# Consensus in networked multi-agent systems

**N.M.D. Niezink**

Faculty of Mathematics and Natural Sciences
Master Applied Mathematics

August 2011

*Supervisor:*
A.J. van der Schaft

# Contents

# Chapter 1

# Introduction

In a networked system dynamic units (agents) operate by interacting over an information exchange network. Networked systems are omnipresent in the many areas of science. Examples vary from technological and information networks to social and biological ones [15]. All these networks have in common that their structure, representing a particular pattern of interactions, usually has a big effect on the behavior of its corresponding system.

Also in system and control theory distributed coordination of multi-agent systems has a strong tradition. An important class of problems studied in this field are the *consensus problems*, for which Borkar and Varaiya [6] and Tsitsiklis [26] have layd the groundwork. The common motivation behind their work is the rich history of consensus protocols in computing science. We refer to [17] for an overview of the history of consensus problems in systems and control theory.

For a networked system of agents reaching *consensus* means to reach an agreement regarding a certain quantity of interest that depends on the states of all agents [17]. Recently, consensus problems and closely related topics have attracted attention of many researchers, which has led to a flood of publications. These include topics such as collective behavior of flocks and swarms, e.g. [16], and synchronization of coupled oscillators, e.g. [22].

Agents try to reach consensus by interaction. There exist many variants of interaction topologies for multi-agent systems. For example, the topology can be static or varying and the communication, that it represents, can be instantaneous or not [19]. Throughout this thesis we assume that the interaction topology is fixed and without communication time-delay. Moreover, we study networks in which interactions between agents can be asymmetric. Results for symmetric interactions easily follow from their asymmetric counterparts.

Graphs form a natural representation of interaction topologies. Undirected graphs model symmetric communication, while directed graphs model assymetric communication. As such, consensus problems lie at the intersection of systems and control theory and graph theory. In particular, the analysis of consensus problems relies heavily on properties of the graph Laplacian, as studied in algebraic graph theory. The fact that the latter is less developed for directed graphs adds a challenge to the analysis of consensus problems on asymmetric-interaction topologies.

The remainder of this thesis consists of three parts. In Chapter 2 we introduce the notion of consensus algorithms and show stability and convergence criteria for these. Instead of giving all agents the same roles we can also assign leader and follower rules to various agents. We study the consequences of this choice in Chapter 3. Finally, in Chapter 4 we derive conditions on passive systems to reach output synchronization, and we study a physical example.

## 1.1 Notation

Throughout this thesis we will adopt the following notation. An $n \times m$ matrix is a matrix having $n$ rows and $m$ columns; $A^T$ denotes transpose of $A$ and $A^{-1}$ its inverse (if $A$ is nonsingular). The class of real $n \times m$ matrices is denoted by $\mathbb{R}^{n\times m}$ and the class of real $n \times 1$ vectors by $\mathbb{R}^n$.

Vectors $0_n$ and $1_n$ denote the all-zero and all-one $n \times 1$ vectors, respectively. The $n \times m$ all-zero matrix is denoted by $0_{n\times m}$ and the $n \times n$ identity matrix by $I_n$.

Let $v_1, \ldots, v_n \in \mathbb{R}^m$. Then $\mathrm{col}(v_1, \ldots, v_n)$ is the $nm \times 1$ vector that stacks $v_1, \ldots, v_n$ one underneath the other. We let $\mathrm{diag}(a_1, \ldots, a_n)$ denote the matrix in $\mathbb{R}^{n\times n}$ with diagonal entries $a_1, \ldots, a_n \in \mathbb{R}$ and zero off-diagonal entries.

The Kronecker product of $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$ is denoted $A \otimes B$ and is a $mp \times nq$ matrix defined by

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}.$$

Some basic properties of the Kronecker product are:

$$\begin{aligned} (A+B) \otimes C &= A \otimes B + B \otimes C, \\ (A \otimes B)^T &= A^T \otimes B^T, \\ (A \otimes B)(C \otimes D) &= AC \otimes BD. \end{aligned}$$

Of course, the products $AC$ and $BD$ must exist for the last identity to hold.

# Chapter 2

# State consensus

In this chapter we will discuss multi-agent systems, in which the agents are intended to reach consensus on their states. To define what it means to reach consensus, we consider a system of $n$ agents and let $x_i \in \mathbb{R}^d$ denote the state of agent $i$. Agents are said to reach *(state) consensus* if they converge to the same value, that is, for all initial states $x_i(0) \in \mathbb{R}^d$ and all $i, j \in V$,

$$|x_i(t) - x_j(t)| \to 0, \qquad \text{as } t \to \infty.$$

In case the consensus value is equal to the average of the initial states of all agents, the agents are said to reach *average consensus*.

We assume that the agents in the system operate by interacting over a network. The *consensus algorithm* (or protocol) is an interaction rule that specifies the information exchange between an agent and all of its neighbours in this network. We will elaborate on the consensus algorithm in Section 2.1.

Whether the consensus algorithm causes agents to reach consensus depends on their interaction topology. As mentioned in the introduction, an interaction topology can be represented by a graph. In such a graph vertices correspond to agents and edges to the interactions between them. We assume that the interactions between agents can be asymmetric. Therefore, we consider directed graphs $G = (V, E)$ with vertex set $V = \{1, \ldots, n\}$ and an edge set $E \subseteq V \times V$ consisting of ordered pairs of distinct vertices [19]. In Section 2.2 we will derive necessary and sufficient conditions on $G$ for the agents to reach (average) consensus. All results on directed graphs can be extended straightforwardly to undirected graphs (symmetric communication), because with any undirected graph a directed one can be associated (see Appendix A).

In Section 2.3 we study the Laplacian matrix. The Laplacian is a matrix representation of a graph and many properties of a graph can be characterized in terms of properties of its corresponding Laplacian. Moreover, the Laplacian matrix is closely related to the standard consensus algorithm. We introduce a procedure to partition this matrix that allows us to express the final state of agents abiding the consensus algorithm in closed form.

## 2.1 Consensus algorithm

Since the agents we study are embedded in a network, they form a system that is inherently distributed. Let the set of *neighbours* of agent $i$ be defined by $N_i = \{j \in V : (j,i) \in E\}$. The *distributed consensus algorithm*, which is central in this thesis, takes into account the distributed nature of the multi-agent system and is defined as

$$\dot{x}_i(t) = \sum_{j \in N_i} (x_j(t) - x_i(t)), \qquad i \in V. \tag{2.1}$$

According to these dynamics, an agent adapts its state solely based one the states of its neighbours, in such a way that its state moves in the direction of the average of the neighbours' states. Intuitively, this implies that the states of the agents will always converge to some bounded solution. The following lemma formalizes this intuition.

**Lemma 2.1.** *Let $\Omega_0$ denote the convex hull of the initial states of the agents, then $x_i(t) \in \Omega_0$ for all $t \geq 0$ and $i \in V$ [11].*

*Proof.* If $|N_i| = 0$, agent $i$ will not move, so $x_i(t) \in \Omega_0$ for all $t \geq 0$. Let $\Omega_t$ denote the convex hull of the states of the agents at time $t$. We will show that in case $|N_i| > 0$ the trajectory $x_i(t)$ will be on the boundary of $\Omega_t$ or is pointing inwards $\Omega_t$. To this order we first rewrite consensus algorithm (2.1) a bit, obtaining

$$\frac{1}{|N_i|}\dot{x}_i(t) = -x_i(t) + \frac{1}{|N_i|} \sum_{j \in N_i} x_j(t).$$

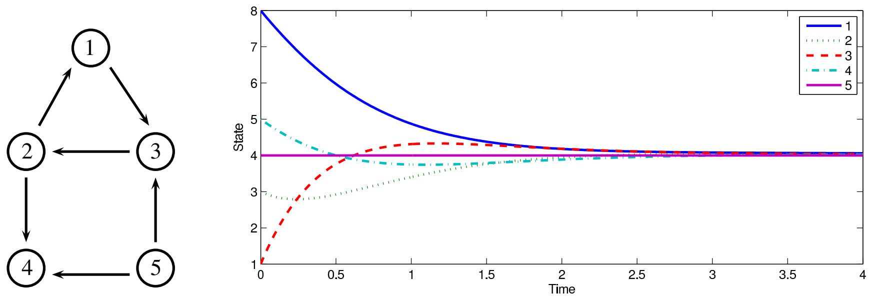
This shows that the motion of agent $i$ is always directed towards the mean of the states of its neighbours, which lies in $\Omega_t$. By convexity of $\Omega_t$ we conclude that the motion of agent $i$ lies within $\Omega_t$. This implies that no agent will ever leave $\Omega_0$. □

Since the solution to system with dynamics 2.1 is bounded for any choice of initial state of the agents, this system is stable by the definition of stability (see Section B.2.1). In Section 2.3.4 the stability of this system is derived in a different way.

### 2.1.1 Consensus algorithm in terms of Laplacian matrix

Consensus algorithm (2.1) can be formulated more concisely in terms of the Laplacian matrix. The Laplacian matrix $L$ of graph $G$ is defined as

$$l_{ij} = \begin{cases} -1 & \text{if } j \in N_i, \\ |N_i| & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$




(a) (b)

Figure 2.1: Graph $G$ representing the interaction topology of the agents in Example 2.2 (a) and the trajectories of these agents (b).

Note that this definition of the Laplacian is equivalent to the one introduced in Appendix A. If we suppose that each undirected edge can be represented by two oppositely oriented directed edges, the Laplacian matrix of an undirected graph is also defined as in (2.2). The Laplacian of an undirected graph is always symmetric.

Let $x = \mathrm{col}(x_1, \dots, x_n) \in \mathbb{R}^{nd}$ denote the vector containing the states of all agents. Consensus algorithm (2.1) can now be reformulated as

$$\dot{x}(t) = -(L \otimes I_d)x(t). \tag{2.3}$$

**Example 2.2.** Consider five agents with states in $\mathbb{R}$ and let their initial states be given by $x(0) = \mathrm{col}(8, 3, 1, 5, 4)$. Their interaction topology is represented by graph $G$ in Figure 2.1a. The Laplacian matrix corresponding to this graph is

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & 0 & 2 & 0 & -1 \\ 0 & -1 & 0 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Figure 2.1b shows the trajectories of the agents. All agents converge to value 4, so they reach consensus.

By definition, the Laplacian $L$ has zero row sums, that is, $L1_n = 0_n$. Therefore, $L$ has at least one zero eigenvalue and $1_n$ is a corresponding eigenvector. From the particular structure of the Laplacian we can derive the following result on its eigenvalues in general.

**Lemma 2.3.** *The eigenvalues of a Laplacian matrix are either zero or lie in the open right half plane.*

*Proof.* This statement is a direct consequence of the Gersgorin's disc theorem (see Appendix B). According this theorem all eigenvalues of $L$ lie in at least one of the closed discs in the complex plane, centered at $l_{ii}$ and with radius $R_i = \sum_{j \neq i} |l_{ij}| = l_{ii}$, $i = 1, \dots, n$. In a Laplacian matrix by definition $l_{ii} \geq 0$, so all these discs lie in the closed right half plane. Moreover, the only eigenvalue that can have zero real part is zero itself. □

### 2.1.2 Generalization

All the results that will be presented in this thesis can be generalized to weighted directed graphs. In these graphs a weight $a_{ij}$ is assigned to each edge $(i, j)$. A non-existent edge $(i, j)$ corresponds here to zero weight $a_{ij}$. The weight of a loop $(i, i)$ in a graph is given by $a_{ii}$, but since we do not allow loops we assume that $a_{ii} = 0$ [19]. The in-degree and the out-degree of vertex $i$ in a weighted graph are respectively defined as

$$d_{in}(i) = \sum_{j=1}^{n} a_{ji} \quad \text{and} \quad d_{out}(i) = \sum_{j=1}^{n} a_{ij}.$$

The consensus algorithm on weighted directed graphs is given by

$$\dot{x}_i(t) = \sum_{j=1}^{n} a_{ji}(x_j(t) - x_i(t)), \qquad i \in V.$$

Like consensus algorithm (2.1), these dynamics can equivalently be formulated using the Laplacian matrix. The Laplacian $L$ of a weighted directed graph is defined as

$$l_{ij} = \begin{cases} -a_{ij} & \text{if } j \neq i, \\ d_{in}(i) & \text{if } j = i. \end{cases}$$

## 2.2 Consensus conditions

In the system described in Example 2.2 the agents reached consensus. Imagine, however, what would have happened if the graph would have consisted out of two unconnected components. In this case the agents from one component would not have had access to the states of any of the agents in the other component. Intuitively, this prevents the agents from reaching consensus.

In this section we derive what condition the graph representing the interaction topology of the agents has to satisfy such that the agents will reach consensus. Throughout the remainder of this chapter we assume that the agents have a one-dimensional state, that is, $d = 1$. The results that are derived can be generalized straightforwardly.

In this section we will gradually work towards the main result, Theorem 2.6, that states that a network of agents reaches consensus under algorithm (2.3) if and only if their interaction topology has a directed spanning tree. To prove this theorem we follow the ideas provided in [22], but reformulate them for the sake of clarity.

We will show that a graph has a directly spanning tree if and only if its corresponding Laplacian has exactly one zero eigenvalue (Lemma 2.4). Since all non-zero eigenvalues of $-L$ lie in the open left half plan (Lemma 2.3), this latter property implies that $e^{-Lt}$ converges to a matrix with properties that are suitable for consensus (Lemma 2.5). In the proof of Theorem 2.6 these results will be combined.

**Lemma 2.4.** *The Laplacian has exactly one zero eigenvalue if and only if its corresponding graph $G$ has a directed spanning tree [22].*

*Proof.* [$\Rightarrow$] Suppose graph $G$ does not have a directed spanning tree, then the condensation $G_c = (V_c, E_c)$ of $G$ does not have a directed spanning tree either. Since $G_c$ is acyclic, we can find a vertex $v \in V_c$ that has no predecessors. Let set $A \subset V$ denote the vertices in $V$ that correspond to vertex $v \in V_c$. Let $W \subset V_c$ contain those vertices that are reachable from $v$ in $G_c$ and let $C \subset V$ contain the corresponding vertices in $V$. Finally, let $B = V \setminus (A \cup C)$. This set is not empty, since $G_c$ has no directed spanning tree, so there are vertices in $V_c$ that are not reachable from $v$. Renumber the vertices in their order of appearance in consecutively $A$, $B$ and $C$. The Laplacian matrix of $G$ then is partitioned as follows:

$$L = \begin{pmatrix} L_A & & \\ & L_B & \\ L_{ac} & L_{bc} & L_C \end{pmatrix}.$$

Since $L1_n = 0_n$, we also have that $L_A 1_{|A|} = 0_{|A|}$ and $L_B 1_{|B|} = 0_{|B|}$, so both $L_A$ and $L_B$ have a zero eigenvalue. Matrix $L$ is a lower diagonal block matrix and $L_A$, $L_B$ and $L_C$ are square matrices, so

$$\det(L - \lambda I_n) = \det(L_A - \lambda I_{|A|}) \det(L_B - \lambda I_{|B|}) \det(L_C - \lambda I_{|C|}).$$

This implies that the algebraic multiplicity of eigenvalue zero in $L$ is at least two.

