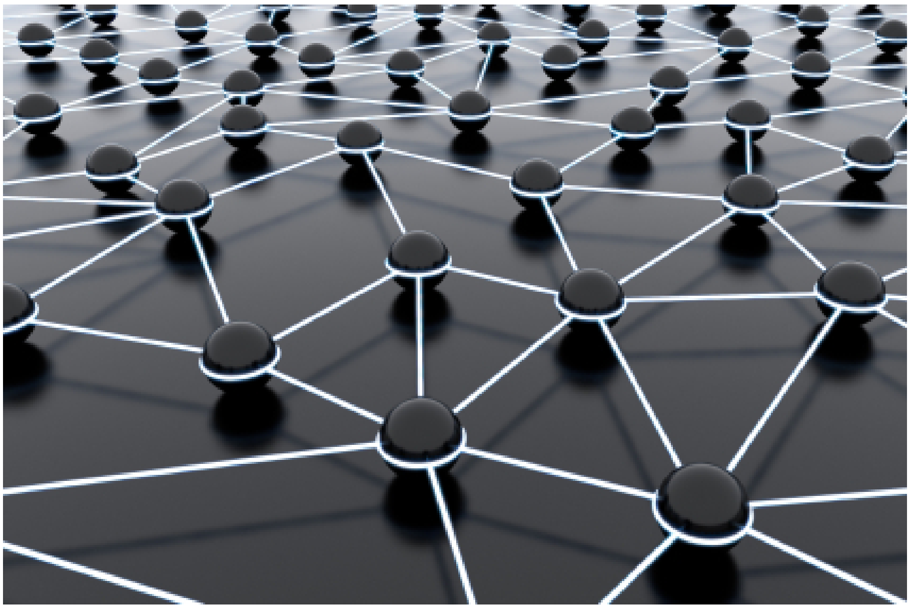
rijksuniversiteit
groningen

Master thesis applied mathematics

Model reduction of multi-agent systems with consensus dynamics

Author:
J.H.C. Klaassens

Supervisor:
prof. dr. Arjan van der Schaft



June 29, 2012

Contents

1 Introduction

This thesis is about systems which are connected to each other, more specifically multi-agent systems, where every agent has its own dynamics. In this thesis I am first looking at consensus algorithms for these multi-agent systems. It is common that a multi-agent system can become very large; so it is convenient if we will find a way of reducing a multi-agent system in such a way that the reduced system has the structure of a multi-agent system again. The structure of the connections between all the agents is given by a graph G. Furthermore, we will see that we can describe the multi-agent system subject to the consensus algorithm with the Laplacian matrix L corresponding to the graph. With this Laplacian and by giving the agents the roll of leader or follower we can write the system in the standard state space representation and so we can use the theory which is well-known in system theory. Based on this we start the search for a good reduction method. This search turns out to be not that trivial and easy as it appears.

After trying some standard reduction techniques which did not result in a reduced system corresponding to a graph, I looked for structure preserving transformations but also these failed to end up in a graph Laplacian. It was time for another approach: if we wish to end up with a new reduced Laplacian what kind of transformation do we have to use to get this reduced Laplacian from the full order Laplacian? This idea has led to the 'merge vertices' technique. Another method that was yet available in the literature is called Kron-reduction [5]. This Kron-reduction uses the Schur-complement to reduce the number of agents in the system. It turns out that also this reduction technique is a candidate for a 'good' reduction. Now that we have found two appropriate reduction techniques we have to analyze if the reduced systems are a good approximation of the full order system. Also questions arise which vertices can be discarded and which vertices must remain in order to get a good approximation.

The structure of the thesis is as follows. We start with some basic notation, followed by a brief introduction to graph theory. After that the consensus algorithms are introduced. Next, six reduction techniques are presented. Four of them are treated in the chapter Reduction and balancing methods. The other two are treated respectively in the chapters Merge vertices and Kron-reduction. In these two chapters there is also attention for the question: how to get a 'good' reduced system? We end by concluding which method is the best.

2 Notation

In this section all basic notation is introduced which will be used throughout this thesis. All notation that is more 'advanced' will be introduced in the sections were it is needed.

$A \in \mathbb{R}^{n\times m}$ denotes a real matrix with n rows and m columns. The transpose of a matrix A is denoted by A^T. Whenever a matrix A is nonsingular the inverse of the matrix is denoted by A^{-1}. We use real vectors $x \in \mathbb{R}^n$, which are $n \times 1$ matrices.
$\mathbf{0}_n = (0, \dots, 0)^T$ and $\mathbf{1}_n = (1, \dots, 1)^T$ vectors with $n \times 1$ zeros and ones respectively.

The Kronecker product is an operation on two matrices of arbitrary size resulting in another matrix [31]. If A is a $m \times n$ and B is a $p \times q$ matrix, then the Kronecker product $A \otimes B$ is the $mp \times nq$ block matrix

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

As a small example consider

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \otimes \begin{pmatrix} 1 & 3 \\ 6 & 2 \end{pmatrix} = \begin{pmatrix} 1 \cdot \begin{pmatrix} 1 & 3 \\ 6 & 2 \end{pmatrix} & 2 \cdot \begin{pmatrix} 1 & 3 \\ 6 & 2 \end{pmatrix} & 3 \cdot \begin{pmatrix} 1 & 3 \\ 6 & 2 \end{pmatrix} \\ 4 \cdot \begin{pmatrix} 1 & 3 \\ 6 & 2 \end{pmatrix} & 5 \cdot \begin{pmatrix} 1 & 3 \\ 6 & 2 \end{pmatrix} & 6 \cdot \begin{pmatrix} 1 & 3 \\ 6 & 2 \end{pmatrix} \end{pmatrix}. \tag{2}$$

A graph is denoted by G and a corresponding reduced graph by G_{red}. In the same way L is the Laplacian matrix of graph G and L_{red} the reduced Laplacian corresponding to the reduced graph G_{red}. Similar A is the graphs adjacency matrix and A_{red} is the adjacency matrix of the reduced graph. In the next chapter I will give a more elaborated overview of the needed graph theory.

3 Graph Theory

In this section fundamental graph theory is presented. We need this graph theory in order to formulate our multi-agent system. Some properties are needed to see what is happening if we apply reduction techniques and to confirm if the new system is still corresponding to a graph. The following main graph theoretical topics are treated:

- Graph $G(V, E)$ defined by its set of vertices V and set of edges E.
- The Laplacian, L, of a graph.
- The adjacency matrix A.
- Eigenvalues of L and graph connectivity.
- Several other important graph theoretical properties.

For more information about (pure) graph theory I refer to the books [9], [3] and lecture notes [12] and the article [7]. Also the articles about consensus of multi-agent systems give a brief introduction to graph theory, see for example the introduction of [18].

We use throughout this thesis an undirected, weighted graph $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$ the vertices and $E \subseteq V \times V$ the edges.
When the edges do not have a direction we speak of *undirected (symmetric)* graphs. An edge is linking two vertices, and thus can be represented as the unordered pair of the vertices linked by this edge [29]. The *order* of a graph is $|V|$, the number of vertices, and the *size* of a graph is $|E|$, the number of edges. The degree of a vertex is the number of edges connected to that vertex.
The connections in a weighted graph can be expressed in an *adjacency matrix* $A = [a_{ij}] \in \mathbb{R}^{n \times n}$ in the following way: $a_{ij} > 0$ if $(j, i) \in E$; otherwise when $(j, i) \notin E$ then $a_{ij} = 0$. Here $(j, i) \in E$ means that there is an edge between i and j. The adjacency matrix of an undirected graph has the property that $a_{ij} = a_{ji}, \forall i \neq j$, because in an undirected graph whenever $(j, i) \in E$ then also $(i, j) \in E$. Moreover because we consider graphs without self-loops we have $A_{ii} = 0, \forall i \in V$.
When the weights are not relevant we set $a_{ij} = 1$ when $(j, i) \in E$.

The *incidence matrix* $I(G)$ of a graph G is the $0, 1$ matrix with rows and columns indexed by the vertices and edges of G, respectively, such that an (i, e) entry of $I(G)$ is 1 whenever vertex i is connected to edge e and otherwise 0. $I(G)$ has dimension $n \times e$ where n is the number of vertices of the graph G and e is the number of edges.

A *path* is a sequence of edges in a graph of the form $(i_1, i_2), (i_2, i_3), \ldots$ where $i_j \in V$. A *spanning tree* is a subgraph which forms a tree and connects *all* vertices. A *tree* is a connected graph without cycles i.e. between every two vertices there is exactly one (simple) path. A *simple path* is a path with no repeated vertices.

Spectral graph theory is the study of the relationship between the eigenvalues and eigenvectors of the adjacency an the Laplacian matrix and the graph itself.

The neighbors of a vertex i are denoted by $N_i = \{j \in V : (i, j) \in E\}$, so all vertices j in the set of vertices V, which have an edge connecting j to i. Using this neighbor set we can easily define $L = [l_{ij}] \in \mathbb{R}^{n \times n}$ the *graph Laplacian* of the network, its elements are defined as follows:

$$l_{ij} = \begin{cases} -a_{ij}, & j \text{ in } N_i \\ \sum_{j=1}^{n} a_{ij}, & j = i \end{cases} \tag{3}$$

Equivalently if we define the degree matrix $D = \text{diag}\{d_1, \dots, d_n\} \in \mathbb{R}^{n \times n}$, with $d_i = \sum_{j=1}^{n} a_{ij}$ for $i = 1, \dots, n$, we can define the Laplacian of the weighted graph as

$$L = D - A,$$

which is symmetric [13]. In the case wherein the weights are not important the non-diagonal entries of the Laplacian are -1 when j is a neighbor of i and has value $|N_i|$ at the diagonal, where $|N_i|$ denotes the number of neighbors of vertex i. At the diagonal entry of the Laplacian stands the sum of the weights of all edges connected to this vertex. When a vertex j is a neighbor of vertex i we get a value $-a_{ij}$ in the Laplacian matrix at entry (i, j). By definition of this graph Laplacian the sum of a row, $\sum_{j=1}^{n} l_{ij} = 0$, for $i \in \{1, \dots, n\}$, is always zero. Also the graph Laplacian satisfies $l_{ij} \leq 0, i \neq j$. Because the graph Laplacian is important in this research a section is dedicated to the Laplacian matrix.

3.1 Laplacian matrix

In this section I present in a structured way all the properties of a symmetric graph Laplacian matrix [32], [13] and [9]. For a graph G and its Laplacian matrix L with eigenvalues $\lambda_1 \leq \lambda_1 \ldots \leq \lambda_n$:

- L is always positive-semidefinite, i.e. $\forall i, \lambda_i \geq 0$.
- The off-diagonal elements are all non-positive, i.e. $l_{ij} \leq 0, i \neq j$.
- The diagonal elements are nonnegative, i.e. $l_{ii} \geq 0, i = 1, \dots, n$.
- The row sum of every row is equal zero, i.e. $\sum_{j=1}^{n} l_{ij} = 0$, for $i = 1, \dots, n$, or written as $L\mathbf{1}_n = 0$.
- The column sum of every column is equal zero, i.e. $\sum_{i=1}^{n} l_{ij} = 0$, for $j = 1, \dots, n$, or written as $\mathbf{1}_n^T L = 0$.
- L has at least one eigenvalue at 0, $\lambda_1(L) = 0$ with right eigenvector $\mathbf{1}_n$ and left eigenvector $\mathbf{1}_n^T$.
- The number of times 0 appears as an eigenvalue of the Laplacian equals the number of connected components in the graph. λ_2 is called the algebraic connectivity. This eigenvalue is greater than 0 if and only if G is a connected graph. If G is connected then the eigenvalues $\lambda_i(L)$ with $i = 2, \dots, n$ are all positive.

In this thesis for simplicity I assume the graphs to be connected. An important remark is conversely that every symmetric positive semi-definite matrix L with diagonal elements ≥ 0 and off diagonal elements ≤ 0, with zero row and column sums is a Laplacian matrix [4].

4 Consensus

In the previous chapter we have introduced graph theory in order to be able to describe a system consisting of vertices which are called agents. All these agents have there own dynamics and they are connected to other agents and together they form a multi-agent system. These connections, with the possibility of weight along the edges, between the agents are described by a graph G and its corresponding Laplacian L. We show how we can write a multi-agent system in state space representation with the Laplacian matrix. If we also introduce a leader-follower network we will see that we can split the Laplacian in order to get the A, B, C, D matrices of the state space representation. Now because we have a multi-agent system, we want the agents to 'behave' in the same way. We will start with first order consensus.

Consider a network of n decision-making agents with the first order dynamics

$$\dot{x}_i(t) = u_i(t) \quad i = 1, \dots, n,$$

note that the state of agent i at time t, $x_i(t)$ can be a vector in $\mathbb{R}^d$, $u_i(t)$ is the control input at vertex i. The communication between such agents is described by a graph $G(V, E)$.
The first question is: what is consensus and how a system can reach it? A consensus algorithm (or protocol) is an interaction rule that specifies the information exchange between an agent and all of its

neighbors on the network [18]. In a network of agents consensus means to reach an agreement regarding a certain quantity of interest that depends on the state of all agents [18]. There are a couple of types of consensus, one of the most common is the average consensus. In this type of consensus it holds that when the consensus is reached: $x_1 = x_2 = \ldots = x_n = \frac{\sum_{i=1}^{n} x_i(0)}{n}$, so every state has the same value and is equal to the average value of the initial states.

