Chapter 5

Synchronization

The problem of consensus in the preceding chapter requires all the agents to converge to the same value, which is *static* in steady state. A generalized notion is the requirement that all the agents converge to the same but *dynamic* values. This is the problem of synchronization.

A familiar example is a network of harmonic oscillators that synchronize their phases and angular velocities. Another example is a group of autonomous vehicles that flock with the same velocities. A physiology example is a network of neurons that fire with the same frequencies. Indeed the synchronization problem typically involves higher-order dynamic models of the agents.

In this chapter we study the synchronization problem of (homogeneous) linear time-invariant dynamic agents. We show that a necessary graphical condition to achieve synchronization is that the digraph contains a spanning tree (the same as that to achieve consensus). Under this condition, we present a distributed algorithm that achieves synchronization.

5.1 Problem Statement

Consider a network of n (> 1) agents. Each agent i $(\in [1, n])$ is modeled by a general linear time-invariant (LTI) dynamic system:

$$\begin{aligned} \dot{x}_i &= Ax_i + Bu_i \\ y_i &= Cx_i + Du_i \end{aligned} \tag{5.1}$$

where $x_i \in \mathbb{R}^p$ is the state vector, $u_i \in \mathbb{R}^q$ the (control) input vector, and $y_i \in \mathbb{R}^r$ the (observation) output vector. A compact graphical notation of LTI is displayed in Fig. 5.1.

The matrices A, B, C, D in (5.1) are of the following sizes:

$$A \in \mathbb{R}^{p \times p}, \quad B \in \mathbb{R}^{p \times q}, \quad C \in \mathbb{R}^{r \times p}, \quad D \in \mathbb{R}^{r \times q}.$$

These matrices are the same for all agents; thus the multi-agent system is called *homogeneous*. Several assumptions are made concerning these matrices.

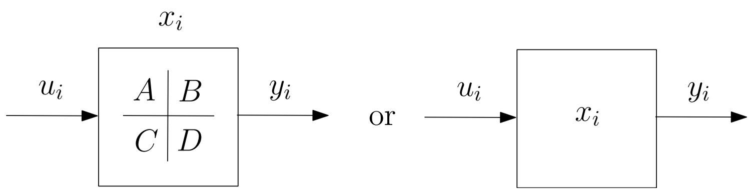

Figure 5.1: Linear time-invariant system

Assumption 5.1 *The matrices A, B, C satisfy the following conditions.*

- *(A, B) is* stabilizable*, i.e. there exists a matrix $F \in \mathbb{R}^{q\times p}$ such that all the eigenvalues of $A + BF$ have negative real parts.*
- *(C, A) is* detectable*, i.e. there exists a matrix $G \in \mathbb{R}^{p\times r}$ such that all the eigenvalues of $A + GC$ have negative real parts.*
- *All the eigenvalues of matrix A have nonpositive real parts.*

The first two assumptions are standard for the feasibility of feedback control design (see Appendix). The third condition means that the uncontrolled agent dynamics does not contain exponentially unstable modes. The reason why this last condition is needed is because we need to ensure that the rate of convergence to synchronization (determined by graph Laplacian) can dominate the possibly divergence of uncontrolled agent dynamics.

Synchronization Problem:

Consider a network of agents modeled by (5.1) interconnected through a digraph $\mathcal{G}$. Suppose that Assumption 5.1 holds. Design a distributed algorithm such that

$$(\forall x_1(0), \dots, x_n(0) \in \mathbb{R}^p)(\forall i, j \in [1, n]) \lim_{t\to\infty} (x_i(t) - x_j(t)) = 0.$$

Example 5.1 *We provide an example to illustrate the synchronization problem. Consider a network of five harmonic oscillators:*

$$\begin{aligned}\dot{x}_{i1} &= x_{i2}\\ \dot{x}_{i2} &= -x_{i1} + u_i\\ y_i &= x_{i1}, \quad i \in [1, 5].\end{aligned}$$