[$\Leftarrow$] A Laplacian has at least one zero eigenvalue. We will prove by induction over the number of edges $m$ of a graph with $n$ vertices that there exists only one such eigenvalue, if $G$ has a directed spanning tree. First suppose $G$ is itself a directed spanning tree ($m = n - 1$). Let vertex 1 be the root vertex and give the vertices at distance 1 from the root numbers $2, \ldots, q_1$, those at distance 2 numbers $q_1 + 1, \ldots, q_2$, etc. In this way the Laplacian $L$ of $G$ satisfies $l_{11} = 0$ and $l_{ii} > 0$ for $i = 1, \ldots, n$. Since $G$ is a tree and thus $(j, i) \in E$ only if $j < i$, $L$ is a lower triangular matrix. Therefore, the eigenvalues of $L$ are equal to its diagonal entries and $L$ has only one eigenvalue equal to 0.

Now suppose that for a graph $G$ on $n$ vertices with $m$ edges, for a certain $m \geq n-1$, the following holds: if $G$ has a directed spanning tree, then the Laplacian of $G$ has exactly one zero eigenvalue. We will show that this statements is also true for graphs with $m + 1$ edges.

Let $\tilde{G}$ be a graph on $n$ vertices with $m + 1$ edges that contains a spanning tree. Since a spanning tree always consists of $n - 1$ edges, we can remove one edge from $\tilde{G}$ in such a way that remaining graph $G$ still has a spanning tree. Assume without loss of generality that we remove edge $(k, 1)$.

Let $\tilde{L}$ and $L$ denote the Laplacians of $\tilde{G}$ and $G$ respectively. We will prove that $\tilde{L}$ has exactly one zero eigenvalue, by showing this for $-\tilde{L}$. Define $M_\lambda = \lambda I_n + M$ for $M \in \mathbb{R}^{n \times n}$. By solving $\det(M_\lambda) = 0$ for $\lambda$ we obtain the eigenvalues of matrix $-M$. In the remainder of this proof we first relate $\det(L_\lambda)$ to $\det(\tilde{L}_\lambda)$ and then use this relation in combination with the Routh-Hurwitz stability criterion (see Section B.2.1) to show that $\tilde{L}$ has exactly one zero eigenvalue.

Since $G$ was obtained from $\tilde{G}$ by the removal of edge $(1, k)$, matrix $\tilde{L}_\lambda$ is equal to $L_\lambda$ except for entries

$$\begin{aligned} (\tilde{L}_\lambda)_{11} &= \lambda + \tilde{l}_{11} = \lambda + l_{11} + 1 = (L_\lambda)_{11} + 1, \\ (\tilde{L}_\lambda)_{1k} &= \tilde{l}_{1k} = l_{1k} - 1 = (L_\lambda)_{1k} - 1. \end{aligned}$$

Let $M([i,j])$ denote the submatrix of $M$ formed by deleting its $i$th row and $j$th column. Evaluating the determinant of $\tilde{L}_\lambda$ by expansion along the first row yields

$$\det(\tilde{L}_\lambda) = \det(L_\lambda) + \det(L_\lambda[1,1]) + (-1)^k \det(L_\lambda[1,k]). \tag{2.4}$$

The Laplacian of the graph induced by vertex set $V' = \{2, \dots, n\}$ is given by

$$E = \begin{pmatrix} \tilde{l}_{22} & \tilde{l}_{23} & \cdots & \tilde{l}_{2k} + \tilde{l}_{21} & \cdots & \tilde{l}_{2N} \\ \tilde{l}_{32} & \tilde{l}_{33} & \cdots & \tilde{l}_{3k} + \tilde{l}_{31} & \cdots & \tilde{l}_{3N} \\ \vdots & \vdots & & \ddots & & \vdots \\ \tilde{l}_{N2} & \tilde{l}_{N3} & \cdots & \tilde{l}_{Nk} + \tilde{l}_{N1} & \cdots & \tilde{l}_{NN} \end{pmatrix}.$$

Expanding the determinant of $E_\lambda$ along the $(k-1)$th column of $E_\lambda$, we find that

$$\det(E_\lambda) = \det(\tilde{L}_\lambda[1,1]) + (-1)^k \det(\tilde{L}_\lambda[1,k]). \tag{2.5}$$

We combine equations (2.4) and (2.5) to obtain the following relation between $\det(\tilde{L}_\lambda)$ and $\det(L_\lambda)$:

$$\det(\tilde{L}_\lambda) = \det(L_\lambda) + \det(E_\lambda). \tag{2.6}$$

Both $L$ and $E$ are Laplacian matrices, so they have at least one zero eigenvalue and we can write

$$\begin{aligned} \det(L_\lambda) &= \lambda^n + a_{n-1}\lambda^{n-1} + \ldots + a_1\lambda, \\ \det(E_\lambda) &= \lambda^{n-1} + b_{n-2}\lambda^{n-2} + \ldots + b_1\lambda. \end{aligned}$$

According to Lemma 2.3 all eigenvalues of a Laplacian matrix lie in the closed right half plane, so the eigenvalues of $-L$ and $-E$ lie in the closed left half plane. With the Routh-Hurwitz stability criterion it follows that $a_1, b_1 \geq 0$. Moreover, since by assumption $L$ has exactly one zero eigenvalue, we have that $a_1 > 0$. This implies that $a_1 + b_1 > 0$, so from equality (2.6) it follows that $\det(\tilde{L}_\lambda)$ can have only one zero root and $\tilde{L}$ has exactly one zero eigenvalue. □

Note that for the proof of necessity we deviated from [22], since the argument there was based on the statement that if $G$ does not have a spanning tree, then

> " . . . there exists a vehicle that separates two subgroups that do not exchange information or there exist at least two vehicles that do not receive any information from their neighbors. "

No proof of this statement was included in [22] and it is not that straightforward to see that these are the only two cases that can arise when $G$ does not have a directed spanning tree.

**Lemma 2.5.** *If a Laplacian matrix $L$ has a simple zero eigenvalue, then $e^{-Lt} \to 1_n v^T$ for $t \to \infty$, where $v$ is nonnegative and satisfies $1_n^T v = 1$ and $L^T v = 0$ [22].*

*Proof.* Let $-L = PJP^{-1}$, where $J$ is the Jordan normal form of $L$. The diagonal entries of $J$ are the eigenvalues of $-L$ and $P$ contains the corresponding (generalized) eigenvectors. From the decomposition of $-L$ it follows that $e^{-L} = Pe^{J}P^{-1}$.

Assume without loss of generality that the upper left entry of $J$ is zero and let the first column of $P$ contain $1_n$ as corresponding eigenvector. Since $L$ has exactly one zero eigenvalue and the others lie in the closed right half plain (see Lemma 2.3), for $t \to \infty$ we have

$$e^{Jt} \to \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Consequently, $e^{-Lt} = Pe^{Jt}P^{-1} \to 1_n v^T$, as $t \to \infty$, where $v^T$ equals the first row of $P^{-1}$. Moreover, $e^{-Lt}1_n = e^{0\cdot t}1_n = 1_n$ (see Lemma B.3). This is true for all $t$, so in particular for $t \to \infty$ we have $1_n v^T 1_n = 1_n$ and $v^T 1_n = 1_n^T v = 1$.

It only remains to be shown that $L^T v = 0_n$ and $v \geq 0$. From the Jordan matrix decomposition of $-L$ it holds that $L^T v = -(P^{-1})^T J^T P^T v$. Since $v^T$ is the first row of $P^{-1}$, we have

$$P^T v = (v^T P)^T = (1\,0\,\cdots\,0)^T.$$

Moreover, all entries in the first row and first column of $J$ are zero, so

$$L^T v = -(P^{-1})^T J^T (1\,0\,\cdots\,0)^T = -(P^{-1})^T 0_n = 0_n.$$

With this we get that $e^{L^T} v = e^0 v = v$ (see Lemma B.3). Matrix $e^{L^T}$ is positive and its largest eigenvalue is 1, so Lemma B.2 implies that $e^{L^T} x = x$ for some $x > 0$. Since eigenvalue 1 is simple, $v = \alpha x$. Earlier we proved that $1_n^T v = 1$ and consequently $\alpha > 0$, so $v \geq 0$. □

**Theorem 2.6.** *Agents abiding algorithm* (2.1) *reach consensus if and only if $G$ has a directed spanning tree. In particular, the consensus value is* $v^T x(0)$, *where $v$ is nonnegative and satisfies $1_n^T v = 1$ and $L^T v = 0$ [22].*

*Proof.* [⇐] The solution of (2.1) is given by $x(t) = e^{-Lt}x(0)$. If $G$ has a directed spanning tree, its Laplacian $L$ has exactly one zero eigenvalue (Lemma 2.4). Lemma 2.5 then implies that for $t \to \infty$, $x(t) \to 1_n v^T x(0)$, where $v$ is nonnegative and satisfies $1_n^T v = 1$ and $L^T v = 0$. So for $i = 1, \ldots, n$ the limit of $x_i(t)$ is $v^T x(0)$.

[⇒] Suppose $G$ does not have a directed spanning tree. Then, according to necessity part of the proof of Lemma 2.4, the Laplacian can be structured as follows:

$$L = \begin{pmatrix} L_A & 0 & 0 \\ 0 & L_B & 0 \\ L_{ac} & L_{bc} & L_C \end{pmatrix}.$$

Now if we take the initial values of agents $1, \ldots, |A|$ bigger than 0 and the others less than 0, Lemma 2.1 implies that for all $t > 0$, $x_i(t) > 0$ for $i = 1, \ldots, |A|$ and $x_i(t) < 0$ for $i = |A|+1, \ldots, N$. Therefore, consensus can never be reached. □

For the necessity part of the proof we deviated from the proof given in [22]. There a proof by contradiction is used, that is, it is assumed that algorithm (2.1) achieves consensus asymptotically but that $G$ does not have a directed spanning tree. The argument continues as follows:

> " Then there exist at least two vehicles $i$ and $j$ such that there is no path in $\mathcal{G}_n$ that contains both $i$ and $j$. Therefore it is impossible to bring data between these two vehicles into consensus, which implies that consensus cannot be achieved asymptotically. "

This argument suggests that 'bringing data between any two vehicles into consensus' is something formal. We could similarly state that that the existence of a directed spanning tree is sufficient for reaching consensus, because then for any pair of vertices $i$ and $j$ there exists a path that contains both $i$ and $j$, so 'data between these two can be brought into consensus'. However, the proofs of Lemma 2.4 and Lemma 2.5 show that this is not that straightforward.

For algorithm (2.1) to achieve average consensus one extra constraint on graph $G$ is necessary: $G$ needs to be balanced. In terms of the Laplacian of $G$ this means that $L^T 1_n = 0_n$. Theorem 2.7 states this result.

**Theorem 2.7.** *Agents abiding algorithm* (2.1) *reach average consensus if and only if* $G$ *has a directed spanning tree and is balanced [22].*

*Proof.* [$\Rightarrow$] Suppose protocol (2.1) achieves average consensus. Theorem 2.6 implies that if consensus is reached, $G$ has a directed spanning tree. It also states that the consensus equilibrium is of the form $v^T x(0)$, where $v$ satisfies $1_n^T v = 1$ and $L^T v = 0$. In this case the equilibrium is $\frac{1}{n} 1_n^T x(0)$, so $v = \frac{1}{n} 1_n$ and $L^T 1_n = 0_n$. Hence, $G$ is balanced.

[$\Leftarrow$] Suppose $G$ has a directed spanning tree and is balanced. According to Theorem 2.6, protocol (2.1) achieves consensus and the consensus equilibrium is $v^T x(0)$, where $v$ satisfies $1_n^T v = 1$ and $L^T v = 0_n$. Since $G$ is balanced, $L^T 1_n = 0$. Matrix $L^T$ has exactly one zero eigenvalue (see Lemma 2.4), so $v = \alpha 1_n$. Given that $1_n^T v = 1$, we find that $\alpha = \frac{1}{n}$, so the equilibrium is the average of the initial states. □

Note that in the literature on consensus algorithms the condition of having a spanning tree, is often replaced by being strongly connected. Corollary A.2 shows that these two conditions are equivalent, if the graph under consideration is balanced.

The conditions under which dynamics (2.1) causes agents with a symmetric interaction topology to reach consensus can be derived from Theorem 2.6 and Theorem 2.7. In this case the graph $G$ representing the interaction topology is undirected and the corresponding Laplacian $L$ is a symmetric matrix. Therefore, $L^T 1_n = L 1_n = 0_n$ and if consensus is reached, the consensus value will always be the average of the initial states of all agents (Theorem 2.7). Moreover, (average) consensus is reached if and only if $G$ is connected (Theorem 2.6).

## 2.3 Laplacian of a directed graph

Let $G = (V, E)$ be a directed graph on $n$ vertices and $L$ the corresponding Laplacian. In [17] it is stated that if $G$ has $c$ strongly connected components, then $\text{rank}(L) = n - c$. This is not true, as can be inferred from the following example.

**Example 2.8.** Consider graph $G$ in Figure 2.2. The Laplacian matrix of this graph is given by

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix}.$$

Graph $G$ has two strongly connected components, the subgraphs induced by vertex sets $V_1 = \{1, 2\}$ and $V_2 = \{3, 4\}$, so the formula given in [17] implies that the rank of $L$ is 2. However, the actual rank of $L$ is 3.

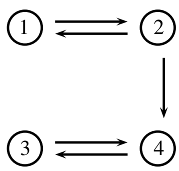



Figure 2.2: Counterexample to statement from [17].