In the multi-agent system agents are intended to reach a consensus about their states. To make a mathematical definition of consensus we consider a system of n agents and let $x_i \in \mathbb{R}^d$ denote the state of agent i. So each of the n agents has a state of dimension d.
Agents reach a state consensus if their states all converge to the same value, that is $\forall x_i(0) \in \mathbb{R}^d$ and $\forall i, j \in V$

$$|x_i(t) - x_j(t)| \to 0 \quad t \to \infty. \tag{4}$$

By reaching consensus we mean that the values of the agents asymptotically converge to an *agreement space* $x_1 = x_2 = \ldots = x_n$. This agreement space can also be expressed as $x = \alpha \mathbf{1}_n$ for an $\alpha \in \mathbb{R}$. This α is called the collective decision of the group of agents. A simple consensus algorithm is the following [18]:

$$\dot{x}_i(t) = \sum_{j \in N_i} (x_j(t) - x_i(t)), \quad i = 1, \ldots, n. \tag{5}$$

This algorithm changes the state of agent i into the direction of the values of the states of the neighbor agents. For example, if the neighbors of i have on average a higher value, then the value of x_i increases. For a graph with weights we can make the following change to the algorithm [18]:

$$\dot{x}_i(t) = \sum_{j \in N_i} a_{ij}(x_j(t) - x_i(t)), \quad i = 1, \ldots, n. \tag{6}$$

This is a distributed consensus algorithm, i.e., it guarantees convergence to a collective decision via local inter agent interactions.

This generalization of the consensus algorithm, if we first assume for simplicity of notation that the state x_i of every agent is in $\mathbb{R}$ (i.e. $d = 1$), can be rewritten in the following compact form:

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{pmatrix} = -Lx \tag{7}$$

where L is the graph Laplacian. This short notation can be verified by noticing that $L = D - A$ and simply working out the product $-(D - A)x$ to obtain (6). Now because our Laplacian L is symmetric we will reach average consensus in our consensus algorithm (6). From this short notation we can easily derive that we will reach average consensus, meaning that we have an agreement space with $\alpha = \frac{1}{n} \sum_i x_i(0)$. Because we have $\mathbf{1}_n^T \dot{x} = -\mathbf{1}_n^T Lx = 0x = 0 \Rightarrow \frac{d}{dt}(\mathbf{1}^T x)(t) = 0$, i.e. the sum, $\sum_i x_i$, is an invariant quantity and so we reach average consensus. In other words the eigenvector of the eigenvalue 0 corresponds to the consensus. In consensus $x_i \to c_i$ for $i = 1, \ldots, n$, we end up with $nc = \mathbf{1}_n^T x(0)$, hence the consensus value only depends on the initial value of the state, $c = \frac{\mathbf{1}_n^T x(0)}{n}$. This results in the following lemma [18].

Lemma 1. *Let G be a connected undirected graph. Then, the algorithm* (6) *asymptotically solves an average-consensus problem for all initial states.*

The graph Laplacian satisfies the following sum-of-squares property:

$$x^T Lx = \frac{1}{2} \sum_{(i,j) \in E} a_{ij}(x_j - x_i)^2 \tag{8}$$

where $\varphi(x) = \frac{1}{2} x^T Lx$ is called the quadratic disagreement function. Now we can rewrite the consensus algorithm in terms of the disagreement function:

$$\dot{x} = -\nabla \varphi(x). \tag{9}$$

This is called the gradient-descent algorithm. This algorithm globally asymptotically converges to the agreement space since the following two conditions hold:

1. L is a positive semidefinite matrix.
2. All equilibria of (6) are given as $\alpha\mathbf{1}_n$ for some $\alpha \in \mathbb{R}$.

If the agents have a state vector $x_i \in \mathbb{R}^d$ then we can rewrite the consensus algorithm (6) for the whole multi-agent system with the help of the Kronnecker product and the Laplacian (as in (7)). First write $x(t) = \mathrm{col}(x_1, \dots, x_n) \in \mathbb{R}^{nd}$ then the consensus algorithm is given by:

$$\dot{x}(t) = -(L \otimes I_d)x(t). \tag{10}$$

From this point we will focus on multi-agent dynamics with individual agent states in $\mathbb{R}$, i.e. $x_i \in \mathbb{R}$. However our results can directly be extended to the case of (10).

4.1 Leader-follower networks and state space representation

We know that we can write the consensus algorithm as in (7). Now instead of giving every agent the same role we can choose follower and leader agents [17], [10], [16], [15], [23]. We let some vertices be leader agents, the complement of the set of leaders in the network will be the set of follower agents. We consider single-leader systems and the more difficult multiple leader systems. With the help of the partition in followers and leaders we can look at some properties of the system like controllability and observability. In this thesis we choose the leader agents in such a way that the whole system becomes controllable, for more information see [23]. Important however is the rearrangement of the Laplacian matrix into four parts. A part only related to the followers, subscript f, a part for the leaders, subscript l and two parts for the connection between the leaders an followers. In this partition the corresponding state of the leaders become x_l and of the followers x_f. Now we reorganize the agents. The first n_f agents are the followers and the next n_l are the leaders (and of course: $n_f + n_l = n$ (the total number of vertices)). Now we can easily divide the Laplacian in parts in the following way:

$$L = \begin{pmatrix} L_f & l_{fl} \\ l_{fl}^T & L_l \end{pmatrix} \tag{11}$$

where $L_f \in \mathbb{R}^{n_f \times n_f}$, $L_l \in \mathbb{R}^{n_l \times n_l}$ and $l_{fl} \in \mathbb{R}^{n_f \times n_l}$. The graph of all the follower vertices $V^f \subseteq V$ is a subgraph of G. Also the graph of all the leader vertices $G_l(V^l, E^l)$ is a subgraph of G [23].

With (11) we can make a state space representation

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{12}$$
$$y(t) = Cx(t) + Du(t). \tag{13}$$

For matrix A we take the part of the Laplacian corresponding to the followers, $-L_f$, because of the symmetry of L we have also symmetry in the A matrix [23], so $A = A^T$. We define $B = -l_{fl}$ as the connection between the states of the leaders and followers. Again by symmetry $C = -l_{fl}^T = B^T$ and $D = -L_l$. Now we have identified our state space model where the state $x(t) = x_f(t)$ corresponds to the state of the followers and the output $y(t) = \dot{x}_l(t)$ corresponds to the first order derivative of the states of the leader vertices, and the input $u(t) = x_l(t)$ corresponds to the state of the leaders. Note that by taking the state of the leader agents as input, the followers do not have any influence anymore on the leaders i.e. the equation $\dot{x}_l(t) = -l_{fl}^T x_f(t) - L_l x_l(t)$ does not hold anymore. We give an other interpretation to this equation, namely that of the output of the system. We obtain:

$$\dot{x}(t) = -L_f x(t) - l_{fl}u(t) \tag{14}$$
$$y(t) = -l_{fl}^T x(t) - L_l u(t). \tag{15}$$

So concluding:

$$L = \begin{pmatrix} L_f & l_{fl} \\ l_{fl}^T & L_l \end{pmatrix} = \left(\begin{array}{c|c} -A & -B \\ \hline -C & -D \end{array}\right) \tag{16}$$

From the article [15] we have:

Lemma 2. *If G is connected then L_f is positive definite.*

Because we can obtain the A, B, C, D matrices from the Laplacian matrix, the Laplacian of the graph will be very important to the further research.
When the network is split into leaders and followers we want to reach a consensus between the states of the agents. We want to steer all the states of the agents to a value $a \in \mathbb{R}$. This is done by the influence of the leaders. The leaders can be *time varying* ($\dot{x}_l \neq 0$) or *constant* ($\dot{x}_l = 0$). Therefore we want to reach consensus with the *agreement value* $(x_f(t), x_l(t))^T = a\mathbf{1}_n$. Following from (11) the dynamics of the followers are given by:

$$\dot{x}_f(t) = -L_f x_f(t) - l_{fl} x_l(t). \tag{17}$$

When leader agents are *constant* the states of the leaders are all the time fixed at the goal value a [17]. So we have $\dot{x}_l(t) = 0$ with initial value $x_l(0) = a\mathbf{1}_{n_l}$ and hence $x_l(t) = a\mathbf{1}_{n_l}$. Therefore the dynamics of the leader agents and the system output become (see (11)):

$$\dot{x}_l(t) = 0 \tag{18}$$
$$y(t) = Cx_f(t) + Da\mathbf{1}_{n_l}. \tag{19}$$

Now we can state the following theorem [15]:

Theorem 1. *Given fixed leader positions $x_l = a\boldsymbol{1}_{n_l}$, with $a \in \mathbb{R}$ the equilibrium point under the follower dynamics in* (17) *is*

$$\bar{x}_f(t) = -L_f^{-1} l_{fl} x_l = -L_f^{-1} l_{fl} a\boldsymbol{1}_{n_l}. \tag{20}$$

Proof. Put (17) equal to zero. From lemma (2), we know that L_f is invertible and hence (20) is well-defined. Hence the equilibrium point is unique. Moreover, since L_f is positive definite, this equilibrium is in fact globally asymptotically stable. □

Moreover by the row sum property of the Laplacian we know that

$$L_f \mathbf{1}_{n_f} + l_{fl}\mathbf{1}_{n_l} = 0 \Rightarrow l_{fl}\mathbf{1}_{n_l} = -L_f\mathbf{1}_{n_f}.$$

Therefore the unique equilibrium point in (20) can be written as $\bar{x}_f(t) = -L_f^{-1}(-L_f)a\mathbf{1}_{n_f} = a\mathbf{1}_{n_f}$. The corresponding output at the equilibrium can also be computed by using the row sum property of the Laplacian we obtain:

$$\bar{y}(t) = C\bar{x}_f(t) + Da\mathbf{1}_{n_l} = -l_{fl}^T a\mathbf{1}_{n_f} - L_l\mathbf{1}_{n_l}a \tag{21}$$
$$= (-l_{fl}^T\mathbf{1}_{n_f} + l_{lf}^T\mathbf{1}_{n_f})a = 0. \tag{22}$$

As a result from the previous theorem and above computations we can compute the *stationary state output*, this results in the following corollary:

Corollary 1. *The corresponding stationary state $\bar{x}(t)$ and stationary state output $\bar{y}(t)$ by a given constant input $\bar{u}(t) = a\boldsymbol{1}_{n_l}$ is given by:*

$$\bar{x}(t) = -A^{-1}B\bar{u}(t) = a\boldsymbol{1}_{n_f} \tag{23}$$
$$\bar{y}(t) = (-CA^{-1}B + D)\bar{u}(t) = 0. \tag{24}$$

The corresponding consensus value for all agents is $(x_f, x_l)^T = (a\mathbf{1}_{n_f}, a\mathbf{1}_{n_l})^T = a\mathbf{1}_n$. Recall that indeed consensus is reached, all states have the same value and we have $L(x_f, x_l)^T = 0$, because the consensus value must lie in the kernel of L which is given by ker$L$ =span$\{\mathbf{1}_n\}$.

This constant leader agent case is very specific. Leaders do not always have the same constant value, they can also have different constant values. Or more general, every value of a leader agent can vary freely in time, the so called *time varying leader* agents. Therefore we state the previous corollary in a more general way:

Corollary 2. *The corresponding stationary state $\bar{x}(t)$ and stationary state output $\bar{y}(t)$ by a constant input $\bar{u}(t)$ is given by:*

$$\bar{x}(t) = -A^{-1}B\bar{u}(t) \tag{25}$$

$$\bar{y}(t) = (-CA^{-1}B + D)\bar{u}(t). \tag{26}$$

Note that we do not reach consensus necessarily in this case. If the leaders have different constant values, i.e. they are not in consensus, then the follower agents may also converge to different values depending on there connections with the leaders. And hence we do not have a consensus situation.

For completeness I note that there are other ways to make a state space representation of a leader-follower network, see for example [10], [11], [15] and [16].

5 Balancing and reduction methods

Now we are able to write a multi-agent system with its corresponding graph into a state-space model. We have seen that in such a system we can reach consensus between the states of the agents by using the leader agents as a constant or time-varying input to the system so the follower agents will asymptotically converge to the value of the leader agents. More specifically in our case the states converge to the average of the initial values of the states. But what if we have a multi-agent system with too many agents. Then it will be desirable to reduce the number of agents and maintain a good representation of the full scale model. Of course, if we reduce we must have again a multi-agent system with less agents and a corresponding graph G_{red}. So the question that arises: are there reducing techniques which preserve the graph structure of the system? In this chapter we search for these techniques. Every technique is explained and tested with help of an example. We start with balancing techniques followed by reduction techniques. For a good overview I refer the reader to [1]. The following reduction techniques are treated:

- Balanced truncation.
- Balanced singular perturbation method.
- Passivity balancing.
- Krylov model reduction.

5.1 Balancing

We want to reduce the model for control purposes and in order to be able to better analyze the system. For simulations it is less intensive to work with a reduced model. The reduced model is required to have the same essential behavior as the full order systems. In terms of the state space representation of a system we wish to replace the system $\dot{x}(t) = Ax(t) + Bu(t)$, $y(t) = Cx(t) + Du(t)$ by $\dot{\bar{x}}(t) = \bar{A}\bar{x}(t) + \bar{B}u(t)$, $y = \bar{C}\bar{x} + \bar{D}u(t)$, where the $\bar{x}$ has less elements than the original state x.
Balancing is finding a basis wherein the states which are difficult to reach are also the states difficult to observe. Such a basis we find by simultaneously diagonalizing the controllability and observability Gramian. From a linear algebraic point of view balancing consists of the simultaneous diagonalization of two positive (semi) definite matrices. After balancing we can create reasonably reduced order models of systems by deleting the part of the state space that is least controllable and least observable [6].

For this balancing technique we need Gramians. A Gramian is a positive semidefinite matrix with a set of not negative real eigenvalues $\lambda_1 \geq \ldots \lambda_n \geq 0$. For balancing we need two special types of Gramians: the observability and controllability Gramian.

Definition 1. ***Gramians.*** *The controllability Gramian W and the observability Gramian M of the systems (A, B, C) are defined as the unique solutions of the Lyapunov equations*

$$AW + WA^T = -BB^T, \qquad A^T M + MA = -C^T C \tag{27}$$

where A, $-BB^T$ and $-C^TC$ are given square matrices and respectively W and M are unknown.

In some cases we can solve explicitly the two (dual) Lyapunov equations. To solve the equations we need that A is Hurwitz.

Definition 2. ***Hurwitz matrix.***
A square matrix A is called Hurwitz matrix (stable matrix) if every eigenvalue of A has strictly negative real part, that is, $Re[\lambda_i] < 0$ for each eigenvalue λ_i.