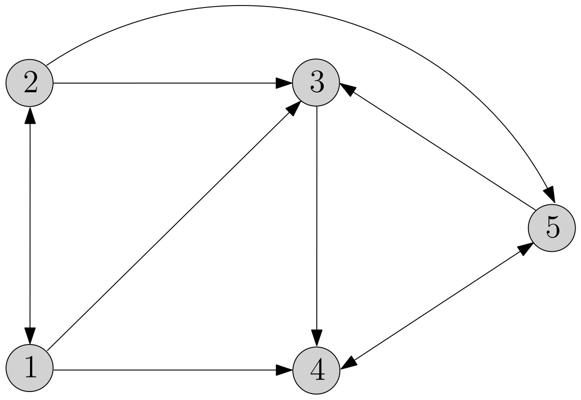

Figure 5.2: Illustrating example of synchronization problem with five agents

This corresponds to (5.1) with

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

Here x_{i1}, x_{i2} are respectively the phase angle and angular velocity of oscillator i. Since

$$\mathrm{rank}([B \ AB]) = 2$$
$$\mathrm{rank}(\begin{bmatrix} C \\ CA \end{bmatrix}) = 2$$

the pair (A, B) is controllable *and thus stabilizable, and the pair (C, A) is* observable *and thus detectable.*[a] *Moreover, the eigenvalues of A are $\pm j$ whose real parts are zero. Hence Assumption 5.1 holds.*
The interconnection of the five oscillators is modeled by the digraph in Fig. 5.2. The neighbor sets of the agents are $\mathcal{N}_1 = \{2\}$, $\mathcal{N}_2 = \{1\}$, $\mathcal{N}_3 = \{1, 2, 5\}$, $\mathcal{N}_4 = \{1, 3, 5\}$, and $\mathcal{N}_5 = \{2, 4\}$. Given arbitrary initial conditions $x_1(0), \ldots, x_5(0) \in \mathbb{R}^2$, the synchronization problem is to design a distributed algorithm such that each oscillator's phase angle (resp. angular velocity) asymptotically converges to the same dynamic phases (resp. dynamic velocities).

[a] A review of these basic concepts of LTI systems is provided in Appendix.

A necessary graphical condition for solving the synchronization problem is given below.

Proposition 5.1 *Suppose that there exists a distributed algorithm that solves the synchronization problem. Then the digraph contains a spanning tree.*

Proof. The proof is by contradiction. Suppose that the digraph $\mathcal{G}$ does *not* contain a spanning tree. Then it follows from Theorem 1.1 that $\mathcal{G}$ has at least two (distinct) closed strong components (say) $\mathcal{G}_1, \mathcal{G}_2$. In this case, consider an initial condition such that the agents in $\mathcal{G}_1$ have initial state $c_1 \in \mathbb{R}^p$, those in $\mathcal{G}_2$ have $c_2 \in \mathbb{R}^p$, and $c_1 \neq c_2$. Since $\mathcal{G}_1$ and $\mathcal{G}_2$ are closed, information cannot be communicated from one to the other. Consequently, there exists no distributed algorithm that can solve the synchronization problem. □

Owing to Proposition 5.1, we shall henceforth assume that the digraph contains a spanning tree.

Assumption 5.2 *The digraph $\mathcal{G}$ modeling the interconnection structure of the networked agents contains a spanning tree.*

5.2 Distributed Algorithm

Example 5.2 *Consider again Example 5.1. To achieve synchronization, a natural idea is to use the consensus algorithm in Chapter 4 on the output y_i $(i \in [1,5])$:*

$$u_i = \sum_{j \in \mathcal{N}_i} a_{ij}(y_j(k) - y_i(k)).$$

For simplicity consider unit weight for all edges (i.e. $a_{ij} = 1$). Then substitute the input u_i into (5.1) and write in vector form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} A - BC & BC & 0 & 0 & 0 \\ BC & A - BC & 0 & 0 & 0 \\ BC & BC & A - 3BC & 0 & BC \\ BC & 0 & BC & A - 3BC & BC \\ 0 & BC & 0 & BC & A - 2BC \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}.$$

More compactly

$$\dot{x} = (I \otimes A - L \otimes BC)x$$

where $x = [x_1^\top \ \cdots \ x_5^\top]^\top$ is the aggregated state, L is the standard Laplacian matrix, and $\otimes$ denotes Kronecker product. With a random initial condition $x(0) \in \mathbb{R}^{10}$, a simulation result *of the above system is displayed in Fig. 5.3. Evidently, synchronization did not occur. Thus the simple idea of achieving consensus on the output fails to work for synchronization.*

