[15] X. Li, X. Wang, and G. Chen. Pinning a complex dynamical network to its equilibrium. IEEE Trans. Circuits Syst. I, Reg. Papers, 51(10):2074–2087, Oct. 2004.

[16] Z. Li, Z. Duan, G. Chen, and L. Huang. Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint. IEEE Trans. Circuits Syst. I, Reg. Papers, 57(1):213–224, Jan. 2010.

[17] Z. Qu. Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles. Springer-Verlag, London, 2009.

[18] Z. Qu, J. Chunyu, and J. Wang. Nonlinear cooperative control for consensus of nonlinear and heterogeneous systems. In Proc. IEEE Conf. Decision Control, pages 2301–2308, 2007.

[19] W. Ren, R.W. Beard, and E.M. Atkins. A survey of consensus problems in multi-agent coordination. In Proc. Amer. Control Conf., pages 1859–1864, Portland, OR, 2005.

[20] W. Ren, K.L. Moore, and Y. Chen. High-order and model reference consensus algorithms in cooperative control of multivehicle systems. J. Dyn. Syst., Meas., Control, 129(5):678–688, 2007.

[21] J. Rinzel. A formal classification of bursting mechanisms in excitable systems. In Teramoto E. and Yamaguti M., editors, Mathematical Topics in Population Biology, Morphogenesis and Neurosciences, pages 267–281. Springer-Verlag, 1987.

[22] P.N. Shivakumar and K.H. Chew. A sufficient condition for nonvanishing of determinants. Proc. Amer. Math. Soc., 43(1):63–66, 1974.

[23] M.H. Stone. The generalized Weierstrass approximation theorem. Mathematics Magazine, 21(4/5):167–184 / 237–254, 1948.

[24] J. Wang and D. Cheng. Consensus of multi-agent systems with higher order dynamics. In Proc. of the 26th Chinese Control Conference, pages 761–765, 2007.

[25] X. Wang and G. Chen. Pinning control of scale-free dynamical networks. Physica A, 310(3-4):521–531, 2002.

[26] H. Zhang and F.L. Lewis. Synchronization of networked higher-order nonlinear systems with unknown dynamics. In Proc. IEEE Conf. Decision Control, pages 7129–7134, Atlanta, GA, 2010.

[27] H. Zhang and F.L. Lewis. Adaptive cooperative tracking control of higher-order nonlinear systems with unknown dynamics. Automatica, 48(7): 1432-1439, 2012.

[28] H. Zhang, F.L. Lewis, and A. Das. Optimal design for synchronization of cooperative systems: state feedback, observer and output feedback. IEEE Trans. Autom. Control, 56(8), 1948-1952, 2011