Now we can state the following theorem:

Theorem 2. *Suppose that A is Hurwitz. Then the observability Gramian is given by*

$$M = \int_0^\infty e^{A^T t} C^T C e^{At} dt \tag{28}$$

and the controllability Gramian by

$$W = \int_0^\infty e^{At} BB^T e^{A^T t} dt. \tag{29}$$

Note that these Gramians are only well defined for asymptotically stable systems (A is Hurwitz).

Definition 3. ***Balanced realization*** *We speak of a balanced realization of the system if the coordinates are chosen in such a way that for the transformed system M and W are equal and diagonal, i.e. $M = W = \Sigma$. Where Σ is a diagonal matrix [19].*

Theorem 3. *The eigenvalues of the product WM are invariant under state space transformations and hence they can be viewed as input/output invariants [19].*

If we have a balanced realization $W = M = \Sigma$ then Σ is a diagonal matrix with at the diagonal the Hankel singular values of the system [19], [30]. For our purposes we can define the Hankel singular values in the following way:

Definition 4. ***Hankel singular values [19]*** *Assume that A is Hurwitz. The Hankel singular values σ_i of the system are defined as*

$$\sigma_i = \sqrt{\lambda_i(WM)}, \quad \textit{for } i = 1, 2, \ldots, n \tag{30}$$

where $\lambda_i(WM)$ is the i'th eigenvalue of the matrix product WM. We also have the convention of ordering the Hankel singular values in such a way that $\sigma_i \geq \sigma_{i+1}$. For a more elaborate view on Hankel singular values and the singular value decomposition theorem see [8] and [2] or any other book on Linear Algebra.

Theorem 4.
- *The singular values of a matrix A and PAQ are identical for any orthogonal matrices P, Q.*
- *The rank of a matrix A equals the number of nonzero singular values of the matrix.*
- *If a matrix A is symmetric, then the singular values of A are the absolute values of its eigenvalues.*
- *If A is positive semidefinite then the singular values are the same as the eigenvalues.*

Proof can be found in any good book about Linear Algebra for example in [8].

We need a state space transformation in order to make the system balanced [20], a corollary in [6] says:

Corollary 3. *Suppose (A, B, C) is a controllable and observable realization there exists a state transformation T such that the equivalent realization $(\tilde{A}, \tilde{B}, \tilde{C}) = (TAT^{-1}, TB, CT^{-1})$ satisfies*

$$\tilde{W} = \tilde{M} = \Sigma$$

with $\Sigma > 0$ diagonal and with

$$\tilde{W} = TWT^T \qquad \tilde{M} = (T^T)^{-1}MT^{-1} \tag{31}$$

respectively the controllability and observability Gramians transformed in the new coordinates.

So for our problem we can always find the balanced realization when we can describe our system in a state space representation with (A, B, C).

This was a brief introduction to balancing. It is time to take a look at our situation. In this thesis we look at a special kind of systems, the so called gradient systems [26]. Consider a linear system (A, B, C, D) given as

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{32}$$
$$y(t) = Cx(t) + Du(t). \tag{33}$$

Definition 5. *Such a system is called a gradient system whenever there exists an invertible symmetric matrix G satisfying*

$$A^TG = GA, \qquad B^TG = C. \tag{34}$$

We work with an undirected graph, i.e. $L = L^T \Rightarrow A = A^T, B^T = C$. It follows that our system is a gradient system, with $G = I$ [26]. If we take a look at the two Lyapunov functions (27) we get:

$$AW + WA^T = -BB^T = -C^TC = A^TM + MA \tag{35}$$
$$\Rightarrow AW + WA^T = A^TM + MA \tag{36}$$
$$\Rightarrow W = M. \tag{37}$$

From the theory on gradient systems we also know that a controllable and observable system is a gradient system if and only if its transfer matrix $H(s) := C(Is - A)^{-1}B + D = H(s)^T$ is symmetric. Moreover we know that G is unique whenever the system is controllable and observable [26]. The system written in balanced coordinates, $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$ is again a gradient system with $G = I$. During balancing we want to keep $W = M$ and diagonalize them with opposite transformations as in corollary 3. After balancing we have that $\tilde{W} = \tilde{M} = \Sigma$ with Σ a diagonal matrix, with at the diagonal the Hankel singular values ordered from high to low. Therefore we have $TWT^T = (T^T)^{-1}MT^{-1}$, now because we already have $W = M$ we obtain $T^T = T^{-1}$ hence $T^TT = I = TT^T$. Thus T is a unitary transformation. In other words T is an orthonormal transformation matrix which diagonalizes W. To diagonalize a symmetric positive definite matrix W we use a matrix P which columns are the eigenvectors of W, i.e. $P^{-1}WP = \Sigma$. From this observations the following important corollary arises:

Corollary 4. *If we have a (special) gradient system with $G = I$ we have:*

- *The transformation matrix $T = P^{-1} = P^T$, where P's columns are the eigenvectors of W.*
- *The Hankel singular values of the system are equal to the singular values of W, which are equal to the eigenvalues of W.*

Proof. See results above this corollary. We have $TWT^T = \text{Diag}(\sigma_1, \ldots, \sigma_n)$ and $W = M$ thus T is symmetric, unitair and orthonormal, more specific we have $WT^T = T^T\text{Diag}(\sigma_1, \ldots, \sigma_n)$. Hence $Wp_i = p_i\sigma_i$, where p_i is the ith column of P (ith eigenvector of W). Hence we have $\lambda_i = \sigma_i$ for $i \in \{1, \ldots, n\}$. □

After balancing we want to reduce the system. We apply two reduction techniques on the balanced system: balanced truncation and singular perturbation balanced truncation. First I give a small theoretic introduction to this two reduction methods and then I will work out an example for both. For the transformation matrix T in the general case I refer to [20], [1] and [21].

Beginning with balanced truncation we order the Hankel singular values from high to low, that is $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$. Now if we want to reduce the order of the system we can do this by only looking at the part of the system which corresponds to the highest Hankel singular values. If holds that for some i that $\sigma_i >> \sigma_{i+1}$ we will reduce the system to those parts corresponding to the larger Hankel singular values $(\sigma_1, \ldots, \sigma_i)$, i.e. we reduce the dimension of the system by discarding those parts of the system which correspond to the smaller Hankel singular values. The remaining system still captures the controllability and observability behavior of the original system [19].

Moreover you can always write a system in balanced coordinates and truncated the part corresponding to the lowest Hankel singular values, hence a balanced truncation is always possible for proofs I refer to [6] and [1]. In balanced coordinates we have:

$$W = M = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{pmatrix} \tag{38}$$

where the Hankel singular values are at the diagonal and are ordered. Now suppose we want to find a realization with order less than n, say r (so $r < n$). We assume that the strict inequality $\sigma_r > \sigma_{r+1}$ is satisfied. So we really only want to truncate at a clear separation of the singular values. The full order model is partitioned as:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \quad C = \begin{pmatrix} C_1 & C_2 \end{pmatrix}, \quad D = D \tag{39}$$

where $A_{11} \in \mathbb{R}^{r \times r}$. So our full model is given by:

$$\begin{pmatrix} \dot{x}^1 \\ \dot{x}^2 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} u \tag{40}$$

$$y = \begin{pmatrix} C_1 & C_2 \end{pmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} + Du \tag{41}$$

where (A, B, C, D) are written in balanced coordinates and are partitioned as above and $x^1 \in \mathbb{R}^r, x^2 \in \mathbb{R}^{n-r}$. The balanced truncation method now sets the states with low Hankel singular values $x^2 = \mathbf{0}$ and discard all differential equations of $\dot{x}^2$. The reduced order approximation now becomes

$$\dot{x}^1 = A_{11}x^1 + B_1u \tag{42}$$

$$y = C_1x^1 + Du \tag{43}$$

with transfer matrix

$$\hat{G}_r = C_1(Is - A_{11})^{-1}B_1 + D \tag{44}$$

in which we need that A_{11} is still Hurwitz. This is indeed the case [6].

Proposition 1.

- *The matrix A_{11}, as defined above, is Hurwitz.*
- *The realization (A_{11}, B_1, C_1, D) is balanced with Hankel singular values $\sigma_1, \ldots, \sigma_r$.*
- *The reduced order model is still a gradient system with $\hat{G} = I_r$ [26].*

By truncating we have now found a balanced state space realization of order $r < n$.

The second method is *singular perturbation balanced truncation.* In this case we will also obtain a reduced order gradient system [26]. The method works as follows. Set the differential equations $\dot{x}^2 = 0$, i.e. $0 = A_{21}x^1 + A_{22}x^2 + B_2u$. By simple computation we have $x^2 = -A_{22}^{-1}A_{21}x^1 - A_{22}^{-1}B_2u$, plug this in the differential equations for $\dot{x}^1$ and in the equation for output y . We now have reduced the system by singular perturbation balanced truncation and we obtain:

$$\dot{x}^1 = (A_{11} - A_{12}A_{22}^{-1}A_{21})x^1 + (B_1 - A_{12}A_{22}^{-1}B_2)u \tag{45}$$

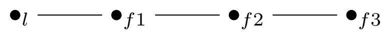
$$y = (C_1 - C_2A_{22}^{-1}A_{21})x^1 + (D - C_2A_{22}^{-1}B_2)u \tag{46}$$

so the reduced linear system matrices $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ are given by $\hat{A} = (A_{11} - A_{12}A_{22}^{-1}A_{21})$, $\hat{C}^T = \hat{B} = (B_1 - A_{12}A_{22}^{-1}B_2)$ and $\hat{D} = D - C_2A_{22}^{-1}C_2^T$. Note that $\hat{A}$ is given by the Schur complement of block A_{22} of matrix A. Therefore $\hat{A}$ is again symmetric, for proof see [26].

I will use both balancing reduction methods on the same graph to obtain a reduced model in the following example.

5.1.1 Examples

We look at the following graph with one leader agent (most left vertex) and three follower agents (the other three)



I will call this graph G_1. First I will look at the case when there are no weights to be taken into account. Then I will rearrange the vertices in such order that the first three are the followers and the last one is the leader agent. So we get $V = \{f_1, f_2, f_3, l\} = \{1, 2, 3, 4\}$ and the edges set $E = \{(4,1); (1,4); (1,2); (2,1); (2,3); (3,2)\}$. The Laplacian becomes:

$$L = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} = L^T \tag{47}$$

Now split L into L_f, l_fl, l_{fl}^T and L_l like in (11) and compose the state space representation like in (14). We obtain:

$$\dot{x}(t) = \begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} u(t) \tag{48}$$

$$y(t) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} x(t) - u = x_1(t) - u(t). \tag{49}$$

$$\tag{50}$$

The A matrix of the system has three negative solutions (eigenvalues) to $det(A - \lambda I) = 0$: $\lambda_1 \approx -3,24698, \lambda_2 \approx -1,55496$ and $\lambda_3 \approx -0,198062$. From this we can conclude that matrix A is Hurwitz. Now we compute the Gramians. Because $A = A^T$ and $B^T = C$ we have $W = M$. Hence, solving one of the Lyapunov equations by hand or in Matlab, we obtain

$$W = M = \begin{pmatrix} 0.3103 & 0.1207 & 0.0690 \\ 0.1207 & 0.1034 & 0.0862 \\ 0.0690 & 0.0862 & 0.0862 \end{pmatrix}.$$

By computing the eigenvalues of W we also have the Hankel singular values of the system, they are given by $\{0.00379763, 0.09838144426, 0.39782092088\}$. Now we have to put the singular values in the order from highest value to lowest. So we get: $\{0.39782092088, 0.09838144426, 0.00379763\} = \{\sigma_1, \sigma_2, \sigma_3\}$, with $\sigma_1 > \sigma_2 > \sigma_3$. Following corollary 3 we can find a orthonormal T in order to transform to a diagonal matrix Σ. We can diagonalize a matrix by putting the eigenvectors as columns in a matrix P, now because we ordered the eigenvalues we obtain:

$$P = \begin{pmatrix} -0.8453 & 0.5144 & -0.1447 \\ -0.4367 & -0.5090 & 0.7418 \\ -0.3079 & -0.6902 & -0.6548 \end{pmatrix}.$$

Take $T = P^{-1}$, with this T we can transform the system in balanced coordinates and then obtain the Gramian in diagonal form (with corollary 3). Recall and note that T is a unitary matrix ($TT^T = I = T^T T$). Also the Laplacian matrix transforms into:

$$L_{new} = \begin{pmatrix} T & 0 \\ 0 & 1 \end{pmatrix} L \begin{pmatrix} T^{-1} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.8980 & -0.8762 & 0.3046 & 0.8453 \\ -0.8762 & 1.3446 & -0.7285 & -0.5144 \\ 0.3046 & -0.7285 & 2.7574 & 0.1447 \\ 0.8453 & -0.5144 & 0.1447 & 1 \end{pmatrix}.$$

Now we have obtained the system in balanced coordinates we are going to reduce the model. The biggest gap is between σ_2 and σ_3. So a good choice will be to reduce the system to a second order system.

$$\begin{aligned} \dot{x}(t) &= \left(\begin{array}{cc|c} -0.898 & 0.8762 & -0.3046 \\ 0.8762 & -1.345 & 0.7285 \\ \hline -0.3046 & 0.7285 & -2.757 \end{array}\right) x(t) + \left(\begin{array}{c} -0.8453 \\ 0.5144 \\ \hline -0.1447 \end{array}\right) u(t) \\ &= \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) x(t) + \left(\begin{array}{c} B_1 \\ \hline B_2 \end{array}\right) u(t). \end{aligned}$$

So by using *balanced truncation* (putting $x_3 = 0$ and discard the third differential equation) we end up with the reduced system (A_r, B_r, C_r, D_r) given by:

$$\begin{aligned} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} &= \begin{pmatrix} -0.898 & 0.8762 \\ 0.8762 & -1.345 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} -0.8453 \\ 0.5144 \end{pmatrix} u(t) \\ y(t) &= \begin{pmatrix} -0.8453 & 0.5144 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - u(t). \end{aligned}$$

If A_r, B_r, C_r, D_r would correspond to $L_{red1} = \begin{pmatrix} -A_r & -B_r \\ -C_r & -D_r \end{pmatrix}$:

$$L_{red1} = \begin{pmatrix} 0.898 & -0.8762 & 0.8453 \\ -0.8762 & 1.345 & -0.5144 \\ 0.8453 & -0.5144 & 1 \end{pmatrix}.$$

Take a close look at the numerical values in this L_{red1} matrix and see that both the row and column sum are not equal to zero. So this L_{red1} is not a Laplacian, therefore the reduced system cannot be represented by a graph. The Gramians of this reduced system however, are exactly as we want, namely:

$$W_{red1} = M_{red1} = \begin{pmatrix} 0.3978 & 0 \\ 0 & 0.0984 \end{pmatrix}.$$

The *second way* to obtain a reduced order model is by the singular perturbation balanced truncation, we set $\dot{x_3} = 0$ we obtain (by direct computation):

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\end{pmatrix} = (A_{11}-A_{12}A_{22}^{-1}A_{21})x_1 + (B_1 - A_{12}A_{22}^{-1}B_2)u = \begin{pmatrix}-0.8644 & 0.7957\\ 0.7957 & -1.152\end{pmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + \begin{pmatrix}-0.8293\\ 0.4761\end{pmatrix}u$$

$$x_3 = -A_{22}^{-1}(A_{21}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + B_2u).$$

If we deduce the associated Laplacian from this reduced system we get:

$$L_{red2} = \begin{pmatrix}0.8644 & -0.7957 & 0.8293\\ -0.7957 & 1.1522 & -0.4761\\ 0.8293 & -0.4761 & 1\end{pmatrix}.$$

Again, unfortunately L_{red2} is again not a Laplacian matrix, so this reduced model can not be represented by a graph.