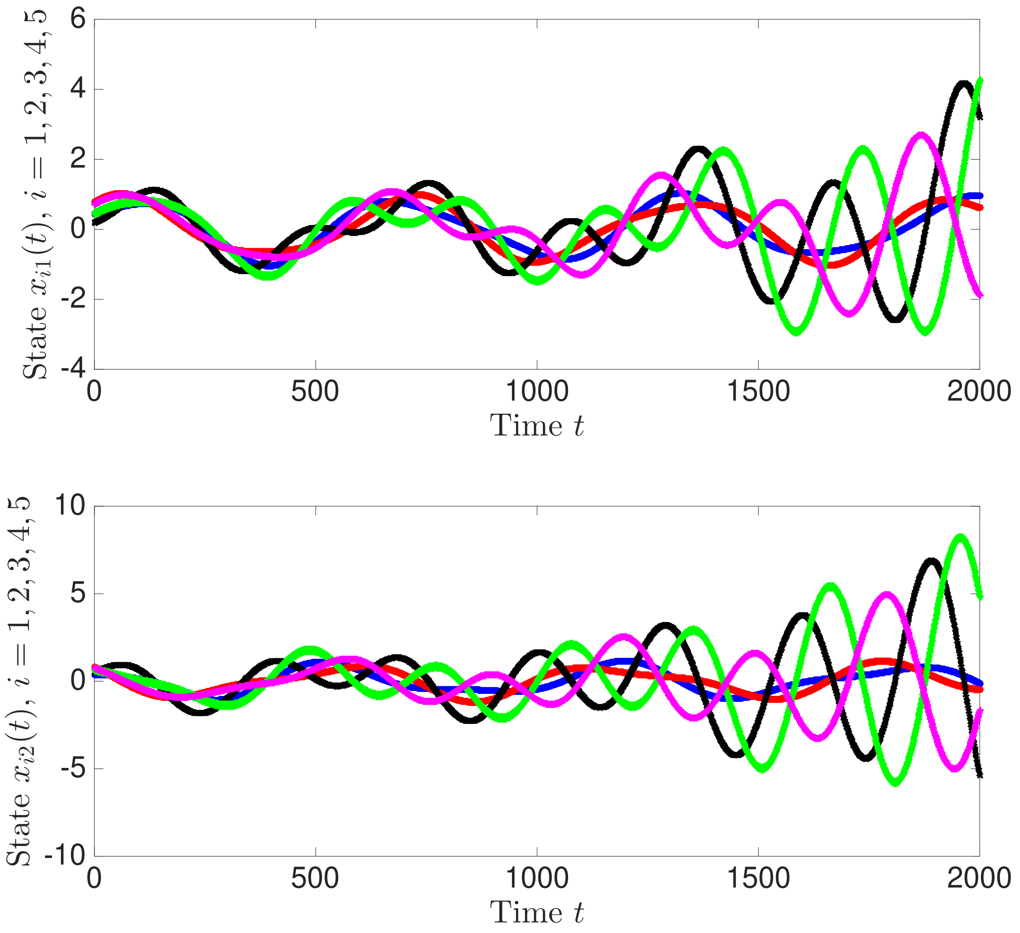

Figure 5.3: Failure to achieve synchronization using consensus algorithm

In the following, we describe a distributed algorithm that employs an *observer* that estimates the state x_i based on the output y_i, as well as a *generator* that applies the consensus algorithm based on stable dynamics.

Synchronization Algorithm (SA):

Every agent i has a dynamic model in (5.1) with an arbitrary initial state $x_i(0) \in \mathbb{R}^p$. Let F, G be matrices such that all the eigenvalues of $A+BF$ and $A+GC$ have negative real parts (such F, G exist under Assumption 5.1). At each time $t \geq 0$, every agent i performs the following updates:

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i + G(C\hat{x}_i + Du_i - y_i) \tag{5.2}$$

$$\dot{\xi}_i = (A+BF)\xi + \sum_{j \in \mathcal{N}_i} a_{ij}(\xi_j - \xi_i) - \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{x}_j - \hat{x}_i) \tag{5.3}$$

$$u_i = F\xi_i. \tag{5.4}$$

Here the *updating weights* $a_{ij} > 0$ are the weights of the edges (j, i) (i.e. the entries of the adjacency matrix); the initial conditions $\hat{x}_i(0) \in \mathbb{R}^p$ and $\xi_i(0) \in \mathbb{R}^p$ are arbitrary.