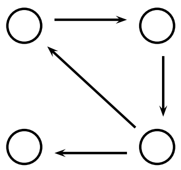
This mistake was already reported in [7]. The authors of [17] mentioned in [18] in return that the misunderstanding is due to a misinterpretation of the definition of strongly connected components of a graph used in [17]. However, it is in fact possible to express the rank of $L$ in terms of characteristics of $G$. In [1] this was shown, but to this end many non-standard graph theoretical concepts were introduced.

In the subsequent, similar results are shown by the combination of a convenient partitioning of the Laplacian matrix (Section 2.3.2) with the approach used in the proof of Lemma 2.4. The idea behind the partitioning is mainly based on some common graph theoretical structures, that are introduced in Section 2.3.1. In Section 2.3.4 we discuss the consequence of the partitioning for consensus theory, partly based on the more general results obtained in Section 2.3.3.

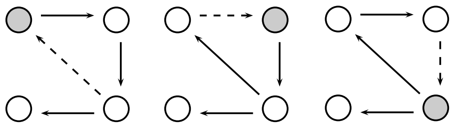
### 2.3.1 Spanning forest of a directed graph

Let $T_1, \ldots, T_k$ be disjunct subgraphs of $G$ that are directed trees[1]. If together they cover all vertices in $G$, then we say that these trees *span* $G$ and $\{T_1, \ldots, T_k\}$ is called a *spanning forest* of $G$. In each tree $T_i$ we can point out a root vertex, from which all other vertices in $T_i$ can be reached. When

[1] We will leave out 'directed' from here on, if no confusion can arise.



(a) A graph $G$



(b) Spanning trees of $G$

Figure 2.3: The root of a spanning tree is not necessarily uniquely defined.

there exists a tree on a graph, the root of this tree is not necessarily uniquely determined. Consider for example Figure 2.3, where three vertices are eligible for being the root of a directed spanning tree on $G$. In general, let the set of vertices that are eligible for being the root of tree $T_i$ be denoted by $R_i$. We call this the *root set* of tree $T_i$. Since there exists a directed path between any two vertices in a root set, this set is strongly connected. The root sets of a graph can be specified as follows.

**Definition 2.9.** A set of vertices $R$ is a *root set* of a graph $G = (V, E)$ if it induces a strongly connected graph on $G$ and there are no $(v, w) \in E$ such that $v \notin R$ and $w \in R$.

It follows from this definition that in the condensation of a graph, the root sets are represented by vertices with an empty predecessor set. Moreover, the number of root sets of $G$ is a lower bound on the cardinality of a spanning forest of $G$.

The concept of a root set is not a standard one. However, we prefer it over the term 'undominated knot' or 'source knot', used in [1] and [8] respectively, since a knot in general refers to an embedding of a circle in $\mathbb{R}^3$. For sake of completeness, Table 2.1 relates the terminology used in this chapter to the terminology used in [1].

| Used here | Used in [1] |
|---|---|
| Directed tree | Diverging tree |
| Spanning forest | Spanning diverging forest |
| Smallest spanning forest | Maximum out forest |
| Cardinality of a smallest spanning forest | Forest dimension |
| With empty predecessor set | Undominated |
| Root set | Undominated knot |

Table 2.1: Comparison of terminology.

### 2.3.2 Partitioning the Laplacian

Using the graph theoretical concepts introduced in the previous section, we propose the following procedure to renumber the vertices of a graph $G = (V, E)$ in such a way that the corresponding Laplacian has a simple block structure. This procedure also allows us to determine a smallest spanning forest of $G$.

**Step 1.** Find all root sets $R_1, \ldots, R_k$ of $G$.

**Step 2.** Find the sets $R_i^+$ of vertices, that are reachable from $R_i$, but not from $R_j$ for $j \neq i$.

**Step 3.** Define $Q = V \setminus \cup_{i=1}^{k}(R_i \cup R_i^+)$. Let $Q_i$ contain those vertices from $Q \setminus \cup_{j=1}^{i-1} Q_j$ that are reachable from $R_i$, but not from $R_j$ for $j = i+1, \ldots, k$.

**Step 4.** Renumber the vertices by their order of appearance in consecutively $R_1$, $R_1^+$, $R_2$, $R_2^+$,..., $R_k$, $R_k^+$, $Q_1$,...,$Q_k$.

Step 1 and 2 of this procedure were already proposed in [1]. The following example illustrates the procedure and will be referred to throughout the remainder of this chapter.

**Example 2.10.** Consider graph $G$ represented in Figure 2.4a.

*Step 1.* The condensation of $G$ is shown in Figure 2.4b. By looking at the vertices in the condensation that have no predecessors, we find that the root sets of $G$ are $R_1 = \{1, 2, 3\}$, $R_2 = \{5\}$, $R_3 = \{11\}$, $R_4 = \{12\}$ and $R_5 = \{16\}$.

*Step 2.* Vertex 4 is only reachable from vertices in $R_1$, so it is contained in $R_1^+$. Vertex 14 is reachable from both $R_4$ and $R_5$, so it is not contained in any of the $R_i^+$. Table 2.2 shows the sets $R_i^+$ corresponding to the sets $R_i$.

*Step 3.* For graph $G$ the set of vertices that is reachable from more than one root set is $Q = \{8, 9, 10, 14, 15\}$ (see Figure 2.4c). Set $Q_1 \subseteq Q$ should contain those vertices that are reachable from $R_1$, but not from $R_2$, $R_3$, $R_4$ or $R_5$. There are no such vertices, so $Q_1 = \emptyset$. Table 2.2 shows all sets $Q_i$.

*Step 4.* The renumbering of the vertices, according to the rule in the last step, is given in Figure 2.4d. We will refer to this new graph as $G'$.

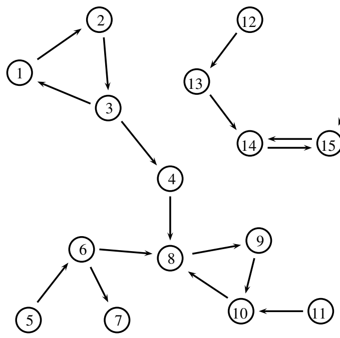
Define $n_i = |R_i \cup R_i^+|$ for $i = 1, \ldots, k$ and $n_Q = |Q|$. By construction, no vertex in $R_i \cup R_i^+$ is reachable from any vertex in $R_j \cup R_j^+$ for $i \neq j$. Consequently the Laplacian matrix is a lower triangular block matrix of the following form:

$$L = \begin{pmatrix} L_1 & & & & \\ & L_2 & & & \\ & & \ddots & & \\ & & & L_k & \\ A_1 & A_2 & \cdots & A_k & L_Q \end{pmatrix}, \tag{2.7}$$

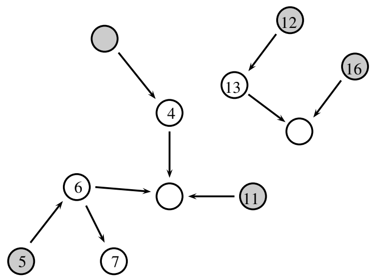
where the $n_i \times n_i$ blocks $L_i$ are equal to the Laplacian matrices of the subgraphs of $G$ induced by the vertices in $R_i \cup R_i^+$. Block $L_Q$ is an $n_Q \times n_Q$ matrix and in its turn has a particular structure as well. Since, by construction, vertices in $Q_i$ are not reachable from vertices in $Q_j$ for $j > i$, $L_Q$ is again a lower diagonal block matrix:

$$L_Q = \begin{pmatrix} B_{Q1} & & & \\ * & B_{Q2} & & \\ * & * & \ddots & \\ * & * & * & B_{Qk} \end{pmatrix}, \tag{2.8}$$

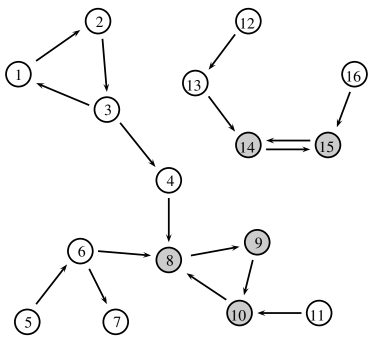
where $B_{Qi}$ is a square block of order $n_{Qi} = |Q_i|$, for $i = 1, \ldots, k$. The asterisks denote blocks that do not necessarily have to be filled with zeros.



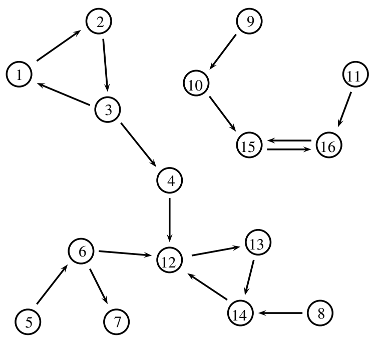
(a) Example graph $G$.



(b) Condensation of $G$. Gray vertices have no predecessors and represent root sets in $G$. *(Step 1.)*



(c) Gray vertices are in $Q$. *(Step 3.)*



(d) Vertices are renumbered. *(Step 4.)*

Figure 2.4: Illustration of graph partitioning procedure.

| $i$ | $R_i$ | $R_i^+$ | $Q_i$ | $V_i$ |
|---|---|---|---|---|
| 1 | $\{1,2,3\}$ | $\{4\}$ | $\emptyset$ | $\{1,2,3,4\}$ |
| 2 | $\{5\}$ | $\{6,7\}$ | $\emptyset$ | $\{5,6,7\}$ |
| 3 | $\{11\}$ | $\emptyset$ | $\{8,9,10\}$ | $\{8,9,10,11\}$ |
| 4 | $\{12\}$ | $\{13\}$ | $\emptyset$ | $\{12,13\}$ |
| 5 | $\{16\}$ | $\emptyset$ | $\{14,15\}$ | $\{14,15,16\}$ |

Table 2.2: Partition of the vertex set of $G$, according to the partitioning procedure.

**Example 2.11.** The Laplacian matrix of the graph $G'$, constructed in Example 2.10, is given in Figure 2.5. The block structure of (2.7) with the five blocks $L_i$ as described in Table 2.2 is clearly visible. Table 2.2 shows that sets $Q_1$, $Q_2$ and $Q_3$ are empty, so block $L_Q$ has only $B_{Q3}$ and $B_{Q5}$ on its diagonal.

Note that the subgraph of $G$ induced by the vertex set $V_i = R_i \cup R_i^+ \cup Q_i$ contains a directed spanning tree $T_i$ for all $i = 1, \ldots, k$. Moreover, the union of all these vertex sets is $V$. As mentioned earlier, the number of root sets of $G$ is a lower bound on the cardinality of a spanning forest of $G$. Since the number of trees created in this way is equal to the number of root sets of $G$, the set $\{T_1, \ldots, T_k\}$ is a smallest spanning forest of $G$. Figure 2.6 shows for graph $G$ from Example 2.10 a smallest spanning forest constructed in this way.

### 2.3.3 Properties of Laplacians of directed graphs

Properties of Laplacian matrices of undirected graphs are widely known and can for example be found in [10]. For Laplacians of directed graphs this is somewhat different. However, based on the partitioning procedure proposed in Section 2.3.2, we can derive some elementary results on directed graphs. Standard results on undirected graphs follow from them.

Lemma 2.4 stated that a Laplacian has exactly one zero eigenvalue if and only if its corresponding graph has a directed spanning tree. The following lemma generalizes this statement.

**Lemma 2.12.** *The number of zero eigenvalues of the Laplacian of a graph $G = (V, E)$ is equal to the cardinality of a smallest spanning forest of this graph.*

*Proof.* Let vertices $V$ be numbered according to the procedure described in Section 2.3.2 and consider block structure (2.7). Since $L$ is a lower triangular block matrix, the algebraic multiplicity of eigenvalue zero in $L$ is equal to the sum of the algebraic multiplicities of eigenvalue zero in matrices $L_1, \ldots, L_k$ and $L_Q$.

Matrix $L_i$ equals the Laplacian matrix of the subgraph of $G$ induced by vertex set $R_i \cup R_i^+$. This subgraph contains a directed spanning tree, so according to Lemma 2.4, $L_i$ has exactly one zero eigenvalue.

Now it remains to show that $L_Q$ has no zero eigenvalues. Consider the block structure of this matrix as given in (2.8). We will show that none of the matrices $B_{Qi}$ can have a zero eigenvalue. Note that matrix $B_{Qi}$ is closely related to the Laplacian matrix $L_{Qi}$ of the subgraph of $G$ induced by vertex set $Q_i$. Suppose vertex $v \in Q_i$, then the number of edges to this vertex from a vertex not in $Q_i$ is given by

$$\widehat{d_{in}}(v) = |\{(w, v) \in E : w \notin Q_i\}|.$$

Let $D_{Qi}$ define the $n_{Qi} \times n_{Qi}$ diagonal matrix with entries $\widehat{d_{in}}(v)$ for $v \in Q_i$. Then

$$B_{Qi} = L_{Qi} + D_{Qi}.$$

$$L = \left(\begin{array}{cccc|ccc|c|cc|c|ccccc}
1 & 0 & -1 & & & & & & & & & & & & & \\
-1 & 1 & 0 & & & & & & & & & & & & & \\
0 & -1 & 1 & & & & & & & & & & & & & \\
0 & 0 & -1 & 1 & & & & & & & & & & & & \\
\hline
 & & & & 0 & & & & & & & & & & & \\
 & & & & -1 & 1 & 0 & & & & & & & & & \\
 & & & & 0 & -1 & 1 & & & & & & & & & \\
\hline
 & & & & & & & 0 & & & & & & & & \\
\hline
 & & & & & & & & 0 & & & & & & & \\
 & & & & & & & & -1 & 1 & & & & & & \\
\hline
 & & & & & & & & & & 0 & & & & & \\
\hline
0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & -1 & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 2 & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 2 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 2 \\
\multicolumn{4}{c}{R_1 \cup R_1^+} & \multicolumn{3}{c}{R_2 \cup R_2^+} & \multicolumn{1}{c}{R_3} & \multicolumn{2}{c}{R_4 \cup R_4^+} & \multicolumn{1}{c}{R_5} & \multicolumn{5}{c}{Q}
\end{array}\right)$$

Figure 2.5: Laplacian of graph $G'$, obtained by the partitioning procedure from graph $G$ in Example 2.10.

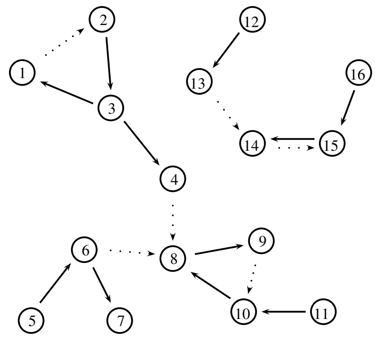



Figure 2.6: A smallest forest of $G$.