By these two examples of balanced reduction techniques we conclude that in general balanced reduction is not a good method to obtain a reduced system which still has a graph corresponding to it. Therefore we try some other reduction methods.

5.2 Passivity balancing

For this type of balancing we start with a passive system. Fortunately our system of the previous example is passive (for information about passive systems I refer to [14]), so we can and will use in this example the same L corresponding to graph G_1 and hence the same system matrices. I treat this type of balancing immediately with an example because we already know that it is unlikely that balancing results in a new Laplacian matrix. And also in this case we will see that balancing does not result into a Laplacian. Now we will start with the example and take a look at the following matrix [27]:

5.2.1 Example

$$\begin{pmatrix}A^TQ+QA & QB-C^T\\ B^TQ-C & -D-D^T\end{pmatrix}. \tag{51}$$

Because $A = A^T, B^T = C$ and we want that the matrix is negative semi definite we obtain:

$$\begin{pmatrix}AQ+QA & QB-B\\ CQ-C & -2D\end{pmatrix} \leq 0. \tag{52}$$

From [2] we know that this holds when $-D-D^T \leq 0$ and its Schur complement $AQ+QA-(QB-B)(-2D)^{-1}(CQ-C) \leq 0$. Notice that this second inequality is exactly the (general) continuous-time algebraic Riccati equation [1] and [27]:

$$A^TQE + E^TQA - (E^TQB+S)R^{-1}(B^TQE+S^T) + X = 0 \tag{53}$$

with $A = A, E = I, S = -B = -C^T, R = -2D$ and $X = 0$. Note that in this example D is invertible because it is just a number different from 0. Now we can find the unique minimum stabilizing solution with the Matlab command care. By computation:

$$Q_{min} = \begin{pmatrix}-0.4016 & -0.3120 & -0.2864\\ -0.3120 & -0.3414 & -0.3465\\ -0.2864 & -0.3465 & -0.3670\end{pmatrix}. \tag{54}$$

If we multiply equation (52) with -1 and compute the minimum solution of the corresponding Riccati equation, we get the maximum solution, Q_{max} of equation (53). Therefore we have to give Matlab the following command: Q_{max} = care(-A,-B,0,2D,B, I).

$$Q_{max} = \begin{pmatrix} -17.5984 & 51.2787 & -34.6802 \\ 51.2787 & -219.4583 & 167.1797 \\ -34.6802 & 167.1797 & -133.4995 \end{pmatrix}. \tag{55}$$

We are however interested in Q_{max}^{-1}. This turns out to be the same matrix as Q_{min}. So:

$$Q_{max}^{-1} = Q_{min}.$$

We want to balance the matrices Q_{min} together with Q_{max}^{-1} in the same way as in the previous example. The system matrices (A, B, C, D) also transform in the same way as in the previous example. We are now interested if the Laplacian matrix has transformed into an other Laplacian matrix for the transformed system. We take $T = P^{-1} = P^T = \begin{pmatrix} v_3 & v_2 & v_1 \end{pmatrix}^T$ with v_1, v_2, v_3 the eigenvectors of Q_{min} corresponding to the eigenvalues of Q_{min} i.e. $\lambda_1, \lambda_2, \lambda_3$.

Now we diagonalize the two matrices (balancing) with T as is written in corollary 3:

$$\Sigma = TQ_{min}T^T = (T^T)^{-1}Q_{max}^{-1}T^{-1} = \begin{pmatrix} -0.0028 & 0 & 0 \\ 0 & -0.1073 & 0 \\ 0 & 0 & -1.0000 \end{pmatrix}. \tag{56}$$

Now if we transform the Laplacian directly with the transformation matrix:

$$L_{new} = \begin{pmatrix} T & 0 \\ 0 & 1 \end{pmatrix} L \begin{pmatrix} T^{-1} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2.8517 & 0.7135 & 0.1025 & 0.1775 \\ 0.7135 & 1.8150 & 0.4601 & 0.7970 \\ 0.1025 & 0.4601 & 0.3333 & 0.5774 \\ 0.1775 & 0.7970 & 0.5774 & 1.0000 \end{pmatrix}.$$

And again, unfortunately, this L_{new} matrix is not a Laplacian. Also this type of balancing does not result into a Laplacian. And therefore we do not even have to take the effort to reduce the system from the balanced system because we already know that the reduced system will not be corresponding to a graph.

5.3 Krylov model reduction

Krylov model reduction is an other type of model reduction. There are several types of Krylov model reduction methods, for example Arnoldi and Lanczos [1], [22]. Look at the system of the form

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{57}$$

and compute the controllability matrix

$$V = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \quad \in \mathbb{R}^{m \times n}. \tag{58}$$

The idea behind Krylov model reduction is to reduce the matrix V with some columns to a matrix

$$V_{red} = \begin{bmatrix} B & AB & A^2B & \dots & A^{r-1}B \end{bmatrix} \quad \in \mathbb{R}^{m \times r} \tag{59}$$

with $r < n$, which is called the Krylov subspace. This reduced V_{red} matrix now has to be made orthonormal with, for example, the Gram-Schmidt process [28]. Then we obtain a matrix $\tilde{V}_{red}$ with orthonormal columns. With this matrix we can now compute our reduced system $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ in the following way [22]:

$$\hat{A} = \tilde{V}_{red}^T A \tilde{V}_{red} \tag{60}$$

$$\hat{B} = \tilde{V}_{red}^T B \tag{61}$$

$$\hat{C} = C\tilde{V}_{red} \tag{62}$$

$$\hat{D} = D. \tag{63}$$

This short theoretical overview of the Krylov method is enough to reduce our example graph G_1. For more information about the Krylov model reduction I refer to [1], [22] and [24].

5.3.1 Example

Again our system is defined by the Laplacian which corresponds to the four agent system with graph G_1. We write it in the form

$$\dot{x}(t) = Ax(t) + Bu(t) = \begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} u(t). \tag{64}$$

We can now build up the controllability matrix V by

$$V = \begin{bmatrix} B & AB & A^2B \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 5 \\ -4 \\ 1 \end{pmatrix} \end{bmatrix}. \tag{65}$$

I reduce this V to V_{red} which have second order i.e. $V_{red} = \begin{bmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix} \end{bmatrix} = [\tilde{v}_1 \; \tilde{v}_2]$. Now I am going to use Gram-Schmidt process to orthonormalize the set of vectors v_{red}.

$$u_1 = \tilde{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{66}$$

$$u_2 = \tilde{v}_2 - \text{projection}_{u_1}(\tilde{v}_2) = \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix} - \frac{\begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}}{\sqrt{1}^2} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}. \tag{67}$$

Observe that indeed $< u_1, u_2 >= 0$. Now we need to normalize the vectors u_1, u_2 by dividing them by their own length and we stack them together to get $\tilde{V}_{red}$. We have obtained:

$$\tilde{V}_{red} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}. \tag{68}$$

Computing the new reduced A matrix in the following way:

$$\hat{A} = \tilde{V}_{red}^T A \tilde{V}_{red} = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}. \tag{69}$$

According to page 62 of Polyuga [22] the system matrices changes in the following way: $\hat{A} = \tilde{V}_{red}^T A \tilde{V}_{red}$, $\hat{B} = \tilde{V}_{red}^T B$, $\hat{C} = C\tilde{V}_{red}$, $\hat{D} = D$. Computations result in the following reduced system:

$$\begin{aligned} \hat{A} &= \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} \\ \hat{B} &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \hat{C} &= \begin{pmatrix} 1 & 0 \end{pmatrix} \\ \hat{D} &= -1. \end{aligned}$$

The corresponding 'Laplacian' of this reduced system is

$$L_{red3} = \left(\begin{array}{cc|c} 2 & -1 & -1 \\ -1 & 2 & 0 \\ \hline -1 & 0 & 1 \end{array}\right). \tag{70}$$

This L_{red3} matrix is almost a Laplacian matrix. Only the second row doesn't sum up to 0. The diagonal elements are positive and the off diagonal elements are negative. The best reducing method up to now.

We conclude this chapter with a section about conditions on a transformation matrix T.

5.4 Conditions on the transformation matrix T

In this section I will first derive some necessary conditions on the coordinate transformation matrix T which was the matrix introduced to diagonalize the Gramian matrices as in (3). By symmetry of the Laplacian ($L = L^T$) in our case we have :

$$A = A^T, \quad B^T = C, \quad D = D.$$

Recall that it directly follows that the controllability and observability Gramians are equal ($W = M$, see definition 1). And recall that T should be an orthogonal matrix ($TT^T = T^T T = I$), hence the inverse of T is given by the transposed of $T^{-1} = T^T$. It is clear that T should be square and invertible. We can derive the following conditions on state transformation matrix T:

Theorem 5. ***Conditions on*** T *We can divide the conditions on T into four blocks. Block 1: the conditions on T that follows from that it should diagonalise the matrix W. Block 2: the conditions on T that follows from the row/column sum property of the (new/transformed) Laplacian. Block 3: the conditions on T that follows from the fact that the diagonal entries of the (new/transformed) Laplacian must be nonnegative and the off-diagonal entries must be nonpositive. Block 4: the transformed Lyapunov equation. Where the conditions in block 2 and 3 all follow from that we want that TLT^T is a Laplacian matrix.*

1.
   - $TT^T = I = T^T T$ *by orthogonality.*
   - $T^T = T^{-1}$ *by orthogonality.*
   - *Columns and rows of T are orthogonal unit vectors. The dot product of two columns/rows equals zero and the length of a column/row vector equals one. By orthogonality.*
   - $\Sigma = TWT^T = TMT^T$ *and so* $\Sigma T = TM$ *and* $T^T \Sigma = MT^T$.
   - $T = P^T$ *where the columns of P are the eigenvectors of W.*
2.
   - $\mathbf{1}^T T = -\mathbf{1}^T CA^{-1}$ *(follows from the column sum of the new Laplacian).*
   - $\mathbf{1}^T TB = -\mathbf{1}^T D$ *(follows from the column sum of the new Laplacian).*
   - $T^T \mathbf{1} = -A^{-1} B\mathbf{1}$ *(follows from the row sum of the new Laplacian).*
   - $CT^T \mathbf{1} = -D\mathbf{1}$ *(follows from the row sum of the new Laplacian).*
3.
   - $-TAT^T$ *and* $-D$ *must be positive on the diagonal and negative off the diagonal.*
   - *Every element of* $-CT^T$ *and* $-TB$ *should be negative.*
4. $T(AM + MA)T^T = TAMT^T + TMAT^T = TAT^T\Sigma + \Sigma TAT^T = \tilde{A}\Sigma + \Sigma\tilde{A} = -\tilde{C}^T\tilde{C}.$

Note that the theorem does not say that it is possible to find a T that satisfies to all conditions of the theorem.

Theorem 6. ***Spectral Theorem.*** *If matrix A is a square symmetric $n \times n$ matrix the eigenvalues of A are real. There is an orthogonal matrix P such that*

$$P^T AP = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix}.$$

Corollary 5. *The columns of P are the eigenvectors of matrix A.*

In our case we have that $W = M = W^T = M^T$. So in order to diagonalize this symmetric Gramian we can take for $T = P^T$ with P the matrix which columns are the eigenvectors of W. With help of corollary 3 we have diagonalized our Gramians and we can write the system into balanced coordinates with the same coordinate transformative T.

As we have seen in the examples in this chapter it is in general not true that a system written in balanced coordinates corresponds to a Laplacian. Therefore we will not look any further for a transformation matrix T which diagonalise W and M as in (3).

6 Merge Vertices

In this chapter we will find a different kind of T, which reduce the system in such a way that there is a Laplacian corresponding to it. This T simply combines two vertices. After describing the method we will analyze if this method approaches the full order system.

In this section we will develop a method to combine vertices in order to get a system with a reduced number of vertices. The method we use is just a transformation that combines the two (or more vertices) we want to combine, just adding their weights in the combined Laplacian matrix. We will see that this transformed and reduced Laplacian is still a Laplacian matrix and hence it corresponds to a weighted reduced graph. The question is now: how does this reduced graph look like? And what is the behavior of the corresponding system in comparison with the full order system? Is there anything we can say about the error which is induced by reducing the number of vertices in this specific way? And what about the static state ($\dot{x} = 0$) response of the reduced systems in comparison with the full order system?

6.1 Merge vertices method

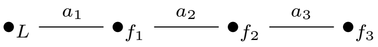
In this section we introduce a quite different T. This T matrix is not square and only consist of 0's and 1's as entries. From now on we use T for this type of matrices. T is introduced further in the following theorem.