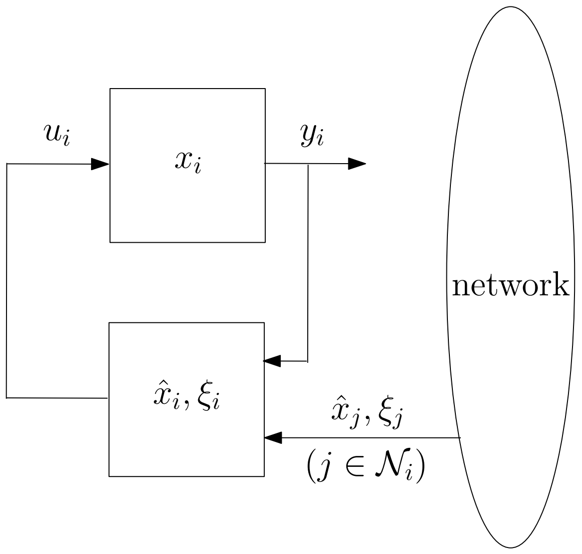

Figure 5.4: Dynamic distributed controller

Remark 5.1 *In words, (5.2) is a local observer that estimates the state x_i based on output y_i and input u_i. The observer has stable dynamics (since $A+GC$ is stable), so that the estimate $\hat{x}_i$ (exponentially) converges to the true state x_i. Next, (5.3) is a local generator also with stable dynamics (since $A+BF$ is stable). This generator executes two consensus algorithms on the generators' states and on the observers' states, for which agent i needs to receive information $\xi_j(t), \hat{x}_j(t)$ or relative information $\xi_j(t) - \xi_i(t), \hat{x}_j(t) - \hat{x}_i(t)$ from each neighbor $j \in \mathcal{N}_i$. The purpose of this generator is to achieve consensus on the generator states on one hand, and on the other hand drive the difference in generator states $\xi_j(t) - \xi_i(t)$ to the difference in estimated states $\hat{x}_j(t) - \hat{x}_i(t)$. Since the estimated states converge to the true states, the difference in any pair of*

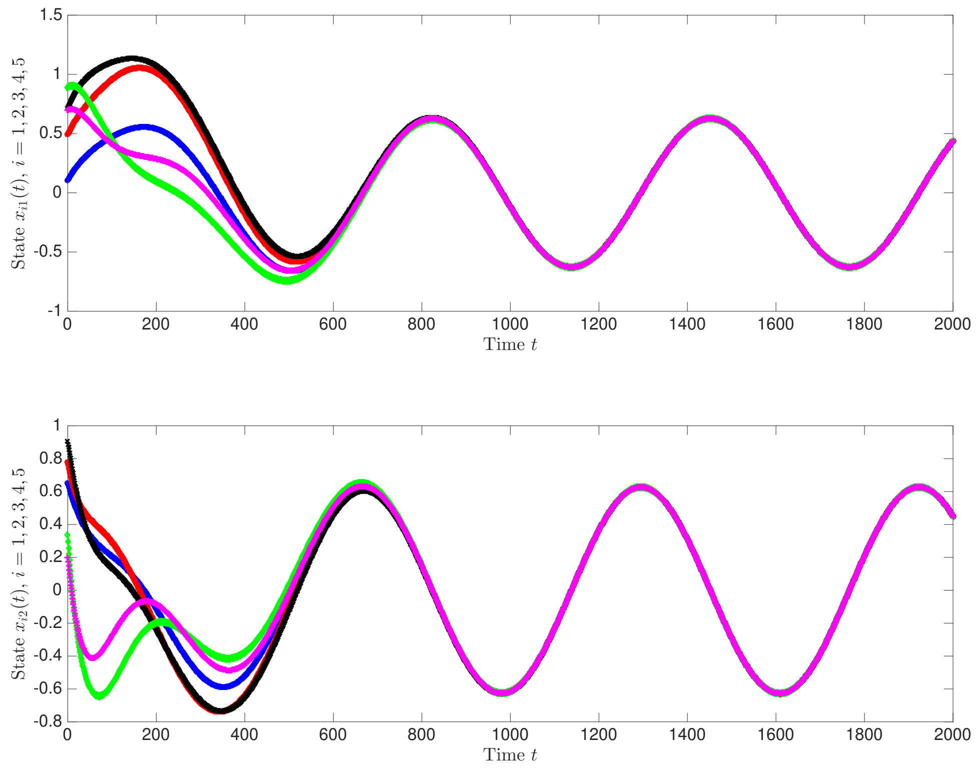

Figure 5.5: Synchronization of true states

true states will diminish, and desired synchronization occurs. Finally, (5.4) computes the control input u_i. Overall, this is a dynamic distributed controller for agent i, whose inputs are y_i (from itself) and $\hat{x}_j, \xi_j$ (from its neighbors) while the output is u_i. A graphical illustration of this dynamic distributed controller is provided in Fig. 5.4.