The $p$th eigenvalue of $B_{Qi}$ is equal to the sum of the $p$th eigenvalue of $L_{Qi}$ and $(D_{Qi})_{pp}$. Since the subgraph of $G$ induced by vertices $Q_i$ contains a spanning tree, $L_{Qi}$ has exactly one zero eigenvalue. Suppose that this spanning tree has vertex $r$ as its root.

Again take a look at the sufficiency part of the proof of Lemma 2.4. For the basic case (number of edges in the graph is $n-1$) it is shown that $G$ has exactly one zero eigenvalue by a convenient renumbering of the vertices in $G$. This renumbering yields that the Laplacian can be written as a lower triangular matrix with upper left entry 0 and all the other diagonal entries larger than 0. Here, number 1 was assigned to the root vertex.

Suppose that the graph induced by $Q_i$ on $G$ has $n_{Qi}-1$ edges. Then we can renumber the vertices in $Q_i$ in the same way and find that number 1 is assigned to root $r$. In graph $G$ there always exists an edge from some vertex not in $Q_i$ to $r=1$, so $(D_{Qi})_{11} > 0$ and in the basic case $B_{Qi}$ has no zero eigenvalues. We can add edges in a similar way as was done in the remainder of the proof of Lemma 2.4 to show that $B_{Qi}$ has no zero eigenvalues in general. □

**Lemma 2.13.** *Suppose $G$ is a graph on $n$ vertices and $L$ its Laplacian. Let $k$ denote the number of trees necessary to span $G$. Then the rank of $L$ is $n-k$.*

*Proof.* To show that the rank of $L$ is $n-k$, we will show that the dimension of the kernel of $L$ is $k$. The kernel of $L$ is equal to the eigenspace of $L$ corresponding to eigenvalue 0, so we will construct $k$ linear independent eigenvalues corresponding to this eigenvalue. The algebraic multiplicity of eigenvalue 0 is always larger or equal to its geometric multiplicity and we have seen in Lemma 2.12 that the algebraic multiplicity of eigenvalue 0 is equal to $k$. The existence of $k$ such eigenvectors would imply that the dimension of the kernel of $L$ is $k$.

Let vertices $V$ be numbered according to the procedure described in Section 2.3.2 and consider block structure (2.7). Since all $L_i$ are Laplacian matrices, $L_i 1_{n_i} = 0_{n_i}$. Now define the following vectors $v_i \in \mathbb{R}^n$ for $i = 1, \ldots, k$. Let

$v_i = \mathrm{col}(v_{i1}, \dots, v_{ik}, v_{iQ})$, where

$$v_{ij} = \begin{cases} 1_{n_i} & \text{if } j = i, \\ -L_Q^{-1} A_i 1_{n_i} & \text{if } j = Q, \\ 0_{n_j} & \text{else.} \end{cases}$$

Matrix $L_Q$ is invertible, because it has no zero eigenvalues as was shown in the proof of Lemma 2.12. By construction, $Lv_i = 0_n$ and $v_1, \dots, v_k$ are $k$ linear independent eigenvectors for eigenvalue 0. □

The rank of an undirected graph $G$ is $n - c$, where $c$ is the number of connected components of $G$ [10]. This can be seen as a consequence of Lemma 2.13. Each root set $R_i$ in this case consists out of one connected components of $G$ and all $R_i^+$ and $Q$ are empty sets.

### 2.3.4 Implications for consensus theory

In Section 2.1 we inferred that the multi-agent system abiding the dynamics $\dot{x}(t) = -Lx(t)$ is stable from the fact that the trajectories $x(t)$ that satisfy these dynamics are bounded (see Lemma 2.3). Another way to determine the stability of this system is by considering the eigenvalues of $-L$. We already derived that the eigenvalues of $-L$ are either zero or lie in the open left half plane (see Lemma 2.3). For the system to be stable we need moreover that the algebraic and geometric multiplicity of eigenvalue zero are the same. In Lemma 2.4 we showed that this is true in case a graph has a directed spanning tree. Combining the results from Lemma 2.12 and Lemma 2.13 we find that for any Laplacian matrix the algebraic and geometric multiplicity of eigenvalue zero are the same. This implies that system $\dot{x}(t) = -Lx(t)$ is stable.

Another interesting implication of the partitioning procedure as introduced in Section 2.3.2 is that it allows us to express the limit state of system $\dot{x}(t) = -Lx(t)$ with initial conditions $x(0) \in \mathbb{R}^n$ in terms of these initial conditions.

Since the Laplacian can be partitioned as in (2.7) the dynamics of the agents in a set $R_i \cup R_i^+$ only depend on the states of the agents in this set. If we consider a block $L_i$ in more detail, we see that it is again of the form:

$$L_i = \begin{pmatrix} L_{R_i} & \\ * & L_{R_i^+} \end{pmatrix}$$

Let $x_{R_i}$ denote the states of the agents in $R_i$. The dynamics of these states are thus given by

$$\dot{x}_{R_i}(t) = -L_{R_i} x_{R_i}(t).$$

In $G$ there exists a directed spanning tree on the vertices in $R_i$, since the subgraph of $G$ induced by $R_i$ is strongly connected. Therefore, from Theorem 2.6 it follows that consensus will be reached in this part of the graph. The consensus value is $v_i^T x_{R_i}(0)$, where $v_i \in \mathbb{R}^{|R_i|}$ is nonnegative and satisfies $1_{|R_i|}^T v_i = 1$ and $L_{R_i}^T v_i = 0_{|R_i|}$. If the subgraph of $G$ induced by $R_i$ is balanced, then by Theorem 2.7 this consensus value is the average of the initial states of agents in $R_i$.

In $R_i \cup R_i^+$ the dynamics are given by

$$\dot{x}_{R_i \cup R_i^+}(t) = -L_i x_{R_i \cup R_i^+}(t).$$

Since by construction there exists a directed spanning tree on $R_i \cup R_i^+$, consensus will be reached on this set. This implies that the agents in $R_i^+$ will converge to $v_i^T x_{R_i}(0)$ as well.

For $t \to \infty$ the state $x(t)$ will converge to an equilibrium. In this equilibrium it holds that $\dot{x}(t) = -Lx(t) = 0_n$. Using this in combination with the partitioning (2.7) we can find the values to which agents in $Q$ will converge. In equilibrium it holds that

$$A_1 \begin{pmatrix} x_{R_1} \\ x_{R_1^+} \end{pmatrix} + \ldots + A_k \begin{pmatrix} x_{R_k} \\ x_{R_k^+} \end{pmatrix} + L_Q x_Q = 0_{|Q|}.$$

Therefore the values of the agents in $Q$ are given by

$$x_Q = L_Q^{-1} \left( A_1 \begin{pmatrix} x_{R_1} \\ x_{R_1^+} \end{pmatrix} + \ldots + A_k \begin{pmatrix} x_{R_k} \\ x_{R_k^+} \end{pmatrix} \right)$$

Since in equilibrium the states of the agents in $R_i \cup R_i^+$ are given by $v_i^T x_{R_i}(0)$, we can reformulate this as

$$x_Q = L_Q^{-1} \left( A_1 1_{n_1} v_1^T x_{R_1}(0) + \ldots + A_k 1_{n_k} v_k^T x_{R_k}(0) \right).$$

Consequently, the values to which the agents will converge can be expressed in terms of the initial states of the agents in the root sets of the graph that represents their interaction structure.

# Chapter 3

# Leader-follower networks

While in the previous chapter, all agents abided the consensus algorithm, in this chapter we will distinguish between *leader* and *follower* agents, as was done in for example [11, 21]. We define the dynamics of leader and follower agents in Section 3.1 and in Section 3.2 we discuss under what conditions a multi-agent system on a leader-follower network reaches consensus. In Section 3.3 we see that the maximum modulus principle that is known in complex analysis has an interpretation for leader-follower networks. In Section 3.4 we assume that network dynamics can be influenced by external signals through the leaders and wonder under what conditions the entire multi-agent system is controllable. Finally, in Section 3.5 we study Schur complements of Laplacian matrices, that have an application in leader-follower networks as well.

## 3.1 Leader and follower dynamics

Consider a multi-agent system with $n$ agents and let directed graph $G = (V, E)$ represent their interaction topology. We use $l$ and $f$ to denote affiliations with leaders and followers, respectively. For example, a *follower graph* is the subgraph of $G$ induced by follower vertex set $V^f \subseteq V$. The cardinality of $V^f$ is $n_f$.

We assume again that all agents are evolving in $\mathbb{R}$. Now collect the states of all followers in a vector $x_f \in \mathbb{R}^{n_f}$ and those of all leaders in $x_l \in \mathbb{R}^{n_l}$. Let $x$ denote the concatenation of these vectors. Assume that the leaders are indexed last in the original graph $G$. Then the graph Laplacian can be partitioned as

$$L = \begin{pmatrix} L_f & L_{fl} \\ L_{lf} & L_l \end{pmatrix}. \tag{3.1}$$

The aim is to steer all the states of agents to a pre-defined goal value $a \in \mathbb{R}$ through changing the dynamics of the leaders. So we do not only want to reach consensus here, but also reach the specific consensus value $x = a \cdot 1_n$. There are multiple ways to define a consensus algorithm in this context, two of which we will present here. The difference between these two lies in the choice of the dynamics for the leaders. The dynamics of the followers are governed by the standard consensus algorithm, as described in Section 2.1. Taking into account the structure from (3.1) we can write the follower dynamics as

$$\dot{x}_f(t) = -L_f x_f(t) - L_{fl} x_l(t). \tag{3.2}$$

Hereafter we describe two types of leader dynamics, those of dynamic leaders and those of static leaders.

**Dynamic leaders.** The dynamics of dynamical leaders are equal to those of the followers, but include an additional attraction term [11]. For leader $i$, define $\delta_i(t) = |a - x_i(t)|$. The dynamics of leader $i$ are given by

$$\dot{x}_i(t) = \sum_{j \in N_i} (x_j(t) - x_i(t)) + F(x_i, a),$$

where $F(x_i, a)$ is the goal atraction function

$$F(x_i, a) = \begin{cases} f(\delta_i)\frac{a - x_i}{\delta_i} & \text{if } \delta_i > 0 \\ 0 & \text{if } \delta_i = 0 \end{cases} \tag{3.3}$$

with $f(\delta) \geq 0$. This function directs state $x_i$ towards $a$; if $x_i(t) > a$, then it contributes negatively to $\dot{x}_i(t)$. The magnitude of the 'steering force' is determined by $f(\delta_i)$.

We require that $f(0) = 0$ and $\lim_{\delta \to 0^+} \frac{f(\delta)}{\delta} < \infty$, so that $\lim_{x \to a} F(x, d) = F(a, a) = 0$ and $F(x, a)$ is continuous. A simple example of a goal attraction function is obtained by choosing $f(\delta) = \delta$.

**Static leaders.** The states of static leaders are assumed to already be fixed at the goal value $a$, that is, $\dot{x}_l(t) = 0$ and $x_l(0) = a \cdot 1_{n_l}$.

Even though the dynamics of dymanic leaders look more complicated than those of static leaders, in case we choose $f(\delta) = \delta$ in goal attraction function (3.3), we can formulate a leader-follower system with dynamics leaders in the static leader framework. If $f(\delta) = \delta$, the dynamics of leader $i$ are given by

$$\dot{x}_i(t) = \sum_{j \in N_i} (x_j(t) - x_i(t)) + (a - x_i(t)).$$

Equivalently, we can write that

$$\begin{aligned} \dot{x}_l(t) &= -L_{lf} x_f(t) - L_l x_l(t) + 1_{n_l} \cdot a - x_l(t) \\ &= -L_{lf} x_f(t) - (L_l + I_{n_f}) x_l(t) + 1_{n_l} \cdot a. \end{aligned} \tag{3.4}$$

Now suppose we introduce an extra agent in the system, agent $n + 1$, whose state is assumed to be fixed at value $a$: $x_{n+1}(0) = a$ and $\dot{x}_{n+1}(t) = 0$. Let the new communication topology be represented by a graph $G' = (V', E')$, where $V' = V \cup \{n+1\}$ and $E' = E \cup \{(i, n+1) : i \in V^l\}$. All leader vertices are now

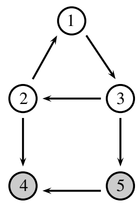



Figure 3.1: Graph representing the communiction topology of a leader-follower system. Gray vertices correspond to leaders.

connected to vertex $n+1$ and we can think of the latter as a 'supreme leader' with followers in $V$. The Laplacian corresponding to $G'$ is given by

$$L' = \left(\begin{array}{cc|c} L_f & L_{fl} & 0_{n_f} \\ L_{lf} & L_l + I_{n_l} & -1_{n_l} \\ \hline 0_{n_f}^T & 0_{n_l}^T & 0 \end{array}\right),$$

where the partitioning is as in (3.1). The dynamics of the agents in $V^l$ in the new dynamical system are still given by (3.4). The dynamics of the agents in $V^f$ have not changed either. Therefore, in this way we have formulated the leader-follower system with dynamic leaders in the static leader framework.

## 3.2 Consensus conditions

In this section we will focus on leader-follower systems with static leaders. Using the results from Chapter 2 we can easily find conditions such that the agents reach consensus.

The presence of leaders in a system does not necessarily imply that they influence the dynamics of the followers in any way. This is illustrated in the following example.

**Example 3.1.** Consider the graph in Figure 3.1. The Laplacian of this graph, partitioned as in (3.1), is given by

$$L = \left(\begin{array}{ccc|cc} 2 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ \hline 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & 2 \end{array}\right).$$

Therefore, in this case $\dot{x}_f(t) = -L_f x_f(t)$. The dynamics of the followers do not depend on the states of the leaders.

This motivates to distinguish between leaders that actually 'fulfill their role as a leader' and those that do not. We therefore introduce the following definition.

**Definition 3.2.** Agent $i$ is called *active* in $S \subseteq V$ if there exists an edge $(i, j) \in E$ such that $j \in S$. In particular, a leader is called active if he is active in $V^f$.

We will show that any leader-follower system with static leaders has an equivalent representation as a system without leaders.