Theorem 7. *If we want to combine two adjacent follower vertices i and j of an undirected given graph G with n vertices by the following transformation*

$$L_r = T^t L T \tag{71}$$

where L is the Laplacian matrix. T is an $n \times (n-1)$ matrix which can be obtained by taking the $n \times n$ identity matrix and add in the $\min\{i,j\}$'th column a 1 on both the i'th row as the j'th row if there is not a 1 already present. After that delete the $\max\{i,j\}$th column of this identity matrix.
Then L_r is a Laplacian matrix.
Note: we assume that the vertices are ordered i.e. the follower vertices are numbered 1 to $n - |\alpha|$ and the $n - |\alpha| + 1$ to n are the leader vertices.

Example. Consider the graph G_1 with weights, we want to combine vertex f_1 with f_2 to get the combined vertex f_{12}. We start with the Laplacian of the graph G_1.

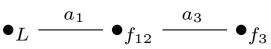


$$L = \begin{pmatrix} a_1 + a_2 & -a_2 & 0 & -a_1 \\ -a_2 & a_2 + a_3 & -a_3 & 0 \\ 0 & -a_3 & a_3 & 0 \\ -a_1 & 0 & 0 & a_1 \end{pmatrix}. \tag{72}$$

Now apply the transformation in order to get the reduced Laplacian L_{12}

$$L_{12} = T^t L T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} L \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} a_1 + a_3 & -a_3 & -a_1 \\ -a_3 & a_3 & 0 \\ -a_1 & 0 & a_1 \end{pmatrix}. \tag{73}$$

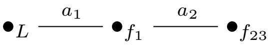
Observe that L_{12} is clearly a Laplacian and corresponds to the following graph



We see that this corresponds with just leaving out the edge between f_1 and f_2. In the same manner vertex f_2 and f_3 of G_1 can be combined; after transformation of L we obtain the new reduced Laplacian:

$$L_{23} = \begin{pmatrix} a_2 + a_1 & -a_2 & -a_1 \\ -a_2 & a_2 & 0 \\ -a_1 & 0 & a_1 \end{pmatrix} \tag{74}$$

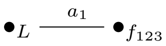
this Laplacian L_{23} corresponds to the graph:



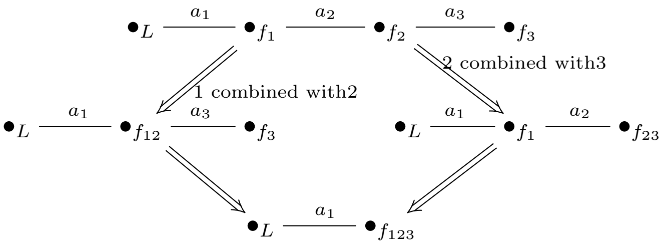
Again we see that this graph could also be obtained by just leaving out the edge between vertex f_2 and f_3 from the original graph G_1.
Now we want to check if combining vertices commutes i.e. do we obtain the same graph if we first combine f_1 with f_2 and then f_{12} with f_3 as when we first combine f_2 with f_3 and then f_1 with f_{23}? By simple computation we will see that:

$$\begin{aligned} L_{123} &= T_{12,3}^t T_{1,2}^t L T_{1,2} T_{12,3} = T_{1,23}^t T_{2,3}^t L T_{2,3} T_{1,23} && (75) \\ &= \begin{pmatrix} a_1 & -a_1 \\ -a_1 & a_1 \end{pmatrix} && (76) \end{aligned}$$

corresponding to the reduced graph (figure 6.1) :



The following picture gives a summary of the example above, it shows that reducing the graph by combining vertices is a commuting operation:

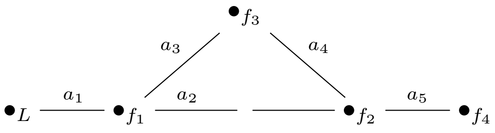


Now we know what is happening when we combine vertices, we can take a look at the stationary state response of the corresponding system. The stationary state response (i.e. when $\dot{x} = 0$) is given by $y = (-CA^{-1}B + D)\bar{u}$. When we compute this response for the full order system of this example we see that $y = 0$, y is a scalar because we have only one leader agent in G_1, for all possible weights a_1, a_2, a_3 and all constant inputs $\bar{u}(t)$.
Also the steady state responses of the reduced systems $y_{12} = y_{23} = y_{123} = 0$ for all edge weights and constant input functions $\bar{u}$. We combine this result in a theorem:

Theorem 8. *For a graph which has one leader vertex and $n-1$ follower vertices in one line the corresponding system has a steady state response of $y = 0$. All the corresponding systems of all possible reduced graphs, by combining vertices, have also steady state response of $y = 0$.*

This follows from the fact that $-CA^{-1}B + D$ is -1 times a Laplacian L. This results and above theorem follow from the theorems and theory which I introduce in a following chapter called *stationary state output.*

Now we only looked at a graph which had all vertices connected in one line. But what happens if we want to combine two vertices which are adjacent to the same vertex? For example we want to combine vertex f_1 with f_2 in the following 'triangle' graph:



Now take as transfer matrix $T_{12} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$. We obtain:

$$L_{12} = T_{12}^t L T_{12} = \begin{pmatrix} a_1 + a_3 + a_4 + a_5 & -a_3 - a_4 & -a_5 & -a_1 \\ -a_3 - a_4 & a_3 + a_4 & 0 & 0 \\ -a_5 & 0 & a_5 & 0 \\ -a_1 & 0 & 0 & a_1 \end{pmatrix}. \tag{77}$$

Clearly this matrix is a Laplacian matrix. We see that by combining two vertices (f_1 and f_2) which have a neighbor in common (f_3) that the edge and weight between f_1 and f_2 is 'deleted' and that the weights of the edges from f_1 to f_3 and f_2 to f_3 is added to obtain the weight on the edge between f_3 and the new formed combined vertex f_{12}.
All possibilities to combine two vertices have now been treated. In the following theorem this is summarized. Note that combining a leader with a follower vertex or a leader with a leader vertex results in the same combination. The only choice is: is the new vertex a leader or a follower vertex? I decide that it will be a leader vertex.

Theorem 9. ***Combining vertices.*** *There are a couple of ways in which two vertices can be combined. For completeness I will treat them all in this theorem.*

- *If we want to combine some vertices which are not neighbors but have a neighbor in common, then the combined vertex will have an edge to this common neighbor with as edge weight the sum of all the edge weights to this common neighbor of the combined vertices.*
- *If we want to combine two vertices which are neighbors, the edge or edges between them are removed and the combined vertex has edges to all vertices which where adjacent to one of the two vertices, the weights on these edges remain unchanged or are added when they have a common neighbor.*
- *If we want to combine two vertices which are not adjacent nor have a neighbor in common then the combined vertices has all the edges from both vertices. In this way circle graph may arise.*
- *Combining vertices is commuting. So it doesn't matter in which order the vertices are combined.*
- *When combining vertices we do not have to differentiate between leader and follower vertices. The only difference is the position in the Laplacian matrix. The choice which vertex is a leader influence the new system matrices A, B, C, D.*

Proof. Take an $n \times n$ complete graph with along the edges the weights $a_1, \dots, a_{n^2-(n-1)*2}$. The corresponding graph Laplacian is given by:

$$L = \begin{pmatrix} a_1+a_2+\ldots+a_{n-1} & -a_1 & -a_2 & -a_3 & -a_4 & \cdots & -a_{n-1} \\ -a_1 & a_1+a_n+\ldots+a_{2n-3} & -a_n & -a_{n+1} & -a_{n+2} & \cdots & -a_{2n-3} \\ -a_2 & -a_n & a_2+a_n+a_{2n-2}+\ldots+a_{3n-5} & -a_{2n-2} & -a_{2n-1} & \cdots & -a_{3n-5} \\ -a_3 & -a_{n+1} & -a_{2n-2} & a_3+a_{n+1}+a_{2n-2}+a_{3n-4}+\ldots+a_{4n-7} & -a_{3n-4} & \cdots & -a_{4n-7} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -a_{n-1} & -a_{2n-3} & -a_{3n-5} & -a_{4n-7} & -a_{5n-9} & \cdots & -a_{n^2-(n-1)*2} \end{pmatrix}$$

This is the most general case we can take, note that if an edge does not exist we can just set the weight along that edge equal to zero. The corresponding Laplacian has on the diagonal in each row the sum of $n-1$ weights. The sum of every row in the Laplacian is equal to zero. Because it is an undirected graph the column sums will be zero. Now if we combine two, three, four, five, ... vertices we first have to sum up the columns of the combining vertices. Therefore the weight of the edges between the vertices to be combined cancels out, because the weight is positively contributing in the diagonal entry of the matrix and negatively contributing in the off diagonal entry. This means that the edges between the vertices which are combined are discarded.
By summing up the columns of the combing vertices, we can go from this new vertex to all neighbors of the combined individual vertices. If we could go from one or more vertex of the combined vertices to

the same neighbor, the weights are simply added to this new edges, as can be seen from the Laplacian matrix.
Then we sum up the rows of the combing vertices in the Laplacian (by multiplying with T^t). We obtain that all the edges previously going to the apart vertices now go to the new combined vertex. Here we also see that if the combined vertices have a same vertex as neighbor then the new vertex gets an edge to this vertex with the sum of the weights of the previous edges. This completes the proof. □

Remark: Every symmetric positive semidefinite matrix L with diagonal elements ≥ 0 and the off-diagonal elements ≤ 0 and L has zero row and zero column sums is a Laplacian matrix [4].

Now we wonder why we only look to this kind of combinations of vertices and why are we not looking to a weighted combination of two or more vertices. The following theorem says that the above combination is the only possibility and I will give a proof immediately:

Theorem 10. *The only way to combine two or more vertices and get a new, reduced, Laplacian matrix is to use the above introduced matrix T.*

Proof. I will proof this by showing in a graph with four vertices that it is in general not possible to combine follower vertex f_1 and f_2 in any other way. From this proof the proof of the general $n \times n$ case can be deduced. Take L the Laplacian for a four vertex complete graph. And $T_{12} = \begin{pmatrix} K & 0 & 0 \\ L & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, with K and L arbitrary numbers. By simple computation we obtain:

$$L_{12} = T_{12}^t L T_{12} = \begin{pmatrix} K^2(a_1+a_2+a_3) - 2KLa_1 + L^2(a_1+a_4+a_5) & -a_2K - a_4L & -a_3K - a_5L \\ -a_2K - a_4L & a_2+a_4+a_6 & -a_6 \\ -a_3K - a_5L & -a_6 & a_3+a_5+a_6 \end{pmatrix}$$

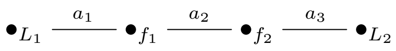
From the remark we know that the column and row sums should be 0. This only happens when

$$K = L = 1.$$

This rules out all the other possible combinations of vertices. It is also easy to see that it does not matter which vertices we combine and how many; the corresponding transformation matrix T should always exist of only ones at the appropriate entries. □

6.2 Merge vertices Hankel singular value error analysis

To set up an error estimate, we are going to look at the Hankel singular values (HSV). Recall $\sigma_i = \sqrt{\lambda_i(WM)}$ where W and M are the controllability and observability Gramians we also know that A is Hurwitz (i.e. $Re(\lambda_i) < 0$). Computing these HSV is the same as computing the eigenvalues of W, as we have seen. The idea is to take a look at the highest singular value of the full order system and compare this to the highest singular value of the corresponding reduced system. This is relevant to do because the highest singular value resembles the part that has the most influence on the system. Later on, in the chapter about the stationary state output, I'm also going to use the stationary state output to determine how good this reduction technique is.
I start with graph G_2 which is the 'line graph' with four vertices, but now with two leader agents and is pictured below.



The Laplacian matrix corresponding to graph G_2 is given by:

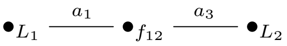
$$L = \begin{pmatrix} a_1+a_2 & -a_2 & -a_1 & 0 \\ -a_2 & a_2+a_3 & 0 & -a_3 \\ -a_1 & 0 & a_1 & 0 \\ 0 & -a_3 & 0 & a_3 \end{pmatrix}. \tag{78}$$

With Matlab I computed for some weights a_1, a_2, a_3 the HSV for the full system of four vertices and of the reduced system with three vertices (one HSV). When I compare the results I see that the highest HSV does not differ that much, so it seems to be a good reduction. To test this further I will analytically work out the HSV for a three vertex system.

Theorem 11. *The HSV of a three vertex reduced line graph with two leaders (G_{2red}) is given by (with the edge with weight a_2 deleted i.e. f_1 and f_2 are combined)*

$$\sigma_{red} = \lambda_{red} = \frac{1}{2}\frac{a_1^2 + a_3^2}{a_1 + a_3}$$

Proof. G_{2red} is pictured as:



with the corresponding reduced graph Laplacian given by:

$$L_{red} = \begin{pmatrix} a_1 + a_3 & -a_1 & -a_3 \\ -a_1 & a_1 & 0 \\ -a_3 & 0 & a_3 \end{pmatrix}. \tag{79}$$

Now compute the eigenvalue of W_{red}; where W_{red} is the controllability Gramian corresponding to the reduced system. Which is the solution of the Lyapunov equation $A_{red}W_{red} + W_{red}A_{red}^T = -B_{red}B_{red}^T$, where $A_{red} = -a_1 - a_3$ and $B_{red} = \begin{pmatrix} a_1 & a_3 \end{pmatrix}$; A_{red} and B_{red} are found by computing the reduced Laplacian L_{12}. Plugging A_{red} and B_{red} in the Lyapunov equation we obtain

$$-2(a_1 + a_3)W_{red} = -(a_1^2 + a_3^2)$$

. Hence $W_{red} \in \mathbb{R}$ and is equal to its own eigenvalue (and singular value). □

For the full order system (four vertices 'line graph' G_2) it is possible to analytically compute the controllability Gramian. However computing analytically the HSV results in long formulas and studying them in there analytical form does not seem profitable. However, if we take $a_1 = a_3 = 1$ and let a_2 variable, we can express the two Hankel singular values of the full order system in a_2. We get:

$$\sigma_1 = \lambda_1 = \frac{1}{2} \qquad \sigma_2 = \lambda_2 = \frac{1}{2(1 + 2a_2)} \tag{80}$$

Now I keep two weights equal to 1 and send the remaining one to infinity. Then I compare the Hankel singular values of the full order system with the one of the reduced system; recall that the Hankel singular values of the system or reduced system are the same as the eigenvalues of W or W_{red} that is why I denote λ_1, λ_2 and λ_{red} for the Hankel singular values of the full order system and the reduced system respectively.

$$a_1 \to \infty \qquad \lambda_1 \to \frac{1}{2}, \qquad \lambda_2 \to \infty \qquad \lambda_{red} \to \infty \tag{81}$$

$$a_2 \to \infty \qquad \lambda_1 \to \frac{1}{2}, \qquad \lambda_2 = \frac{1}{2(1 + 2a_2)} \to 0 \qquad \lambda_{red} = \frac{1}{2} \tag{82}$$

$$a_3 \to \infty \qquad \lambda_1 \to \frac{1}{2}, \qquad \lambda_2 \to \infty \qquad \lambda_{red} \to \infty \tag{83}$$

We conclude this section with the following theorem:

Theorem 12. *For the removed edge wheight a_2 large enough the largest Hankel singular value of the reduced system is close to the largest Hankel singular value of the full order system. Moreover when $a_2 \to \infty$ it even converges to the Hankel singular value of the reduced order system. And the Hankel singular values of the reduced system does not depend on the weight(s) of the deleted edge(s).*

Proof. See the observations above this theorem. □

7 Kron-reduction

In this chapter Kron-reduction theory is treated. This reduction technique is based on the Schur complement. So we start with the Schur complement. Then we introduce the Kron-reduction method and analyze how well it works.