Remark 5.2 *If $C = I$, i.e. $y_i = x_i$, then the observer in (5.2) is not needed. Namely in this special case, SA becomes*

$$\dot{\xi}_i = (A + BF)\xi + \sum_{j \in \mathcal{N}_i} a_{ij}(\xi_j - \xi_i) - \sum_{j \in \mathcal{N}_i} a_{ij}(x_j - x_i)$$

$$u_i = F\xi_i.$$

Let

$$x := \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{np}, \quad \hat{x} := \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_n \end{bmatrix} \in \mathbb{R}^{np}, \quad \xi := \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_n \end{bmatrix} \in \mathbb{R}^{np}$$

be the aggregated true state, estimated state, and generator state of the networked agents. Then the equations (5.1), (5.2), and (5.3) become

$$\begin{aligned} \dot{x} &= (I_n \otimes A)x + (I_n \otimes BF)\xi \\ \dot{\hat{x}} &= (I_n \otimes (A+GC))\hat{x} + (I_n \otimes BF)\xi - (I_n \otimes GC)x \\ \dot{\xi} &= (I_n \otimes (A+BF) - L \otimes I_p)\xi + (L \otimes I_p)\hat{x}. \end{aligned} \tag{5.5}$$

Note that the Laplacian matrix L appears only in the last equation of the generator dynamics.

Example 5.3 *Let us revisit Example 5.2. First, we assign desired eigenvalues for $A+BF$ and $A+GC$. Say for both matrices, let the desired eigenvalues be $-1,-2$. Then by* pole assignment *(see Appendix), we obtain*

$$F = \begin{bmatrix} -1 & -3 \end{bmatrix}, \quad G = \begin{bmatrix} -3 \\ -1 \end{bmatrix}.$$

Substituting A, B, C, F, G, L into (5.5) and performing simulation with a set of random initial conditions $x(0), \hat{x}(0), \xi(0)$, we obtain the synchronized states of the oscillators as displayed in Fig. 5.5. Observe that both phase angles and angular velocities of the five oscillators converge to the same dynamic values. The estimated states also synchronize (Fig. 5.6), as they converge to the true states that are synchronized. Finally, the generator states converge to 0 (Fig. 5.7), for these generators are so designed that the difference in pairwise generator states converge to the difference in pairwise estimated states (the latter converges to 0).

5.3 Convergence Result

The following is the main result of this section.

Theorem 5.1 *Suppose that Assumptions 5.1 and 5.2 hold. Then SA solves the synchronization problem.*

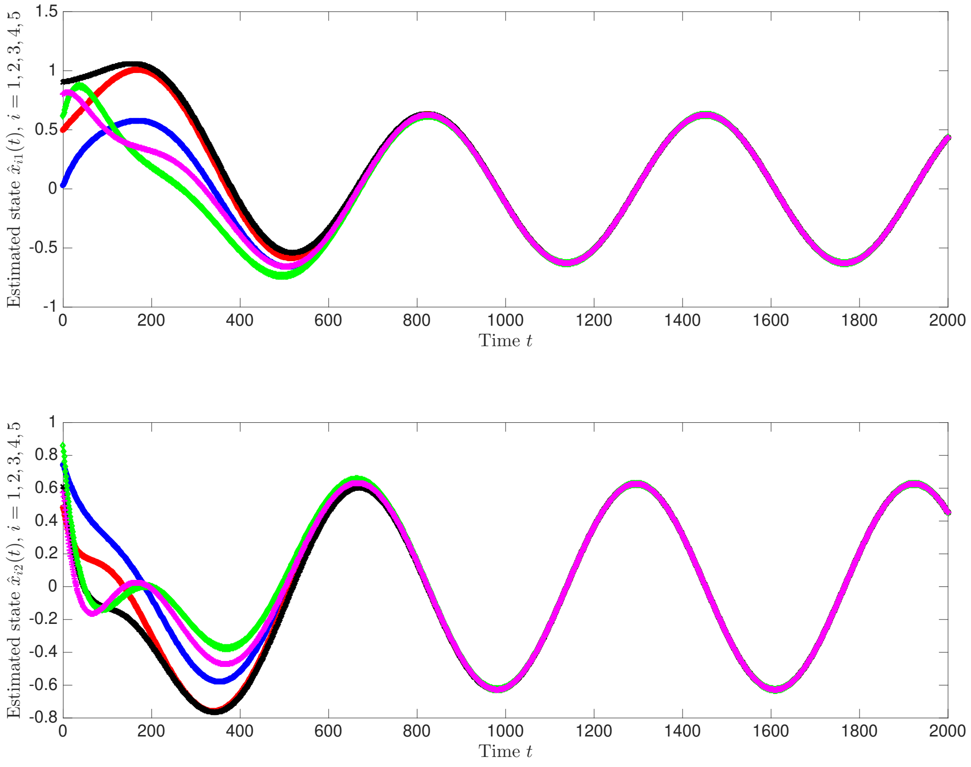

Figure 5.6: Synchronization of estimated states

To proceed, let us first consider the third equation in (5.5):