**Lemma 3.3.** *Consider a leader-follower system $\Sigma_1$ on a graph $G$ with static leaders that have initial value $a \in \mathbb{R}$. Let $G'$ be the graph corresponding to Laplacian matrix*

$$L' = \begin{pmatrix} L_f & L_{fl} & 0_{n_f} \\ 0_{n_l \times n_f} & I_{n_l} & -1_{n_l} \\ 0_{n_f}^T & 0_{n_l}^T & 0 \end{pmatrix},$$

*and assume that in the system $\Sigma_2$ on $G'$ agent $n+1$ has initial value $a$. Then agents 1 through $n$ in $\Sigma_1$ will converge to the same values as agents 1 through $n$ in $\Sigma_2$.*

*Proof.* First note that the dynamics in $\Sigma_1$ are given by

$$\dot{x}(t) = -\begin{pmatrix} L_f & L_{fl} \\ 0_{n_l \times n_f} & 0_{n_l \times n_l} \end{pmatrix} x(t) = -\tilde{L}x(t), \tag{3.5}$$

where $x_l(0) = a \cdot 1_{n_l}$. Let $\tilde{G} = (V, \tilde{E})$ be the graph that corresponds to Laplacian matrix $\tilde{L}$. This graph is equal to $G$ with all edges that entered a leader vertex removed. Graph $G' = (V', E')$, that corresponds to Laplacian matrix $L'$, is the graph with vertex set and edge set defined as follows:

$$\begin{aligned} V' &= V \cup \{n+1\}, \\ E' &= \tilde{E} \cup \{(i, n+1) : i \in V^l\}. \end{aligned}$$

System $\Sigma_2$ is defined on this graph. In this system it was supposed that $x_{n+1}(0) = a$.

Set $R_1 = \{n+1\}$ is a root set of $G'$ and by construction $V^l \in R_1^+$, so by the results from Section 2.3.4 we have that $x_l \to a \cdot 1_{n_l}$ for $t \to \infty$. Therefore, the limit state of the agents would have been the same, if we would have assumed that $x_l(0) = a \cdot 1_{n_l}$. In that case the dynamics of the agents in $V^l$ would be $\dot{x}_l(t) = 0$, so if we assume that $x_l(0) = a \cdot 1_{n_l}$, the dynamics of the first $n$ agents in $\Sigma_2$ are given by (3.5). Therefore, agents 1 through $n$ in $\Sigma_1$ will converge to the same values as agents 1 through $n$ in $\Sigma_2$. □

From the combination of this lemma and the fact that on a regular graph consensus is reach if and only if this graph has a spanning tree (see Theorem 2.6), we can derive the following conditions for a leader-follower system to reach consensus.

**Lemma 3.4.** *A leader-follower system with static leaders on graph $G$ reaches consensus if and only if $G'$ has a directed spanning tree.*

If such a spanning tree exists, its root is of course vertex $n+1$. The following lemma provides with a consensus condition that does not require us to consider graph $G'$.

**Lemma 3.5.** *A leader-follower system with static leaders on graph $G$ reaches consensus if and only if in each root set of $G$ at least one leader is active.*

*Proof.* [$\Leftarrow$] If in each root set of $G$ at least one leader is active, then in $G'$ every vertex is reachable from vertex $n+1$, so $G'$ has a directed spanning tree and by Lemma 3.4 the system reaches consensus.

[$\Rightarrow$] Suppose there exists a root set of $G$ where no leader is active. By construction, this set will also be a root set of $G'$. Since $\{n+1\}$ is a root set of $G'$ as well and one root sets can (by definition) not be reached from another root set, $G'$ can not be spanned by one tree. □

In case graph $G$ is undirected, Lemma 3.5 implies that consensus is reached if and only if graph $G'$ is connected. Lemma 3.4 implies that consensus is reached if and only if there exists at least one leader in each connected component of $G$.

## 3.3 Maximum modulus principle on graphs

In this section we consider leader-follower systems with static agents, where the leaders do not necesserily have to have the same initial state $a$. We call the vertices corresponding to leader agents *boundary vertices* and those corresponding to followers *internal vertices* [23]. The introduction of this terminology allows us to talk about the boundary of a graph and state a maximum modulus principle on graphs that is comparable to the maximum modulus principle known in complex analysis.

The graph Laplacian can be considered as an analog of the continuous Laplace operator $\Delta$. For a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ this operator is defined as

$$\Delta f(x) = \left( \frac{\partial^2}{\partial x_1^2} + \ldots + \frac{\partial^2}{\partial x_n^2} \right) f(x),$$

where $x = \text{col}(x_1, \ldots, x_n)$. A function $f$ that satisfies $\Delta f(x) = 0$ is called a *harmonic function*. Let $\Omega \in \mathbb{R}^n$ be a bounded and connected open set. The maximum modulus principle states that for a function $f$ continuous on the closure $\overline{\Omega}$ of $\Omega$ and harmonic on $\Omega$, the maximum modulus is attained on the boundary $\partial\Omega$, that is,

$$\sup_{x \in \overline{\Omega}} |f(x)| = \sup_{x \in \partial\Omega} |f(x)|.$$

This implies that $f$ can attain no local maximum or minimum in $\Omega$.

Let us now formulate a similar set-up on a directed graph $G = (V, E)$. We have already defined internal and boundary vertices. Assume that $G$ consists of at least one boundary vertex. Now call a function $h : V \to \mathbb{R}$, where $h(i) = x_i$, *harmonic* if

$$L_f x_f + L_{fl} x_l = 0_{n_f}. \tag{3.6}$$

A harmonic function is called constant if $x_1 = x_2 = \ldots = x_n$. A similar definition of harmonic functions for the case of undirected graphs is given in [4]. The maximum modulus principle for undirected graphs is formulated in [4] as well. In the following we will state and prove a similar statement for directed graphs. The undirected graph case then follows as a consequence.

**Theorem 3.6.** *The maximum and minimum of a non-constant harmonic function $h$ on a directed graph $G$ are attained on a boundary vertex, if in $G$ each internal vertex is reachable from a boundary vertex.*

*Proof.* Function $h$ is harmonic, so for some $x_1, \ldots, x_n$ equation (3.6) holds. This implies that for all vertices $i \in N^f$:

$$\sum_{k \in N_i} (x_i - x_k) = 0.$$

This can be rewritten as

$$x_i = \sum_{k \in N_i} \frac{x_k}{|N_i|} = \frac{1}{|N_i|} \sum_{k \in N_i^f} x_k + \frac{1}{|N_i|} \sum_{k \in N_i^l} x_k,$$

where $N_i^l$ and $N_i^f$ are the sets of leader and follower neighbours of $i$.

We first show that the maximum is attained on a boundary vertex, that is, $x_i \leq \max\{x_j : j \in N^l\}$ for $i \in N^f$. Suppose there exists a vertex $i \in N^f$ such that $x_i > x_j$ for all $k \in N^l$ and choose this vertex such that $x_i \geq x_j$ for $j \in N^f$. We will show by induction that such a vertex can not exist at any finite distance from a boundary vertex. Vertex $i$ can not be at distance 1 from a boundary vertex, because then

$$x_i = \frac{1}{|N_i|} \sum_{k \in N_i^f} x_k + \frac{1}{|N_i|} \sum_{k \in N_i^l} x_k < \frac{|N_i^f| + |N_i^l|}{|N_i|} x_i = x_i.$$

Assume that vertex $i$ can not exist at some distance $D > 1$ from a boundary vertex. We will show that it can not exist at distance $D + 1$ either. In that case $|N_i^l| = 0$ and

$$x_i = \sum_{k \in N_i^f} \frac{x_k}{|N_i|}.$$

This yields that $x_j = x_i$ for all vertices $j \in N_i$. One of these vertices has distance $D$ to a boundary vertex, but by assumption a vertex with this state value could not exist at distance $D$ from a boundary vertex. Consequently, vertex $i$ could not have had distance $D + 1$ from a boundary vertex. By induction we now have shown that $x_i \leq \max\{x_j : j \in N^l\}$ for $i \in N^f$. In a similar way it can be derived that $x_i \geq \min\{x_j : j \in N^l\}$ for all $i \in N^f$. □

Note that the condition in Theorem 3.6 is equivalent to demanding that in each root set of $G$ one leader is active or that the graph $G'$, as defined in Section 3.2, has a directed spanning tree. Another way to formulate Theorem 3.6 is that under either of these conditions in an equilibrium state of a leader-follower system, for all followers $i$ the following will hold:

$$\min\{x_j(0) : j \in V^l\} \leq x_i \leq \max\{x_j(0) : j \in V^l\}.$$

The maximum modulus principle on undirected graphs follows directly from Theorem 3.6.

**Corollary 3.7.** *The maximum and minimum of a non-constant harmonic function $h$ on an undirected graph $G$ are attained on boundary vertices of $G$, if there exists at least one leader in each connected component of $G$.*

## 3.4 Controllability

In Section 3.1 we defined leader-follower systems in which the leaders abided fixed dynamics. In this section we generalize this and assume that the states of the leaders can be chosen arbitrarily: $x_l(t) = u(t)$, where $u(t)$ is an exogenous control signal. In this section the controllability of system (3.2) will be investigated.

In [21] the controllabity of single-leader networks was investigated using the Popov-Belevitch-Hautus (PBH) test (see Section B.2.2). In system (3.2) only a part of the Laplacian matrix is included. The following lemma allows us to study the controllability of system (3.2) by the consideration of an extended system, that does include the complete Laplacian. Consider the following general system:

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{3.7}$$

with $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ and $A$ and $B$ of compatible dimensions.

**Lemma 3.8.** *System* (3.7) *is controllable if and only if system*

$$\begin{pmatrix} \dot{x}(t) \\ \dot{u}(t) \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} + \begin{pmatrix} 0_{n\times m} \\ I_m \end{pmatrix} v(t) \tag{3.8}$$

*is controllable, where $v \in \mathbb{R}^m$ and $C$ and $D$ of compatible dimensions.*

*Proof.* System (3.8) is not controllable if and only if there exists a vector $q = \text{col}(q_x, q_u)$ with $q_x \in \mathbb{R}^n$ and $q_u \in \mathbb{R}^m$ satisfying

$$q^T \begin{pmatrix} A & B \\ C & D \end{pmatrix} = q^T\lambda \quad \text{such that} \quad q^T \begin{pmatrix} 0_{n\times m} \\ I_m \end{pmatrix} = 0^T_{n+m}.$$

The latter requirement is equivalent to demanding $q_u = 0_m$. This implies that system (3.8) is not controllable if and only if there exists a vector $q_x \in \mathbb{R}^n$ such that

$$\begin{pmatrix} q_x^T A & q_x^T B \end{pmatrix} = \begin{pmatrix} q_x^T & 0_m^T \end{pmatrix} \lambda.$$

This is true if and only if there exist a $q_x \in \mathbb{R}^n$ such that $q_x^T A = \lambda q_x^T$ for some $\lambda$ and $q_x^T B = 0_m^T$, that is, system (3.7) is not controllable. □

Now again consider a leader-follower system with a communication topology represented by a graph $G = (V, E)$ with Laplacian $L$. Let the dynamics of the system be as in (3.2) and assume $x_l(t) = u(t)$, where $u(t)$ is an exogenous control signal. The following statement about the controllability of the followers is a direct consequence of Lemma 3.8.

**Corollary 3.9.** *System* (3.2) *is controllable if and only if $L$ has no left eigenvalue with zeros on all indices that correspond to a leader.*

This statement is a generalization of Proposition 5.4 from [21]. It generalizes this proposition in three ways. Firstly, leader-follower systems with multiple leaders are considere here, instead of single leader systems. Secondly, Corrolary 3.9 provides with necessary and sufficient conditions for system (3.2) to be controllable, while in [21] only a neccessary condition was given. Finally, Corrolary 3.9 is formulated for directed graphs instead of for undirected graphs.

So far we have looked at controllability, but we can also study observability in this context. To this end we define the following output function:

$$y(t) = -L_{lf}x_f(t) - L_l x_l(t). \tag{3.9}$$

The observability conditions for the system given by equations (3.2) and (3.9) with input $x_l(t)$ follow from the PBH test for observability (see Section B.2.2) and strongly resemble the conditions given in Corollary 3.9, since controllability and observability are dual properties.

**Lemma 3.10.** *The system given by equations* (3.2) *and* (3.9) *is observable if and only if $L$ has no right eigenvalue with zeros on all indices that correspond to a leader.*

The Laplacian matrix of an undirected graph is symmetric. Therefore, a system given by equations (3.2) and (3.9) defined on an undirected graph is controllable if and only if it is observable.

## 3.5 Schur complements of Laplacians

Let $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{n\times l}$, $C \in \mathbb{R}^{k\times m}$ and $D \in \mathbb{R}^{k\times l}$ and define matrix $M$ as

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

Then the *Schur complement* of $A$ with respect to $M$ is defined by $D - CA^{+}B$, where $A^{+}$ denotes the generalized inverse[1] of $A$. Schur complements have a wide range of applications, for example in numerical analysis and multivariate statistics [29] and the analysis of electrical networks [14, 25].

Also in the study of leader-follower networks Schur complements arise. Consider a leader-follower system with dynamics

$$\begin{pmatrix} \dot{x}_l(t) \\ \dot{x}_f(t) \end{pmatrix} = -\begin{pmatrix} L_l & L_{lf} \\ L_{fl} & L_f \end{pmatrix} \begin{pmatrix} x_l(t) \\ x_f(t) \end{pmatrix}.$$

[1] For the definition and properties of the generalized inverse we refer to [2].

We assume that the leaders are static, that is, $\dot{x}_l(t) = 0_{n_l}$. Equivalently we could require that

$$-L_l x_l(t) - L_{lf} x_f(t) = 0_{n_l}.$$

This yields the following follower dynamics:

$$x_f(t) = -(L_f - L_{fl} L_l^+ L_{lf}) x_f(t).$$

Matrix $L_f - L_{fl} L_l^+ L_{lf}$ is the Schur complement of $L_l$ with respect to $L$. This idea was already used in [25]. Here it is stated that any symmetric positive semi-definite matrix $L$ with non-negative diagonal elements, non-positive off-diagonal elements and with zero row and column sums, can be considered as a weighted Laplacian matrix of a certain undirected graph, and conversely. It is shown that if a graph $G$ is connected, then all Schur complements of the corresponding Laplacians are well defined and are again Laplacian matrices of a connected graph. In the remainder of this section we will derive similar results for undirected graphs.