7.1 Schur complement

The *Schur complement* [3]. Let A be an $n \times n$ matrix partitioned as

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

where A_{11} and A_{22} are square matrices. If A_{11} is nonsingular then the Schur complement of A_{11} in A is defined to be the matrix $A_{22} - A_{21}A_{11}^{-1}A_{12}$. Similarly, if A_{22} is nonsingular then the Schur complement of A_{22} in A is $A_{11} - A_{12}A_{22}^{-1}A_{21}$. We can make the following identity, which can be verified by just working out the matrix product:

$$\begin{pmatrix} I & 0 \\ -A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} I & -A_{11}^{-1}A_{12} \\ 0 & I \end{pmatrix} = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{pmatrix}. \tag{84}$$

From this identity we can easily prove that:

Corollary 6. ***Schur complement formula for the determinant***

$$det(A) = (detA_{11})det(A_{22} - A_{21}A_{11}^{-1}A_{12}) \tag{85}$$

holds. We will refer to this as the Schur complement formula for the determinant. We can also see

Corollary 7. ***Schur non singularity.***
If A and A_{11} are nonsingular then also $A_{22} - A_{21}A_{11}^{-1}A_{12}$ must be nonsingular. Similarly, if A and A_{22} are nonsingular then also $A_{11} - A_{12}A_{22}^{-1}A_{21}$ must be nonsingular.

We are going to use the Schur complement in systems of the form:

$$\begin{pmatrix} \dot{x}^{[1]} \\ \dot{x}^{[2]} \end{pmatrix} = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{12} & A_{22} \end{array}\right) \begin{pmatrix} x^{[1]} \\ x^{[2]} \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} u. \tag{86}$$

Now if we, when we want to reduce the model, take $\dot{x}^{[2]} = 0$ we get for $x^{[2]} = -A_{22}^{-1}(A_{21}x^{[1]} + B_2u)$. If we plug this expression for $x^{[2]}$ into the differential equation for $\dot{x}^{[1]}$ we get:

$$\dot{x}^{[1]} = (A_{11} - A_{12}A_{22}^{-1}A_{21})x^{[1]} + (B_1 - A_{12}A_{22}^{-1}B_2)u. \tag{87}$$

Note that the matrix coefficient standing in front of $x^{[1]}$ is just the Schur complement of A_{22} in A. And so we can use the properties of the Schur Complement. By the Schur Complement we have [13]:

Lemma 3. *Suppose that a symmetric matrix is partitioned as*

$$A = \begin{pmatrix} A_1 & A_2 \\ A_2^T & A_3 \end{pmatrix}$$

where A_1 and A_3 are square. A is positive definite if and only if both A_1 and $A_3 - A_2^T A_1^{-1} A_2$ are positive definite.

The following theorem states that taking the Schur complement of a Laplacian results in again a Laplacian matrix [17] and [25]:

Theorem 13. *If a graph is connected, all diagonal elements of L are positive . Furthermore, all Schur complements of L are well defined and are symmetric, positive semidefinite, with diagonal elements > 0, off-diagonal elements ≤ 0, and with zero row and column sums, i.e. all Schur complements of L are again Laplacian matrices.*

7.2 Kron-reduction method

Kron-reduction of the Laplacian matrix of a graph gives us again a Laplacian of a reduced graph [5]. This form of reduction is obtained by applying the Schur complement of the original Laplacian matrix with respect to a subset of vertices. It is used in classic circuit theory and in other related disciplines such as electrical impedance tomography, smart grid monitoring, transient stability assessment in power networks and analysis and simulation of induction motors and power electronics. We define the Laplacian matrix L as above. Now we will apply the Schur complement on the Laplacian matrix L. We will see, in general and by examples, that the reduced lower dimensional matrix L_{red} is again a well defined Laplacian matrix and therefore a graph can be associated to it. For our purpose to see if this is a good reduction technique we are going to take a look at the eigenvalues of L and L_{red} and check if they are related. Because it is the standard in the literature that the leader vertices are in the top left block of the matrix and the follower vertices in the bottom right block we adapt this notation in this chapter for convenience.
Now take $\mathcal{I}_n$ as the index set of all vertices, $\beta = \{n_l + 1, \ldots, n\}$ as the indices of the follower vertices and $\alpha = \{1, \ldots, n_l\} = \mathcal{I}_{n_l}$ the indices of the vertices corresponding to the leader vertices. Now consider again the consensus algorithm:

$$\begin{pmatrix} \dot{x}_\alpha \\ \dot{x}_\beta \end{pmatrix} = -\left(\begin{array}{c|c} L_l & L_{lf} \\ L_{fl} & L_f \end{array}\right)\begin{pmatrix} x_l \\ x_f \end{pmatrix} = -\left(\begin{array}{c|c} L_{\alpha\alpha} & L_{\alpha\beta} \\ L_{\beta\alpha} & L_{\beta\beta} \end{array}\right)\begin{pmatrix} x_\alpha \\ x_\beta \end{pmatrix} = \left(\begin{array}{c|c} D & C \\ B & A \end{array}\right)\begin{pmatrix} x_l \\ x_f \end{pmatrix} \tag{88}$$

Now by using Gaussian elimination of the vertices corresponding to state vectors in x_β (the followers) and putting $\dot{x}_\beta = 0$ we get a reduced network with $|\alpha|$ vertices. This reduced matrix L_{red} is given by applying the Schur complement with respect to vertices β, that is $L_{red} = \text{Ł}_{\alpha\alpha} - L_{\alpha\beta}L_{\beta\beta}^{-1}L_{\beta\alpha}$ and thus $L_{red} \in \mathbb{R}^{|\alpha|\times|\alpha|}$. The accompanying matrix is given by: $L_{ac} = -L_{\alpha\beta}L_{\beta\beta}^{-1} \in \mathbb{R}^{|\alpha|\times(n-|\alpha|)}$. This accompanying matrix maps the vertices corresponding to $\dot{x}_\beta$ to the vertices corresponding to $\dot{x}_\alpha$.

Example. Consider a four point star-like graph (see figure 1) with all weights one, three leaders (corresponding to the vertex set α) and one follower (corresponding to β) which we want to reduce with one follower vertex by applying Kron-reduction. The corresponding Laplacian is given by

$$L = \left(\begin{array}{c|ccc} 3 & -1 & -1 & -1 \\ \hline -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{array}\right) = \left(\begin{array}{c|c} L_{\beta\beta} & L_{\beta\alpha} \\ \hline L_{\alpha\beta} & L_{\alpha\alpha} \end{array}\right)$$

$$\Rightarrow L_{red} = \text{Ł}_{\alpha\alpha} - L_{\alpha\beta}L_{\beta\beta}^{-1}L_{\beta\alpha} = \begin{pmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{pmatrix}.$$

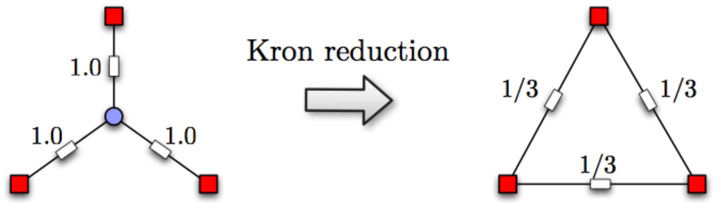


Figure 1: Kron-reduction of a star-like graph

Note that this example is denoted in the way I defined the structure of the Laplacian in the previous chapters. In the following more abstract theory, to agree with the theory about Kron-reduction the leaders will be arranged in the top left corner of the matrix and the followers in the bottom right corner, as said before. It is easily verified that L_{red} in the previously example is again a Laplacian matrix and hence it has a corresponding graph. In the previous example it is clear that the first vertex is deleted and

that the other three vertices remain. Now because the first vertex was connected to all other vertices, the new graph has connections between every vertex. Moreover, the weight on every edge is the weight 1 of the old connection to the deleted vertex, divided by the total weight (3) of the deleted vertex.
Now we try to make more clear what is happening with Kron-reduction in a more general graph. Therefore we take a complete graph and reduce this graph with one vertex (the first follower vertex) in the above manner.

$$L = \left(\begin{array}{ccc|c} a_1+a_2+a_3 & -a_1 & -a_2 & -a_3 \\ -a_1 & a_1+a_4+a_5 & -a_4 & -a_5 \\ -a_2 & -a_4 & a_2+a_4+a_6 & -a_6 \\ \hline -a_3 & -a_5 & -a_6 & a_3+a_5+a_6 \end{array}\right). \tag{89}$$

We obtain for the reduced system a three vertex graph with the following Laplacian:

$$L_{red} = \begin{pmatrix} \frac{a_1(a_2+a_3)}{a_1+a_2+a_3}+a_4+a_5 & -\frac{a_1a_2}{a_1+a_2+a_3}-a_4 & -\frac{a_1a_3}{a_1+a_2+a_3}-a_5 \\ -\frac{a_1a_2}{a_1+a_2+a_3}-a_4 & \frac{a_2(a_1+a_3)}{a_1+a_2+a_3}+a_4+a_6 & -\frac{a_2a_3}{a_1+a_2+a_3}-a_6 \\ -\frac{a_1a_3}{a_1+a_2+a_3}-a_5 & -\frac{a_2a_3}{a_1+a_2+a_3}-a_6 & \frac{a_3(a_1+a_2)}{a_1+a_2+a_3}+a_5+a_6 \end{pmatrix}. \tag{90}$$

This reduced Laplacian has a clear structure. The first thing that can be noticed is that vertex f_1 has been deleted together with all the corresponding edges but all the corresponding edge weights are maintained in other edges of the reduced Laplacian.
In order to construct the theory about Kron-reduction we need some notation first.
Just like in the rest of this thesis we consider an undirected, connected and weighted graph without self-loops with n vertices and adjacency matrix $A \in \mathbb{R}^{n\times n}$ and Laplacian matrix L.
Now if we take a Schur complement with respect to a subset of vertices, it turns out that the lower dimensional matrix L_{red} is again a well-defined Laplacian matrix. This is called Kron-reduction. With $n \in \mathbb{N}$ we have the index set $\mathcal{I}_n = \{1, 2, \ldots, n\}$. We have already introduced the two index sets α corresponding to the leader vertices (which we want to keep) and β corresponding to the follower vertices in such a way that $\alpha, \beta \subset \mathcal{I}_n$. Let $L[\alpha, \beta]$ denote the submatrix of L by taking the rows with index α and columns indexed by β. Furthermore we have the shorthand notation $L[\alpha, \beta) = L[\alpha, \mathcal{I}_n\backslash\beta]$, $L(\alpha, \beta] = L[\mathcal{I}_n\backslash\alpha, \beta]$ and $L(\alpha, \beta) = L[\mathcal{I}_n\backslash\alpha, \mathcal{I}_n\backslash\beta]$. For $i, j \in \mathcal{I}_n$ we have $L[i, j] = L[i, j] = L_{i,j}$.
Now we define the Schur complement again, in new notation, which has the major advantage that we can choose the rows and columns we want. We want to reduce (some of) the follower vertices and want to keep the leaders. So we take the Schur complement with respect to $L(\alpha, \alpha) = L[\beta, \beta]$. If $L(\alpha, \alpha)$ is not singular then the $|\alpha| \times |\alpha|$ dimensional matrix

$$L_{red} := L/L(\alpha, \alpha) = L[\alpha, \alpha] - L[\alpha, \alpha)L(\alpha, \alpha)^{-1}L(\alpha, \alpha] \tag{91}$$

is the Kron-reduced matrix of L.
Since L is symmetric, the eigenvalues are real and can be arranged in increasing order:

$$0 = \lambda_1(L) \leq \lambda_2(L) \leq \ldots \leq \lambda_n(L)$$

We have that $A, L, \lambda_2(L) > 0$ if and only if G is connected.

Now we can formulate the following Lemma [5]:

Lemma 4. ***Structural Properties of Kron-reduction***. *Let $L \in \mathbb{R}^{n\times n}$ be a symmetric irreducible Laplacian matrix and let α be a proper subset of $\mathcal{I}_n$ with $|\alpha| \geq 2$. Then the following statements hold:*

- ***Existence:*** *The Kron-reduced matrix $L_{red} = L/L(\alpha, \alpha)$ is well defined.*
- ***Closure:*** *L_{red} is also a Laplacian matrix.*

Proof.