$$\begin{aligned}\dot{\xi} &= (I_n \otimes (A + BF) - L \otimes I_p)\xi + (L \otimes I_p)\hat{x} \\ &= (I_n \otimes (A + BF))\xi + (L \otimes I_p)(\hat{x} - \xi).\end{aligned}$$

Since the eigenvalues of $A + BF$ have negative real parts, the convergence of $\xi(t)$ depends on that of $(\hat{x}(t) - \xi(t))$. Let

$$\epsilon := \hat{x} - \xi.$$

Then $\dot{\epsilon} = \dot{\hat{x}} - \dot{\xi}$. Substituting $\dot{\hat{x}}, \dot{\xi}$ by the second and third equations in (5.5) and arranging the

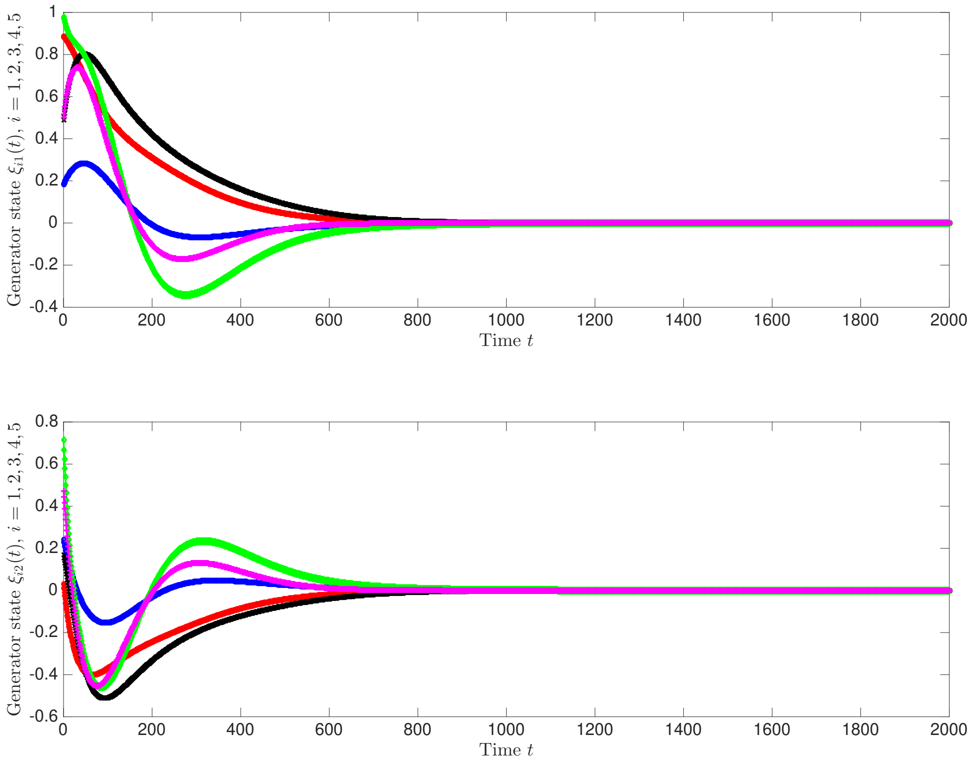

Figure 5.7: Convergence of generator states

terms yield

$$\dot{\epsilon} = (I_n \otimes A - L \otimes I_p)\epsilon - (I_n \otimes GC)(x - \hat{x}).$$

Ignoring for now the second term (i.e. the state estimation error which exponentially vanishes):

$$\dot{\epsilon} = (I_n \otimes A - L \otimes I_p)\epsilon; \tag{5.6}$$

thus corresponding to each ϵ_i ($i \in [1, n]$) is a consensus-like algorithm:

$$\dot{\epsilon}_i = A\epsilon_i + \sum_{j \in \mathcal{N}_i} a_{ij}(\epsilon_j - \epsilon_i). \tag{5.7}$$

The following lemma states that for every $i \in [1, n]$, $\epsilon_i(t)$ converges to $\epsilon_0(t)$ which is a solution of