First of all, any square matrix $L$ with non-negative diagonal elements, non-positive off-diagonal elements and with zero row sums, can be considered as a weighted Laplacian matrix of a certain directed graph, and conversely.

**Theorem 3.11.** *All Schur complements of a Laplacian of a directed graph $G = (V, E)$ with $|V| > 2$ are well defined if and only if $G$ is strongly connected. In particular, all Schur complements are again Laplacian matrices of strongly connected directed graphs.*

*Proof.* [$\Leftarrow$] Since all vertices in $G$ have positive indegree, the diagonal elements of $L$ are strictly positive. We will first show, adopting the proof line from [25, 28], that the Schur complement of upper left entry of $L$ is a Laplacian matrix of a directed graph.

Let L([1,1]) denote the matrix obtained from $L$ by deleting the first row and column and let $l_r$ and $l_c$ be the first row and column of $L$, respectively, without their first elements. Then the Schur complement of element $l_{11}$ is given by

$$\hat{L} = L([1,1]) - \frac{1}{l_{11}} l_c l_r. \tag{3.10}$$

Since all elements of $l_c$ and $l_r$ are non-positive, the off-diagonal elements of $\hat{L}$ are also non-positive. Let the entries of $\hat{L}$ be denoted by

$$\hat{L} = \begin{pmatrix} \hat{l}_{22} & \dots & \hat{l}_{2n} \\ \vdots & \ddots & \vdots \\ \hat{l}_{n2} & \dots & \hat{l}_{nn} \end{pmatrix}.$$

This matrix again has zero row sums, since for all $i = 2, \dots, n$ we have

$$\sum_{j=2}^{n} \hat{l}_{ij} = \sum_{j=2}^{n} \left( l_{ij} - \frac{1}{l_{11}} l_{i1} l_{1j} \right) = \sum_{j=2}^{n} l_{ij} - \frac{l_{i1}}{l_{11}} \sum_{j=2}^{n} l_{1j}.$$

Matrix $L$ has zero row sums, so $l_{11} = -\sum_{j=2}^{n} l_{1j}$ and

$$\sum_{j=2}^{n} \hat{l}_{ij} = \sum_{j=2}^{n} l_{ij} + l_{i1} = 0.$$

This also implies that the diagonal elements of $\hat{L}$ are non-negative, so $\hat{L}$ is again a Laplacian matrix.

We will now prove that the the diagonal elements of $\hat{L}$ are strictly positive. Suppose they are not, then since we already showed that they are non-negative, for all $i = 2, \ldots, n$,

$$\hat{l}_{ii} = l_{ii} - \frac{1}{l_{11}} l_{1i} l_{i1} = 0. \tag{3.11}$$

By definition of the graph Laplacian we have that $l_{11} > -l_{1i}$ and $l_{ii} > -l_{i1}$ (the sum of all incoming weights is larger than only one of them) and by assumption $l_{ii} > 0$. Define $a = -l_{1i}/l_{11}$, then $0 \leq a \leq 1$. Equation (3.11) implies that $-a l_{i1} = l_{ii}$ and therefore $a = 1$ and $l_{ii} = -l_{i1}$. From $a = 1$ it follows that $l_{11} = -l_{1i}$. Consequently, only edge $(1, i)$ is entering vertex $i$ and only edge $(i, 1)$ is entering vertex 1. Thus, since $|V| > 2$, graph $G$ can not be strongly connected. This leads to a contradiction, so the diagonal elements of $\hat{L}$ are strictly positive.

Graph $G$ is strongly connected, so for all $i, j \in V$ there exists a path from $i$ to $j$. Let $\hat{G} = (\hat{V}, \hat{E})$ be the graph on vertices $\hat{V} = \{2, \ldots, n\}$ corresponding to Laplacian matrix $\hat{L}$. Since $\hat{l}_{ij} = l_{ij} - \frac{1}{l_{11}} l_{i1} l_{1j} \leq l_{ij}$, the edge set $\hat{E}$ contains at least those edges for which $(i, j) \in E$. Therefore, for all vertices $i, j \in V \setminus \{1\}$ for which there exists a path in $G$ from $i$ to $j$ not passing vertex 1, there exists such a path in $\hat{G}$ as well.

We will show that in case the path from $i$ to $j$ in $G$ does cross vertex 1, there still exists a path from $i$ to $j$ in $\hat{G}$. Suppose that in this path vertex 1 is preceded by vertex $v$ and succeeded by vertex $w$, so $l_{1v} < 0$ and $l_{w1} < 0$. By the preceding paragraph we know that there exists a path from $i$ to $v$ in $\hat{G}$ and also one from $w$ to $j$. Moreover,

$$\hat{l}_{wv} = l_{wv} - \frac{1}{l_{11}} l_{1v} l_{w1} < 0,$$

so vertex $(v, w) \in \hat{E}$. Therefore, there exists a path from $i$ to $j$ in $\hat{G}$ as well. This proves that $\hat{G}$ is strongly connected.

In order to prove the claim for an arbitrary Schur complement, we note that any Schur complement can be obtained by the successive application of taking Schur complements with respect to diagonal elements. □

In Theorem 3.11 we assume that $|V| > 2$ for graph $G = (V, E)$ to let equation (3.11) lead to a contradiction. In case $|V| = 2$, matrix $L$ looks like

$$L = \begin{pmatrix} a & -a \\ -b & b \end{pmatrix},$$

where $a, b \geq 0$. To ensure that arbitrary Schur complements can be taken we need that $a, b > 0$, that is, $G$ has to be strongly connected also in this case. The Schur complement of either diagonal element of $L$ is 0.

We now resume the proof of Theorem 3.11.

*Proof (continued).* [$\Rightarrow$] Suppose graph $G$ is not strongly connected. Then $G$ consists out of more than one strongly connected component. Let $S_1$ denote such a strongly connected components that has no incoming edges (represented by a vertex without predecessors in the condensation of $G$) and let $\{1, 2, \ldots, k\}$ be the vertices from $S_1$. Then the Laplacian of $G$ is given by

$$L = \begin{pmatrix} l_{11} & \ldots & l_{1k} & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ l_{k1} & \ldots & l_{kk} & 0 & \ldots & 0 \\ l_{k+1,1} & \ldots & l_{k+1,k} & l_{k+1,k+1} & \ldots & l_{k+1,n} \\ \vdots & & \vdots & & \ddots & \vdots \\ l_{n1} & \ldots & l_{nk} & \ldots & \ldots & l_{nn} \end{pmatrix}.$$

We will show that the Schur complement of the leading diagonal $k \times k$ block $L_{S_1}$ is not well defined. If we take the Schur complement of $l_{11}$ as in (3.10), matrix $\hat{L}$ is structured similarly to $L$ and the upper left $(k-1) \times (k-1)$ block represents a strongly connected component. After repeating this process $k-1$ times, the upper left element will be zero, so the Schur complement of $L_{S_1}$ is not well defined. □

In Theorem 3.11 we have seen that the Schur complement of the Laplacian $L$ of a strongly connected graph $G$ is again the Laplacian of a strongly connected graph $\hat{G}$. It can be shown in a similar way that if we additionaly assume that $G$ is balanced, that is, $L^T 1_n = 0_n$, then $\hat{G}$ is balanced as well. To this end, we only need to repeat the argument provided after equation (3.10) for zero column sums.

In [25] it was already proven that if an undirected graph $G$ is connected, then all Schur complements of the Laplacian of $G$ are well defined and, in particular, they are again Laplacian matrices of a connected undirected graphs. Note that this statement is a special case of Theorem 3.11 and only necessary and no sufficient conditions are provided.

# Chapter 4

# Output synchronization

Suppose we have a network of $n$ agents, whose communication topology is represented by graph $G = (V, E)$. The dynamics of agent $i$ are given by

$$\Sigma_i : \begin{cases} \dot{x}_i = f(x_i, u_i), \\ y_i = h(x_i, u_i) \end{cases}$$

Here $x_i \in \mathbb{R}^d$ represents the state, $u_i \in \mathbb{R}^m$ the input and $y_i \in \mathbb{R}^m$ the output of agent $i$ and functions $f$ and $h$ are sufficiently smooth. The agents are said to *output synchronize* if for all $i, j \in V$,

$$|y_i(t) - y_j(t)| \to 0, \qquad \text{as } t \to \infty.$$

Note that this definition strongly resembles the definition of state consensus. The main difference is, however, that here the agents can synchronize to time-varying steady-state values and in the consensus problems we encountered so far the asymptotic states of the agents are constant. Moreover, so far the dynamics imposed on the agents were rather simple, linear systems where $f(x_i, u_i) = u_i$ with $u_i = \sum_{j \in N_i}(y_j - y_i)$, and $y_i = x_i$. Synchronization models are usually nonlinear.

In this chapter we assume that each agent's dynamics is represented by a system that is affine in its input $u$ and does not have a feedthrough term, conform [9]. These systems are of the following form

$$\Sigma_a : \begin{cases} \dot{x} = f(x) + g(x)u, \\ y = h(x), \end{cases}$$

with $g(x)$ an $n \times m$ matrix [24]. Such a system is passive if there exists a non-negative real-valued storage function $S(x)$ such that for all input functions $u$,

$$\frac{\partial S(x)}{\partial x}^T [f(x) + g(x)u] \leq u^T h(x). \tag{4.1}$$

For a discussion on passive systems, we refer to Appendix B. Inequality (4.1) is the differential dissipation inequality, applied to system $\Sigma_a$. Equivalently, system $\Sigma_a$ is passive if there exists a storage function $S(x)$ that satisfies the *Hill-Moylan conditions*:

$$\frac{\partial S(x)}{\partial x}^T f(x) \leq 0 \quad \text{and} \quad \frac{\partial S(x)}{\partial x}^T g(x) = h^T(x).$$

Suppose that all $n$ agents have a system $\Sigma_i$, that is affine in its input $u$ and without feedthrough term and suppose these sytems are passive. In [9] it is assumed that the agents are coupled together using

$$u_i = \sum_{j \in N_i} (y_j - y_i). \tag{4.2}$$

Note that there is in fact no difference between this input and the one considered in the previous chapters, as was made clear in the introduction of this chapter. Let $x = \text{col}(x_1, x_2, \ldots, x_n) \in \mathbb{R}^{nd}$, $u = \text{col}(u_1, \ldots, u_n) \in \mathbb{R}^{nm}$ and $y = \text{col}(y_1, \ldots, y_n) \in \mathbb{R}^{nm}$. In this way function (4.2) can for all agents can be written simultaneously as

$$u = -(L \otimes I_m)y, \tag{4.3}$$

where $L$ is the Laplacian of graph $G$.

**Theorem 4.1.** *If $G$ is strongly connected and balanced, then the system is globally stable and the agents output synchronize.*

*Proof.* Let $S(x) = S_1(x_1) + \ldots + S_n(x_n)$ be a Lyapunov function candidate, where $S_i(x_i)$ denotes the storage function for agent $i$. Using the Hill-Moylan conditions and the control function (4.3), we get

$$\begin{aligned} \dot{S} = \frac{\partial S(x)}{\partial x}^T \dot{x} &= \sum_{i=1}^{n} \frac{\partial S_i(x_i)}{\partial x_i}^T \dot{x}_i \\ &= \sum_{i=1}^{n} \frac{\partial S_i(x_i)}{\partial x_i}^T (f(x_i) + g(x_i)u_i) \\ &\leq \sum_{i=1}^{n} y_i^T u_i \\ &= -y^T (L \otimes I_m) y \end{aligned}$$

Matrix $L \otimes I_m$ can be decomposed into a symmetric part $L_S$ and a skew-symmetric part $L_{SS}$. Since $y^T L_{SS} y = 0$, we only have to conside the symmetric part, which is $\frac{1}{2}(L \otimes I_m + (L \otimes I_m)^T)$. The communication graph is balanced, so according to Lemma A.3 we have that $L + L^T = DD^T$, where $D$ is the incidence matrix of graph $G$. Consequently, $L \otimes I_m + (L \otimes I_m)^T = (D \otimes I_m)(D \otimes I_m)^T$ and

$$\dot{S} \leq -\frac{1}{2} |(D^T \otimes I_m)y|^2 \leq 0,$$

so the system is globally stable. Consider the set $E = \{x \in \mathbb{R}^{nd} : \dot{S}(x) = 0\}$. This set is characterized by all trajectories such that $|(D^T \otimes I_m)y| = 0$. The last property implies that $y_i = y_j$ for all $j \in N_i$. It follows from Lasalle's invariance principle that, if $G$ is strongly connected, the agents can output synchronize. □

## 4.1 Example: driven pendula

As an example of a physical synchronization system we consider a set of driven pendula (see also [9]). We assume here that all pendula have unit mass. Let $\varphi_i$ be the angular displacement of pendulum $i$ and $\dot{\varphi}_i$ its angular velocity. The motion of such a pendulum is governed by the following non-linear dynamics:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \varphi_i(t) \\ \dot{\varphi}_i(t) \end{pmatrix} = \begin{pmatrix} \dot{\varphi}_i(t) \\ -\frac{g}{l_i}\sin(\varphi_i(t)) + \frac{u_i}{l_i} \end{pmatrix}, \tag{4.4}$$

where $g$ is the gravitational constant, $l_i$ is the length of pendulum $i$ and $u_i$ the driving force. The energy $V_i$ of a pendulum is given by

$$V_i(\varphi_i, \dot{\varphi}_i) = \tfrac{1}{2} l_i^2 \dot{\varphi}_i^2 + g l_i (1 - \cos(\varphi_i)).$$

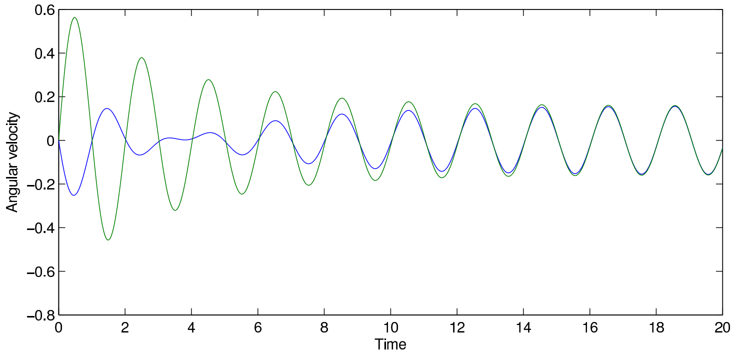
Taking the time-derivate yields that

$$\dot{V}_i = \frac{\partial V}{\partial \varphi_i}\dot{\varphi}_i + \frac{\partial V}{\partial \dot{\varphi}_i}\ddot{\varphi}_i = g l_i \sin(\varphi_i)\dot{\varphi}_i - l_i^2 \dot{\varphi}_i (\tfrac{g}{l_i}\sin(\varphi_i) - \tfrac{u_i}{l_i}) = u_i l_i \dot{\varphi}.$$

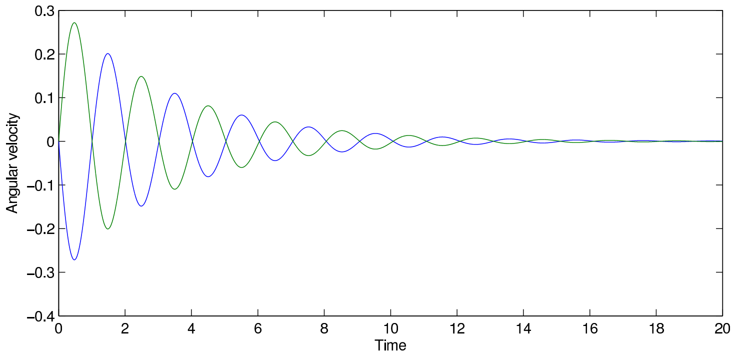
So we conclude that with output $y_i = l_i \dot{\varphi}_i$ the system passive. We will now consider a set of these pendula such that pendulum $i$ is coupled to the pendula in $N_i$ using input (4.2). We assume that all pendula have unit length $l_i$.