- We have to prove that $L(\alpha,\alpha)$ is invertible. This is true because L is weakly diagonal dominant since $L_{ii} = \sum_{j=1,j\neq i}^{n} |L_{ij}|$ for all $i \in \mathcal{I}_n$. Now due to the irreducibility of L we have that $L_{ii} > \sum_{j=1,j\neq i,j\notin\alpha}^{n} |L_{ij}|$ holds for at least one $i \in \mathcal{I}_n \backslash \alpha$. Therefore $L(\alpha,\alpha)$ is also irreducible, weakly diagonal dominant and has at least one row with strictly positive row sum. Hence $L(\alpha,\alpha)$ is invertible.

- This follows from the properties of L and the closure properties of the Schur complement. Now because L is a symmetric M-matrix we can conclude that the Kron-reduced matrix L_{red} is also a symmetric M-matrix. Now assume, without loss of generality, that $\alpha = \mathcal{I}_{|\alpha|}$ (the indices of the leader vertices) for the Laplacian holds:

$$\left(\begin{array}{c|c} L[\alpha,\alpha] & L[\alpha,\alpha) \\ \hline L(\alpha,\alpha] & L(\alpha,\alpha) \end{array}\right) \left(\begin{array}{c} \mathbf{1}_{|\alpha|} \\ \hline \mathbf{1}_{n-|\alpha|} \end{array}\right) = \left(\begin{array}{c} \mathbf{0}_{|\alpha|} \\ \hline \mathbf{0}_{n-|\alpha|} \end{array}\right). \tag{92}$$

  Taking the Schur complement (elimination of the second block of equations) we have $\mathbf{0}_{|\alpha|} = L_{red}\mathbf{1}_{|\alpha|}$, which shows that also the reduced matrix is again a Laplacian matrix.

□

By this lemma we now know that if we reduce our matrix L with Kron-reduction we end up with a reduced matrix L_{red} which is still a Laplacian matrix with the same properties. We will see that a positive load $L_{ii} > 0$ in the full order model lowers the values in $L_{red}[i,j]$ in the reduced network and is increasing the reduced loads $L_{red}[i,i]$ [5]. This is clear if we take a look at (90). If we increase the value of $a_1 + a_2 + a_3$ then we see that the value of $L_{red}[i,i]$ increases because the sum of products $a_1a_2 + a_1a_3$ increases faster, this holds for every row. From (90) it is clear that the off diagonal elements in row i decrease.

From now on we assume throughout, without loss of generality, that $\mathcal{I}_{|\alpha|} = \alpha$. Instead of doing the reduction to all follower vertices at once, it is much more intuitive to do it one by one. By reducing a graph with one vertex at a time we get much more insight in what is happening to a graph by applying Kron-reduction. And so we get to the definition of iterative Kron-reduction [5]:

Definition 6. ***Iterative Kron-reduction.***
Iterative Kron-reduction associates to a symmetric irreducible Laplacian matrix L and indices $\{1,\ldots,|\alpha|\}$ a sequence of matrices $L \in \mathbb{R}^{(n-l)\times(n-l)}, l \in \{1,\ldots,n-|\alpha|\}$ defined by the following Schur complement

$$L^l = L^{l-1}/L^{l-1}_{k_lk_l} \tag{93}$$

where $L^0 = L$ and $k_l = n+1-l$ is the lowest diagonal entry of the follower vertices in L^{l-1}.

By Lemma 4 we now know that for all l in this sequence L^l is a Laplacian inducing a graph G. From this iterative definition we can observe a couple of things for the graph induced by L^l:

- Connectivity is maintained, this is because
- All connections are maintained between vertices which are not deleted and
- The edges $\{i,y\}$ disappear whenever an edge was connected to a vertex y which is deleted.
- A new edge $\{i,j\}$ is created if and only if k_l is connected to both vertex i and j before the reduction.

From the above four observations and by computation we can see that the iterative Kron-reduction can be written in components in the following way:

$$L^l_{ij} = L^{l-1}_{ij} - \frac{L^{l-1}_{ik_l}L^{l-1}_{jk_l}}{L^{l-1}_{k_lk_l}} \tag{94}$$

for all vertices $i,j \in \{1,\ldots,n-l\}$. Now take a good look at the component formula (94). Assume that i or j is not connected via an edge to k_l we get in formula (94) $L^l_{ij} = L^{l-1}_{ij} - 0 = L^{l-1}_{ij}$, in words: the edge $\{i,j\}$ has the same weight as before the reduction. Now assume both i and j are connected to k_l with an edge with weight $a > 0$ and $b > 0$ respectively. In the reduced Laplacian we obtain: $L^l_{ij} = L^{l-1}_{ij} - \frac{ab}{L^{l-1}_{k_l k_l}}$. So if there was no edge connecting vertex i with j in L^{l-1} then there is an edge in the graph induced by L^l with weight $-\frac{ab}{L^{l-1}_{k_l k_l}} < 0$. If there was already a connection between i and j in L^{l-1} then the weight along this edge in L^l is increased by the product of the weights a and b divided by the total weight sum of the edges connected to k_l. All of this we can summarize in a lemma [5]:

Lemma 5. ***Properties of iterative Kron-reduction.***
Consider the sequence $\{L^l\}_{l=1}^{n-|\alpha|}$ defined via the iterative Kron-reduction then the following holds:

- ***Well-posedness:*** *L^l is well defined for $l \in \{1, \ldots, n - |\alpha|\}$.*
- ***Quotient property:*** *The Kron-reduced matrix $L_{red} = L/L(\alpha, \alpha)$ can be obtained by iterative reduction of all vertices $k_l \in \mathcal{I}_n \backslash \alpha = \beta$ that is:*

$$L_{red} = L^{n-|\alpha|} = L^{|\beta|} \tag{95}$$

- ***Row sum:*** *$\sum_{j=1}^{n-l} L^l_{ij} = A^l_{ii} = A^{l-1}_{ii} = 0$ for $i \in \{1, \ldots, n - l\}$.*

Proof. Statement 2 is simply proved by the Quotient Formula stating that the Schur complements can be taken iteratively or in a single step. Furthermore, the Quotient Formula states that all intermediate Schur complements L^l exist for $l \in \{1, \ldots, n - |\alpha|\}$. This fact together with closure properties in Lemma 4 proves that matrices L^l are well-defined. Now to prove the last statement, first note that $A^0 = A$ and $L^0 = L$, for notational comfort we take $l = 1$ and $k_1 = n$ then the ith row sum is given by:

$$\sum_{j=1}^{n-1} L^1_{ij} = \sum_{j=1}^{n-1} (L_{ij} - \frac{L_{in} L_{jn}}{L_{nn}}) = \sum_{j=1}^{n-1} (L_{ij} - \frac{A_{in} A_{jn}}{L_{nn}}) = \sum_{j=1}^{n-1} (L_{ij}) - \frac{A_{in}}{L_{nn}} \sum_{j=1}^{n-1} (A_{jn}) = A_{in} - A_{in} = 0. \tag{96}$$

□

In the proof we used the component formula (94) and the identities $\sum_{j=1}^{n-1} L_{ij} = A_{in}$ and $\sum_{j=1}^{n-1} A_{jn} = L_{nn}$.

We are now at the point that we can analyze and characterize the properties of Kron-reduction. We take a look at how the L_{red} and the corresponding graph G_{red} are obtained from L and G. This is formulated in the next theorem [5]:

Theorem 14. ***Topological properties of Kron-reduction.***
Let G, G_{red} be undirected weighted graphs associated to L, $L_{red} = L/L(\alpha, \alpha)$. The following statements hold:

- ***Edges:*** *Two vertices $i, j \in \alpha$ are connected by an edge in G_{red} if and only if there exists a path from i to j in G whose vertices belong to $\{i, j\} \cup |\beta|$*
- ***Reduction of connected components:*** *If the vertices $\beta \subseteq \mathcal{I} \backslash \alpha$ form a connected subgraph of G, then the vertices $\hat{\alpha} \subseteq \alpha$ adjacent to β in G form a clique (a subset of its vertices such that every two vertices in the subset are connected by an edge) in G_{red}.*

The proof will consist out of observations we already have made.

Proof. We want to reduce a single follower vertex k with the iterative one step component Kron-reduction formula (94). Due to the closure of Laplacians under iterative Kron-reduction, lemma 4, we know that every strictly negative element L_{ij} is rendered to a strictly negative element L^1_{ij} in the reduced graph; this follows directly from formula (94) where the first term on the right hand side is negative and the second nonpositive. Therefore all edges, as we have already seen, in G are also maintained in the graph induced by L^1_{ij}. By the same iterative Kron-reduction a 0 at L_{ij} is converted to a strictly negative element in L^1_{ij} if and only if both vertices i and j are neighbors of vertex k. As a consequence of this, reduction of vertex k leads to a complete graph among all vertices that were adjacent to this vertex k. Hence by single step Kron-reduction we obtain a loop-less well defined Laplacian matrix L^1_{ij}. From the second statement in lemma 5 we know that the one-step reduction of all follower vertices is equivalent to the iterative reduction of all follower vertices. And so we can repeat the arguments of the one-step reduction of a single vertex and therefore statement 1 holds. Now if we apply statement 1 to all vertices in β we see that we end up with vertices $\hat{\alpha}$, were there is a connection between every two vertices in this set $\hat{\alpha}$, i.e. a clique. □

As a result of theorem 14 the connectivity between the remaining vertices after reduction can only increase. This means that the algebraic connectivity $\lambda_2(L)$ should increase by the reduction. This is formulated in the next theorem [5]:

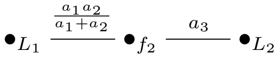
Theorem 15. ***Spectral Properties of Kron-reduction.***
For the spectrum of the Kron-reduced matrix $L_{red} = L/L(\alpha, \alpha)$ we have that for any $r \in \mathcal{I}_{|\alpha|}$

$$\lambda_r(L) \leq \lambda_r(L_{red}) \leq \lambda_r(L[\alpha,\alpha]) \leq \lambda_{r+n-|\alpha|}(L) \tag{97}$$

For proof see [5].

7.3 Kron-reduction Hankel singular value error analysis

We consider again the line graph G_2 with two leaders and four vertices in total (G_2 is pictured in the previous chapter, also the Laplacian is given in that chapter). Now we use Kron-reduction with respect to f_1 we obtain the graph G_{2red}:



with corresponding Laplacian matrix given by:

$$L_{red} = \begin{pmatrix} \frac{a_1a_2}{a_1+a_2} + a_3 & -\frac{a_1a_2}{a_1+a_2} & -a_3 \\ -\frac{a_1a_2}{a_1+a_2} & \frac{a_1a_2}{a_1+a_2} & 0 \\ -a_3 & 0 & a_3 \end{pmatrix} \tag{98}$$

All weight values of the edges are kept in the new (formed) edges of the Kron-reduced matrix G_{2red}, so it is complicated to analyze this error. Therefore I will take a look at the influence of all weights, starting with a_1. By sending a_1 to infinity the Hankel singular value of the reduced system is converging to the value 0.5, if we keep the weights $a_1 = a_3 = 1$. The smallest HSV of the full order system tends to zero but the highest singular value is growing linear and tends to infinity if $a_1 \to \infty$.

Now send a_2 to infinity and keep the other two weights $a_1 = a_3 = 1$. In the full order system one HSV goes to 0.5 and the other goes to 0. In the reduced order system the HSV goes to 0.5. So when $a_2 \to \infty$ the reduced system seems to be a good reduction, i.e. the error between the highest singular values goes to zero.

In the case of $a_3 \to \infty$ the lowest HSV of the full order system goes to zero. The largest tends to infinity but it seems that it stays close to the HSV of the reduced order model. The difference between them is converging to 0.25. When a_3 is getting really large the error is oscillating around 0.25, whether this is because of accuracy of the numerical computation or the differences really fluctuated when a_3 is going to infinity is unknown to me.

Presented in a more structured way we obtain for the full order system (recall: we send one of the three weights to infinity and the other two weights we keep both equal to 1):

$$a_1 \to \infty \qquad \lambda_1 \to \frac{1}{2}, \;\; \lambda_2 \to \infty \tag{99}$$

$$a_2 \to \infty \qquad \lambda_1 \to 0, \;\; \lambda_2 \to \frac{a_1^2 + a_3^2}{2(a_1 + a_3)} = \frac{1}{2} \tag{100}$$

$$a_3 \to \infty \qquad \lambda_1 \to \frac{1}{2}, \;\; \lambda_2 \to \infty. \tag{101}$$

And for the Kron-reduced system (reduced by vertex f_1) we have that the controllability Gramian of this reduced graph is $W_{KronRed}$ is a scalar, its value is equal to the eigenvalue and is given by:

$$W_{KronRed} = \frac{a_1^2 a_2^2 + a_1^2 a_3^2 + a_2^2 a_3^2 + 2a_1 a_2 a_3^2}{2(a_1 + a_2)(a_1 a_2 + a_1 a_3 + a_2 a_3)} \tag{102}$$

and hence for the reduced system we have:

$$a_1 \to \infty \qquad \lambda_{red} \to \frac{a_2^2 + a_3^2}{2(a_2 + a_3)} = \frac{1}{2} \tag{103}$$

$$a_2 \to \infty \qquad \lambda_{red} \to \frac{a_1^2 + a_3^2}{2(a_1 + a_3)} = \frac{1}{2} \tag{104}$$

$$a_3 \to \infty \qquad \lambda_{red} = \frac{1 + 4a_3^2}{4(1 + 2a_3)} \to \infty. \tag{105}$$

I conclude with two theorems:

Theorem 16. *For a_2 large enough the largest Hankel singular values of the reduced system are close to the largest Hankel singular values of the full order system.*

Proof. See observations above. □

Theorem 17. *The steady-state response of the reduced system is for every value of a_1, a_2, a_3 equal to the steady-state response of the full order system and is of course a Laplacian matrix times the input vector.*

Proof. I will treat this theorem in full detail in the next section about stationary state response. □

8 Stationary state output

The stationary response or static input/output gain is the response to a constant input. In this case we look at the response of a constant input when time is going to infinity. This is a way of saying something about how good the reduced system is comparing to the full order system. My definition of stationary response is based on the definition in the book [19].