As a first experiment we take two pendula, one with initial angle of 0.1, the other with initial angle -0.2. The initial velocity of both pendula is 0. The resulting dynamics are shown in Figure 4.1a. We see that indeed the pendula synchronize on their angular velocities, as expected from Theorem 4.1. Taking opposite but equal angles results in the system damping out, as can be seen Figure 4.1b. Also here synchronization takes place.

In Figure 4.2 we have a system of four pendula for various topologies and fixed initial states. The initial velocities of all pendula is 0. In Figure 4.2a we see that when all pendula influence each other the pendula synchronize quickly. On the other hand, if we have a circular topology, where pendulum 1 influences 2, which in its turn influences 3 etc., then convergence takes longer to achieve. Finally, we see in Figure 4.2c what happens if the topology has two components that are not connected. Within the components synchronization takes place, but not between the components.

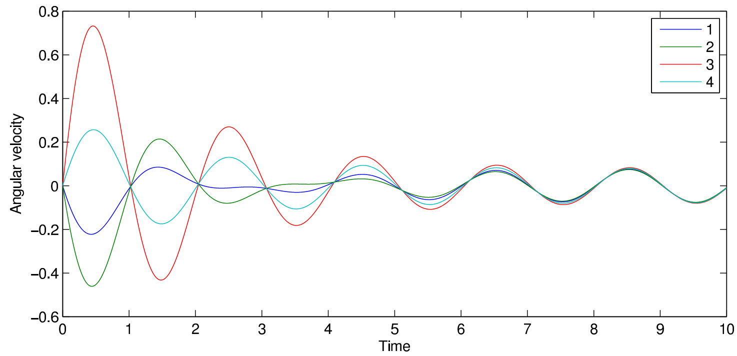



(a) Initial angles 0.1 and -0.2.

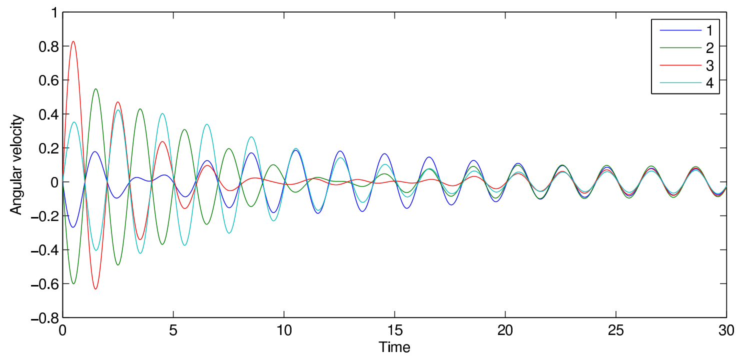



(b) Initial angles 0.1 and -0.1.

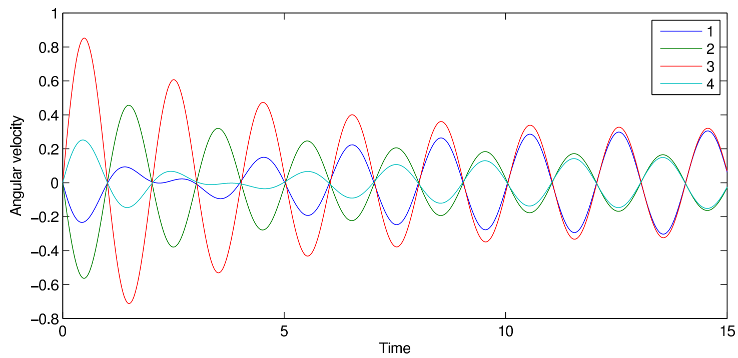
Figure 4.1: Two pendula with various initial angles. Both pendula influence each other and after a while their angular velocities synchronize.




(a) Every pendulum influences every other pendulum.




(b) Pendula with circular interaction topology (1 to 2, 2 to 3, etc.).




(c) Pendula 1 and 3, and 2 and 4 influence each other.

Figure 4.2: Four pendula with stationary initial angles $\varphi_1 = 0.1$, $\varphi_2 = 0.2$, $\varphi_3 = -0.3$ and $\varphi_4 = -0.1$ respectively and varying interaction topologies.

# Appendix A

# Graph theory

In this appendix we give a brief review of some basic concepts in (algebraic) graph theory. The appendix provides the general background which we draw from in the remainder of this thesis and is included mainly to render the work self-contained. Moreover, it aims to clarify some graph theoretical ambiguities that exist in literature on consensus algorithms. For a thorough overview of the field of (algebegraic) graph theory we refer to [4, 10].

## A.1 Definitions

An *undirected graph* $G$ is an ordered pair of sets $(V, E)$, where $V$ is the set of vertices and $E$ the set of edges, given by a subset of the unordered pairs $\{i, j\}$ of distinct vertices of $V$. In a *directed graph* the edge set $E$ consists of ordered pairs $(i, j)$ of distinct vertices of $V$. Vertex $j$ is called the *head* and vertex $i$ the *tail* of directed edge $(i, j)$. We only consider graphs with finite $V$ and $E$, that have no loops or parallel edges.

Any undirected graph $G$ can be regarded as a directed one, by replacing all its edges with two oppositely oriented directed edges with the same ends. This is the *associated directed graph* of $G$ [5]. Another way to obtain a directed graph from an undirected graph is by replacing each of its edges by just one of the two possible directed edges with the same ends. Such a directed graph is called an *orientation* of $G$. With any directed graph $G$ we can associate an undirected graph by introducing an undirected edge between two vertices if they are joined by at least one directed edge in $G$. This is called the *underlying undirected graph* of $G$.

A (directed) *path* of length $r$ from $i_0$ to $i_r$ is a graph with vertex set $\{i_0, i_1, \ldots, i_r\}$, such that the edge set consists of the (directed) edges from $i_n$ to $i_{n+1}$ for $n = 0, \ldots, r-1$. Here vertex $i_r$ is said to be *reachable* from vertex $i_0$. The *distance* between two vertices is the length of the shortest path connecting them. An undirected graph is *connected* if any two vertices can be joined by a path. Similarly, a directed graph is *strongly connected* if any two vertices can be joined by a directed path. A directed graph is called *weakly connected* if its underlying undirected graph is connected.

Graph $G' = (V', E')$ is a *subgraph* of a graph $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. If $G'$ contains exactly those edges of $G$ that join two vertices in $V'$, then $G'$ is said to be the subgraph *induced by* $V'$. An induced subgraph of $G$ that is maximal, subject to being (strongly) connected, is called a *(strongly) connected component* of $G$.

A *spanning tree* of an undirected graph $G$ is a subgraph of $G$, that contains all of its vertices and in which any two vertices are connected by exactly one path. A *directed spanning tree* is a subgraph of a directed graph $G$, in which there exists a vertex $r$ such that there is a directed path from $r$ (the *root*) to all other vertices in $G$.

In an undirected graph the *degree* $d(i)$ of a vertex $i$ is equal to the number of vertices to which it is adjacent. In a directed graph the *in-degree* $d_{in}(i)$ of a vertex $i$ is the number of edges ending in $i$ and its *out-degree* $d_{out}(i)$ is the number of edges starting from $i$. If for all vertices $i \in V$ the in-degree equals the out-degree, the graph is called *balanced*.

## A.2 Connectedness

In this section the connection between strong and weak connectedness and the existence of a directed spanning tree in a directed graph will be explored. In the literature on consensus algorithms these concepts are used interchangeably. We will end the section with a result that provides some clarification regarding this issue.

Any strongly connected graph is also weakly connected. The underlying undirected graph of a directed spanning tree is connected, so a graph that contains a directed spanning tree is weakly connected. In a strongly connected graph any vertex can serve as a root a directed spanning tree, so such a graph contains a directed spanning tree. Note that neither having a spanning tree nor weakly connectedness implies strongly connectedness. To this end an additional requirement is needed.

**Lemma A.1.** *In case a directed graph $G$ is balanced, $G$ is weakly connected if and only if it is strongly connected [10].*

*Proof.* Strongly connectedness trivially implies weakly connectedness. Now let $G$ be a weakly connected and balanced graph. We can partition its vertex set into strongly connected components $S_1, S_2, \ldots, S_m$. Since $G$ is balanced, we have for $k = 1, \ldots, m$ that

$$\sum_{i \in S_k} d_{in}(i) = \sum_{i \in S_k} d_{out}(i). \tag{A.1}$$

Edges within $S_k$ contribute to both the in-degree and the out-degree sums. Substracting the total number of these edges from both sides in equation (A.1) yields that the number of edges leaving $S_k$ is equal to the number of edges entering $S_k$.

Consider a graph $H$ with vertices $s_1, s_2, \ldots, s_m$ and an edge from $s_i$ to $s_j$ if there exists an edge from a vertex in $S_i$ to one in $S_j$. Suppose that $m > 1$. If for some vertex $s_k$ it holds that $d_{in}(s_k) = 0$, then the previous argument implies $d_{out}(s_k) = 0$. This contradicts the weakly connectedness of $G$, so $d_{in}(s_k) > 0$ and $d_{out}(s_k) > 0$ for all $k = 1, \ldots, m$. Since $m$ is finite, the graph $H$ contains a cycle. This contradicts the maximality of the strongly connected components $S_1, S_2, \ldots, S_m$. Therefore, $m = 1$ and $G$ is a strongly connected graph. □

In the literature on consensus algorithms the concepts directed spanning tree, weakly connected and strongly connected are used, but often diffferent concepts are used to express the same result. The following corollary may in that case give some clarification.

**Corollary A.2.** *In case a directed graph $G$ is balanced, the following three statements are equivalent:*

*(i) $G$ has a directed spanning tree,*

*(ii) $G$ is weakly connected,*

*(iii) $G$ is strongly connected.*

## A.3 Laplacian matrix

The *incidence matrix* $D$ of a directed graph $G = (V, E)$ is a matrix with rows and columns indexed by the vertices and edges of $G$, respectively, such that

$$d_{if} = \begin{cases} -1 & \text{if the vertex } i \text{ is the tail of edge } f, \\ 1 & \text{if the vertex } i \text{ is the head of edge } f, \\ 0 & \text{otherwise.} \end{cases} \tag{A.2}$$

The *Laplacian* $L$ of $G$ is defined as

$$l_{ij} = \begin{cases} -1 & \text{if } (j, i) \in E, \\ d_{in}(i) & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{A.3}$$

**Lemma A.3.** *Let $G = (V, E)$ be a directed balanced graph with Laplacian $L$ and incidence matrix $D$. Then $L + L^T = DD^T$ [13].*

*Proof.* From the definition of the Laplacian it follows directly that

$$(L + L^T)_{ij} = \begin{cases} -1 & \text{if either } (i, j) \in E \text{ or } (j, i) \in E, \\ -2 & \text{if both } (i, j) \in E \text{ and } (j, i) \in E, \\ 2 \cdot d_{in}(i) & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{A.4}$$

In a balanced graph the in-degree equals the out-degree for all vertices, so in case $i = j$ we have

$$(DD^T)_{ii} = d_{in}(i) + d_{out}(i) = 2 \cdot d_{in}(i) = (L + L^T)_{ii}.$$

In the other cases we consider the sum $(DD^T)_{ij} = \sum_{f \in E} d_{if} d_{jf}$. The only non-zero entries in this sum are the ones where edge $f$ involves both vertices $i$ and $j$. Because $d_{i(i,j)} = -1$ and $d_{j(i,j)} = 1$, the sum will be equal to minus the number of edges that involve both $i$ and $j$. This proves $(L + L^T)_{ij} = (DD^T)_{ij}$ in the cases that $i \neq j$. □

The Laplacian of an undirected graph $G$ is defined as the Laplacian (2.2) of the associated directed graph of $G$. Such a Laplacian is a symmetric matrix. Combined with the previous lemma this yields the following result for undirected graphs.

**Lemma A.4.** *Let $G$ be an undirected graph and let graph $H$ be an arbitrary orientation of $G$. Then the Laplacian $L$ of $G$ can be decomposed as $L = D_H D_H^T$, where $D_H$ is the incidence matrix of $H$.*

*Proof.* Let $D_G$ denote the incidence matrix of the associated directed graph of $G$. All edges in this graph come in pairs: $(i, j)$, $(j, i)$. Graph $H$ is a subgraph of $G$, obtained by removing one edge from each of these pairs. Therefore, all non-zero off-diagonal entries in $D_G D_G^T$ will be $-2$ and in $D_H D_H^T$ these same entries will be $-1$, conform expression (A.4). Also the sum $d_{in}(i) + d_{out}(i)$ will be twice as large for vertices $i$ in the associated directed graph of $G$ as in $H$. Consequently, $D_G D_G^T = 2D_H D_H^T$. Since $L$ is symmetric,

$$2L = L + L^T = D_G D_G^T = 2D_H D_H^T$$

and therefore $L = D_H D_H^T$. □

# Appendix B

# Systems theory

This appendix summarizes a few definitions and results from systems theory. Only what is relevant for this thesis is discussed. The results that are explicitly stated as lemma or theorem are referred to from somewhere else in the thesis.

## B.1 Spectral properties of matrices

The *algebraic multiplicity* of an eigenvalue $\lambda$ of $A$ is the multiplicity of $\lambda$ as a root of the characteristic polynomial $\det(\lambda I_n - A)$. The *geometric multiplicity* of $\lambda$ is the dimension of the kernel of $\lambda I_n - A$. For each eigenvalue the geometric multiplicity is smaller or equal to the algebraic multiplicity.

**Theorem B.1** (Gersgorin disc theorem)**.** *Every eigenvalue of a matrix $A \in \mathbb{C}^{n\times n}$ lies inside one of the Gersgorin discs $D(a_{ii}, R_i)$, which is a closed disc centered at $a_{ii}$ and with radius $R_i = \sum_{j\neq i} |a_{ij}|$ [2].*

*Proof.* Let $\lambda$ be an eigenvalue of $A$ and $x$ a corresponding eigenvector. Choose $i$ such that $|x_i| \geq |x_j|$ for all $j = 1, \ldots, n$. Since $\lambda$ is an eigenvalue of A,

$$\sum_{j=1}^{n} a_{ij} x_j = \lambda x_i.$$

Substracting $a_{ii}x_i$ from both sides and dividing by $x_i$ gives that

$$|\lambda - a_{ii}| = \left| \frac{\sum_{j\neq i} a_{ij} x_j}{x_i} \right| \leq \sum_{j\neq i} |a_{ij}| = R_i,$$

so eigenvalue $\lambda$ lies in the closed disc $D(a_{ii}, R_i)$. □

A vector $x \in \mathbb{R}^n$ is called *positive* (or *nonnegative*) if all of its entries are positive (nonnegative). We denote this by $x > 0_n$ ($x \geq 0_n$). Positive or nonnegative matrices are defined similarly.

Let $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of a matrix $A \in \mathbb{R}^{n \times n}$. The *spectral radius* of $A$ is defined as

$$\rho(A) = \max_{i=1,\ldots,n} |\lambda_i|.$$

**Lemma B.2.** *If $A \in \mathbb{R}^{n \times n}$ is nonnegative, then $\rho(A)$ is an eigenvalue of $A$ and there exists a nonnegative vector $x \neq 0_n$, such that $Ax = \rho(A)x$ [12].*

## B.2 Linear differential systems

A *linear time-invariant differential system* is a system that can be written as

$$\begin{aligned} x(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{B.1}$$

where $x(t) \in \mathbb{R}^n$ denotes the *state* (vector) of the system, $u(t) \in \mathbb{R}^m$ the *input* (vector) and $y(t) \in \mathbb{R}^p$ the *output* (vector). The matrices $A, B, C, D$ have sizes $n \times n$, $n \times m$, $p \times n$ and $p \times m$, respectively. Since these matrices are constant in time, the system is called *time-invariant*. We will assume that given initial condition $x(0) = x_0$ and an input $u(t)$, $t \geq 0$ are *admissible*, that is, they are such that the solution $x(t)$ and output $y(t)$ of this system are well defined.

Now consider system

$$\dot{x}(t) = Ax(t) \tag{B.2}$$

with initial condition $x(0) = x_0$, where $t \geq 0$. The solution of this system is given by $x(t) = e^{At}x_0$, where the *matrix exponential* $e^A \in \mathbb{R}^{n \times n}$ is defined as

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}.$$

The matrix exponential is easily determined in case $A$ is diagonizable, that is, there exists an invertible matrix $P$ such that $P^{-1}AP = D$, where $D = \text{diag}(\lambda_1, \ldots, \lambda_n)$. In that case $e^A = Pe^D P^{-1}$, where $e^D = \text{diag}(e_1^\lambda, \ldots, e_n^\lambda)$. If $A$ is not diagonizable, $e^A$ can be determined using the *Jordan normal form* $J$ of $A$ (see for example [20]), which is close to a diagonal form. The following lemma shows the relation between the eigenvalues and eigenvectors of $A$ and $e^A$.

**Lemma B.3.** *If $\lambda$ is an eigenvalue of $A$ and $v$ a corresponding eigenvector, then $e^{At}v = e^{\lambda t}v$.*

### B.2.1 Stability

Consider the first order differential equation $\dot{x} = f(x)$, with $x \in \mathbb{R}^n$ and let $x(t, x_0)$ denote its solution at time $t$, given initial condition $x(0) = x_0$. A vector $\overline{x}$ that satisfies $f(\overline{x}) = 0$ is called an *equilibrium point*. An equilibrium point is called *stable* if for every $\epsilon > 0$ there exists a $\delta > 0$ such that $|x_0 - \overline{x}| < \delta$ implies $|x(t, x_0) - \overline{x}| < \epsilon$ for all $t > 0$. This means that the solution remains in a neighbourhood of the equilibrium point, provided that $x_0$ lies sufficiently close to it. An equilibrium point is called *asymptotically stable* if it is stable and in addition the solution converges to the equilibrium point, provided that $x_0$ lies sufficiently close to it. Equilibrium point $\hat{x}$ is called *unstable* if it is not stable.

Again consider system (B.2). This system is called (asymptotically) stable if the equilibrium point $\overline{x} = 0_n$ is (asymptotically) stable. Let $\lambda_1, \ldots, \lambda_k$ be the distinct eigenvalues of $A \in \mathbb{R}^{n \times n}$. The origin $\overline{x} = 0_n$ is asymptotically stable if and only if $\text{Re}(\lambda_i) < 0$ for all $i = 1, \ldots, k$. It is stable if and only if $\text{Re}(\lambda_i) \leq 0$ for all $i = 1, \ldots, k$ and for each eigenvalue $\lambda_i$ with $\text{Re}(\lambda_i) = 0$, the algebraic and geometric multiplicity are the same.

With the *Routh-Hurwitz stability criterion* we can determine whether a system is assymptotically stable without the explicit computation of the eigenvalues of $A$. The criterion indicates the number of roots of the characteristic equation $\det(A - \lambda I) = 0$ with non-negative real part based on the coefficients $a_0, \ldots, a_n$ in

$$\det(A - \lambda I) = a_n \lambda^n + a_{n-1} \lambda^{n-1} + \ldots + a_1 \lambda + a_0.$$

Let $a_n$ be positive. If it is negative, then multiply all coefficients by $-1$. Routh stability criterion consists of several steps, but here we only present the first one.

**Step 1.** If any of the coefficients $a_0, a_1, \ldots, a_{n-1}$ is negative or zero, there is at least one root of the characteristic equation with positive real part.

For a clear description and illustrative examples of the complete Routh criterion we refer to [3].

### B.2.2 Controllability and observability

System (B.1) is called *controllable* if any state $x_1 \in \mathbb{R}^n$, can be reached starting from an arbitrary state $x_0 \in \mathbb{R}^n$, in finite time $t_1 > 0$, by application of a suitable admissible input $u$. The system is called *observable* if the initial state $x_0$ can be constructed from knowing $u$ and $y$ on the interval $[0, t_1]$ for some finite $t_1$. By means of the Popov-Belevitch-Hautus (PBH) tests we can test whether a system is controllable or observable.

**Theorem B.4** (PBH test for controllability)**.** *System* (B.1) *is not controllable if and only if there exist a left eigenvector of $A$, i.e., $q^T A = \lambda q^T$ for some $\lambda$, such that $q^T B = 0_m^T$.*

**Theorem B.5** (PBH test for observability)**.** *System* (B.1) *is not observable if and only if there exist a right eigenvector of $A$, i.e., $Aq = \lambda q$ for some $\lambda$, such that $Cq = 0_p$.*

Note that controllability is completely determined by matrices $A$ and $B$ and observability by matrices $A$ and $C$.

## B.3 Dissipative non-linear systems

Consider state space system

$$\Sigma : \begin{cases} \dot{x} = f(x, u), \\ y = h(x, u) \end{cases}$$

where $x \in \mathbb{R}^n$, $u, y \in \mathbb{R}^m$ and functions $f$ and $h$ are sufficiently smooth. Assume that moreover a real-valued function $s$ defined on $\mathbb{R}^{2m}$ is given. This function will be called the *supply rate*. We assume that $s$ is locally integrable [27].

Let $x(t_1)$ denote the state of $\Sigma$ at time $t_1$ resulting from initial condition $x(t_0) = x_0$ and input $u$ [24]. A system $\Sigma$ said to be *dissipative* with respect to supply rate $s$ if there exists a function $S : \mathbb{R}^n \to \mathbb{R}^+$, called the *storage function*, such that for all $x_0$, $t_1 > t_0$ and input functions $u$,

$$S(x(t_1)) \leq S(x(t_0)) + \int_{t_0}^{t_1} s(u(t), y(t))\mathrm{d}t, \tag{B.3}$$

The above inequality is called the *dissipation inequality*. It expresses that at time $t_1$ the 'stored energy' $S(x(t_1))$ in the system can not be larger than the sum of the energy present at a previous time $t_0$ and the energy that was supplied in the meanwhile, $\int_{t_0}^{t_1} s(u(t), y(t))\mathrm{d}t$. Inequality (B.3) therefore represents the fact that no energy can be 'created' in the system, only dissipated. If it holds with equality for all $x_0$, $t_1 > t_0$ and input functions $u$, then $\Sigma$ is *lossless* with respect to $s$.

A system $\Sigma$ is said to be *passive* if it is dissipative with respect to supply rate $s(u, y) = u^T y$. If it is lossless with respect to this supply rate, $\Sigma$ is called *conservative*.

If storage function $S$ is continuous differentiable, we can rewrite the dissipation inequality by dividing by $t_1 - t_0$ and lettting $t_1 \to t_0$, thus obtaining

$$\frac{\partial S(x)}{\partial x}^T f(x, u) \leq s(u, h(x, u)). \tag{B.4}$$

This inequality is called the *differential dissipation inequality*.

# Bibliography


[1] R.P. Agaev and P.Yu. Chebotarev. The matrix of maximum out forests of a digraph and its applications. *Automation and Remote Control*, 61(9):1424–1450, 2000.

[2] Dennis S. Bernstein. *Matrix mathematics*. Princeton University Press, 2009.

[3] S.K. Bhattacharya. *Linear control systems*. Dorling Kindersley (India) Pvt. Ltd., 2011.

[4] Béla Bollobás. *Modern graph theory*. Springer-Verlag New York, Inc., 1998.

[5] J.A. Bondy and U.S.R. Murty. *Graph theory*. Springer, 2008.

[6] Vivek Borkar and Pravin P. Varaiya. Asymptotic agreement in distributed estimation. *IEEE Transactions on Automatic Control*, 27(3):650–655, 1982.

[7] Pavel Chebotarev. Comments on "Consensus and cooperation in networked multi-agent systems". *Proceedings of the IEEE*, 98(7):1353–1354, 2010.

[8] Pavel Chebotarev and Rafig Agaev. Matrices of forests and the analysis of digraphs. Retrieved August 18, 2011 from `http://arxiv.org/PS_cache/math/pdf/0508/0508171v2.pdf`, 2006.

[9] Nikhil Chopra and Mark W. Spong. Passivity-based control of multi-agent systems. In Sadao Kawamura and Mikhail Svinin, editors, *Advances in Robot Control*, pages 107–134. Springer Berlin Heidelberg, 2006.

[10] Chris Godsil and Gordon Royle. *Algebraic graph theory*. Springer-Verlag New York, Inc., 2001.

[11] Tove Gustavi, Dimos V. Dimarogonas, Magnus Egerstedt, and Xiaoming Hu. On the number of leaders needed to ensure network connectivity in arbitrary dimensions. In *Mediterranean Conference on Control and Automation*, 2009.

[12] Richard. A Horn and Charles R. Johnson. *Matrix analysis.* Cambridge University Press, 1985.

[13] Changsheng Li and Yuzhen Wang. Protocol design for output consensus of port-controlled hamiltonian multi-agent systems. 2010. Still in review: will be published in Automatica.

[14] Sujit Kumar Mitra, P. Bhimasankaram, and Saroj B. Malik. *Matrix partial orders, shorted operators and applications.* World Scientific Publishing Co. Pte. Ltd., 2010.

[15] M.E.J. Newman. *Networks, an introduction.* Oxford University Press Inc., New York, 2010.

[16] Reza Olfati-Saber. Flocking for multi-agent systems: algorithms and theory. *IEEE transactions on automatic control*, 51(3):401–420, 2006.

[17] Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. Consensus and cooperation in networked multi-agent systems. In *Proceedings of the IEEE*, 2007.

[18] Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. Reply to "Comments on "Consensus and cooperation in networked multi-agent systems"". *Proceedings of the IEEE*, 98(7):1354–1355, 2010.

[19] Reza Olfati-Saber and Richard M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE transactions on automatic control*, 49(9):1520–1533, 2004.

[20] G.J. Olsder and J.W. van der Woude. *Mathematical systems theory.* VSSD, 2005.

[21] Amirreza Rahmani, Meng Ji, Mehran Mesbahi, and Magnus Egerstedt. Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM Journal on Control and Optimization*, 48(1):162–186, 2009.

[22] Wei Ren and Randal W. Beard. *Distributed consensus in multi-vehicle cooperative control: theory and applications.* Springer-Verlag London Limited, 2008.

[23] A.J. van der Schaft and B.M. Maschke. Port-Hamiltonian dynamics on graphs: Consensus and coordination control algorithms. In *2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, Annecy, France, 2010.

[24] Arjan van der Schaft. *$L_2$-gain and passivity techniques in nonlinear control.* Springer-Verlag London Limited, 2000.

[25] Arjan van der Schaft. Characterization and partial synthesis of the behavior of resistive circuits at their terminal. *Systems & Control Letters*, 59:423–428, 2010.

[26] John Nikolaos Tsitsiklis. *Problems in decentralized decision making and computation.* PhD thesis, Massachusetts institute of technology, 1984.

[27] Jan C. Willems. Dissipative dynamical systems - part i: general theory. *Archive for Rational Mechanics and Analysis*, 45(5):321–351, 2010.

[28] J.C. Willems and E.I. Verriest. The behavior of resistive circuits. In *Proceedings of the Joint 48th IEEE Conference on Decision and Control and the 28th Chinese Control Conference*, pages 8124–8129, Shanghai, 2009.

[29] Fuzhen Zhang. *The Schur complement and its applications.* Springer Science+Business Media, Inc., 2005.