To derive directly the stationary state output for $t \to \infty$ when we apply a constant input $\bar{u}(t) = c$ is given. The general output is given by:

$$y(t) = Ce^{At}x_0 + \int_0^t Ce^{A(t-s)}B\bar{u}(t)ds + D\bar{u}(t). \tag{106}$$

Because we assume the matrix A is Hurwitz we have that for $t \to \infty \;\Rightarrow\; Ce^{At}x_0 \to 0$. If we get the constant vectors $C, e^{At}, B, \bar{u}$ out of the integral and then integrate the remaining integral and let $t \to \infty$ we obtain:

$$\begin{aligned}
y(t) &= \int_0^t Ce^{A(t-s)}B\bar{u}(t)ds + D\bar{u}(t) = Ce^{At}\int_0^t e^{-As}dsB\bar{u}(t) + D\bar{u}(t)\\
&= Ce^{At}[e^{-As}]_0^t - A^{-1}B\bar{u}(t) + D\bar{u}(t)\\
&= Ce^{At}[e^{-At} - I] - A^{-1}B\bar{u}(t) + D\bar{u}(t)\\
&= -C[I - e^{At}]A^{-1}B\bar{u}(t) + D\bar{u}(t)\\
t \to \infty &\Rightarrow e^{At} \to 0\\
\Rightarrow y(t) &= (-CA^{-1}B + D)\bar{u}(t)
\end{aligned}$$

Loosely speaking the stationary state response with constant input $\bar{u}(t)$ is obtained by assuming that the state converges to a certain value when $t \to \infty$. Therefore $\dot{x}(t) \to 0$. Our state space equations for $t \to \infty$ then become:

$$\begin{cases} 0 = Ax(t) + Bu(t) \\ y(t) = Cx(t) + D\bar{u}(t) \end{cases} \Rightarrow \begin{cases} x(t) & = & -A^{-1}B\bar{u}(t) \\ y(t) & = & (-CA^{-1}B + D)\bar{u}(t) \end{cases} \tag{107}$$

As said with this stationary response on a constant input function we can compare the behavior of the full order system with the behavior of the reduced system. In a further research we could also look to stationary response on harmonic oscillations and compare this for the full order and reduced system.

From now on we are going to apply a constant input $\bar{u}(t)$ to the system and take a look at the stationary response $\bar{y}(t) = (-CA^{-1}B+D)\bar{u}(t)$. More specific we are interested in the matrix $(-CA^{-1}B+D)$ which is the static input-output gain; for notational comfort I will denote this matrix by H, i.e.

$$H = (-CA^{-1}B + D). \tag{108}$$

We start by looking at single-leader systems. We first state that matrix H is -1 times a symmetric Laplacian and therefore has zero row and column sums. Then we can state a theorem which is similar to that we have seen in the consensus chapter namely: corollary 1.

Theorem 18. *The matrix H is -1 times a Laplacian matrix. In particular, $\mathbf{1}_{n_l}^T H = \mathbf{0}^T$ and $H\mathbf{1}_{n_l} = \mathbf{0}_{n_l}$.*

Proof. Consider the Laplacian matrix L partitioned as in (11), clearly $-L$ has zero row and column sums. The Schur complement of block A of the matrix $-L$ equals $D - CA^{-1}B$ which is equal to H. The Schur complement of block $-A$ of the matrix L is given by $-D + CA^{-1}B$ and is again a Laplacian matrix by properties of the Schur complement. Hence $-H$ is a Laplacian. □

This brings us to a theorem about the stationary state response of a single-leader system

Theorem 19. *For a graph, G, with one leader and $n-1$ follower agents the corresponding system has a stationary sate response of $\bar{y}(t) = 0$. Moreover all the corresponding systems of all possible (follower) reduced graphs by merging vertices or applying Kron-reduction also have stationary state response $\bar{y}(t) = 0$.*

Proof. From the previous theorem 18 we know that $\mathbf{1}_{n_l}^T H = 0$ and $H\mathbf{1}_{n_l} = 0$ holds. Now we only have one leader ($n_l = 1$); 'matrix' $D \in \mathbb{R}$, hence $H \in \mathbb{R}$, therefore $H * 1 = 0$ and so H must be zero, i.e. $H = 0$. The corresponding stationary state response is given by $\bar{y}(t) = 0\bar{u}(t) = 0$. The 'matrix' H_{red} for follower reduced systems is also in $\mathbb{R}$ and again the 'row' and 'column' sum should be zero. Therefore also for all reduced systems the stationary state response equals 0. □

In the two following subsections we analyze the static input output gain of the Kron-reduction and Merging vertices techniques. The stationary response of single-leader systems is not useful to say something about how good the reduced system is behaving. Therefore we will take a look at multiple leader systems for both the Kron-reduction and Merging vertices reduction techniques.

8.1 Kron-reduction stationary state response

We start with system consisting of two leaders and $n-2$ followers. We have $D \in \mathbb{R}^{2\times 2}, B, C^T \in \mathbb{R}^{n-2\times 2}$ and $A \in \mathbb{R}^{n-2\times n-2}$ and so $H \in \mathbb{R}^{2\times 2}$. From theorem 18 it follows that

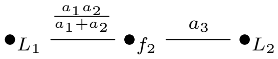
$$H = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \times J \tag{109}$$

with $J \in \mathbb{R}$. We start our search with the two leader line graph G_2, with two followers. By using Mathematica it turns out that for this graph $J = \frac{a_1 a_2 a_3}{a_1 a_2 + a_1 a_3 + a_2 a_3}$. Now apply Kron-reduction with respect to follower vertex f_1,

$$L_{red} = L/L[1,1] = L[\{2,3,4\};\{2,3,4\}] - L[\{2,3,4\},1]L^{-1}[1,1]L[1,\{2,3,4\}] \tag{110}$$

$$= \begin{pmatrix} \frac{a_1a_2}{a_1+a_2} + a_3 & -\frac{a_1a_2}{a_1+a_2} & -a_3 \\ -\frac{a_1a_2}{a_1+a_2} & \frac{a_1a_2}{a_1+a_2} & 0 \\ -a_3 & 0 & a_3 \end{pmatrix} = \begin{pmatrix} -A_r & -B_r \\ -C_r & -D_r \end{pmatrix}. \tag{111}$$

The graph G_{2red} is now given by:



Compute $H_r = D_r - C_rA^{-1}B_r$ it follows that for a reduced system holds that $J_r = J$. From this and the observation that computing H (or J) is the same as applying Kron-reduction to $-L$ with respect to all followers the presumption arises that the stationary state response of every Kron-reduced system always equals that of the full order system. We formulate this presumption in the following theorem.

Theorem 20. *The static input-output gain is invariant under Kron-reduction with respect to followers.*

Proof. Consider an auxiliary graph G and its Laplacian L, where L is partitioned again as in (11). The stationary state output gain is given by $H = D - CA^{-1}B$. Note that H is just the Laplacian of the Kron-reduced system reduced with all its followers, i.e.

$$-\overline{L_{red}} = -L/-L(\alpha,\alpha) = -L[\alpha,\alpha] - (-L[\alpha,\beta])(-L[\beta,\beta]^{-1})(-L[\beta,\alpha]) \tag{112}$$

$$= D - CA^{-1}B = H. \tag{113}$$

Note, by properties of Kron-reduction this proves again that H is just -1 times a Laplacian matrix for a Kron-reduced system reduced by one or by a part of the followers $\bar{\beta} \subset \beta$. The reduced Laplacian is given by $L_{red} = L/L[\bar{\beta},\bar{\beta}] = L[\bar{\alpha},\bar{\alpha}] - L[\bar{\alpha},\bar{\alpha})L^{-1}(\bar{\alpha},\bar{\alpha})L(\bar{\alpha},\bar{\alpha}]$, where $\bar{\alpha} = \alpha \cup (\beta/\bar{\beta})$ are all the leaders together with the follower agents which we want to preserve. The corresponding stationary state output factor $H_r = D_r - C_rA_r^{-1}B_r$. H_r corresponds again to the graph in which all (remaining) follower agents are Kron-reduced. By the iterative property of Kron-reduction we obtain: $H_r = -\overline{L_{red}} = H$. Denoted in a scheme:

$$-L \to -L^1 \to \ldots \to -L^{n_f} = -\overline{L_{red}} = H \tag{114}$$

where the arrows stand for Kron-reduction. Hence the stationary state response corresponds with the Kron-reduction of all (remaining $n_f - i$) followers of $-L^i$, for $i \in \{1,\ldots,n_f\}$ and so all Kron-reduced systems have the same stationary state response as the full order system. □

From the proof above some corollaries immediately follow:

Corollary 8. $H = D - CA^{-1}B = D_r - C_rA_r^{-1}B_r = H_r$ *for a Kron-reduced system.*

Corollary 9. *The static input-output gain H can be expressed in terms of Kron-reduction as follows:*

$$H = -\overline{L_{red}} = -L/L(\alpha,\alpha) \tag{115}$$

where α is the index set of all leaders and $\beta = I/\alpha$ the index set of all followers. So the static input-output gain H can be computed by taking to Schur complement of the block consisting of all follower agents of the matrix $-L$.

8.2 Merging vertices stationary state response

We start with the line graph with four vertices of which two leaders G_2. From the previous section, Kron-reduction stationary state response, we know that computing the stationary state response is just taking the Schur complement of the Laplacian with respect to all the followers. Recall that for this graph G_2 the stationary state response is given by (109), where $J = \frac{a_1a_2a_3}{a_1a_2+a_1a_3+a_2a_3}$. Now we merge vertices f_1 with f_2 by the transformation matrix T_{12}, our reduced Laplacian is:

$$L_{12} = \begin{pmatrix} a_1 + a_3 & -a_1 & -a_3 \\ -a_1 & a_1 & 0 \\ -a_3 & 0 & a_3 \end{pmatrix} \tag{116}$$

the corresponding static input-output gain of the reduced system is given by

$$H_r = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \frac{a_1 a_3}{a_1 + a_3}.$$

Where $J_r = \frac{a_1 a_3}{a_1 + a_3}$. The difference between J and J_r will reduce to zero whenever $a_2 \to \infty$, i.e.

$$\lim_{a_2 \to \infty} J = J_r. \tag{117}$$

In words: whenever the deleted weight (the weight on the edge between the two merging vertices) is becoming arbitrary high, the stationary state response of the system is arbitrary close to that of the reduced system. This observation leads us to the following conjecture:

Conjecture 1. *The stationary state response of the full order system is equal to the stationary state response of the 'merging vertices reduced' system if the weights on the edges between all the merging vertices is infinite.*

9 Conclusion

The tested balancing methods did not preserve the graph structure. We have found two structure-preserving model reduction methods.

- Merging vertices
- Kron-reduction

With the Merging Vertices method we reduce the system by merging some vertices. The Kron-reduction uses the Schur complement to reduce the number of vertices in the system. To analyze how well both methods are working we looked at the static input-output gain and the Hankel singular values of both the full order as the reduced system. Analyzing the static input-output gain we observed that the Kron-reduction is performing well, the static input-output is even invariant under Kron-reduction with respect to the followers. The research on the static state input-output response of the merging vertices has led to the conjecture that the stationary state response of the full order system is equal to the stationary state response of the reduced system if the weights on the edges between the vertices which are merging is going to infinity. Hence, if the weights of the deleted edges are large enough also the merging vertices reduced system gives a good reduced model in terms of the static state input-output gain. The Hankel singular values of the system turned out to be equal to the eigenvalues of the corresponding control Gramian W. Based on the analyses of the HSV we cannot draw a conclusion which method is better. In both methods the highest HSV of the full order system is converging to the HSV of the reduced order system whenever $a_2 \to \infty$. Or more general, if *conjecture* 1 is true, the weight of the deleted edges should go to infinity in order to get a good reduced system. For further research studying more graphs is important to proof that *conjecture* 1 is true.

References

[1] A.C. Antoulas and D.C. Sorensen. Approximation of large-scale dynamical systems: An overview, 2001.

[2] R.B. Bapat. *Linear Algebra and Linear Models.* Universitext (1979). Springer, 2000.

[3] R.B. Bapat. *Graphs and Matrices.* Springer, Hindustan book agency, 2011.

[4] B. Bollobás. *Modern Graph Theory.* Graduate Texts in Mathematics. Springer, 1998.

[5] F. Dorfler and F. Bullo. Kron Reduction of Graphs with Applications to Electrical Networks. *ArXiv e-prints*, feb 2011.

[6] G.E. Dullerud and F.G. Paganini. *A Course in Robust Control Theory: A Convex Approach.* Texts in Applied Mathematics. Springer, 2000.

[7] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, april 1972.

[8] S.H. Friedberg, A.J. Insel, and L.E. Spence. *Linear algebra.* Pearson Education, 2003.

[9] C Godsil and G Royle. *Algebraic Graph Theory.* Springer-Verlag, 2001.

[10] J.A. Guerrero. Leader-based multi-agent consensus: Robust control design and stability analysis.

[11] T. Gustavi, D.V. Dimarogonas, M. Egerstedt, and Xiaoming Hu. On the number of leaders needed to ensure network connectivity in arbitrary dimensions. In *Control and Automation, 2009. MED '09. 17th Mediterranean Conference on*, pages 98 –103, june 2009.

[12] Tero Harju. Lecture notes on graph theory, 2011.

[13] Jiangping Hu, Yiguang Hong, and Linxin Gao. Tracking control for multi-agent consensus with an active leader and variable topology. 2007.

[14] D. Jetsema, J.M.A. Scherpen, and A.J. van der Schaft. Lecture notes for: Nonlinear systems theory (part 1,2 and 3), February 2011.

[15] Meng Ji, A. Muhammad, and M. Egerstedt. Leader-based multi-agent coordination: controllability and optimal control. In *American Control Conference, 2006*, June 2006.

[16] R. Lozano, M.W. Spong, J.A. Guerrero, and N. Chopra. Controllability and observability of leader-based multi-agent systems. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 3713 –3718, dec. 2008.

[17] Nynke Niezink. Consensus in networked multi-agent systems. Master thesis applied mathematics, Rijksuniversiteit Groningen, 2011.

[18] R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, jan. 2007.

[19] G.J. Olsder and J.W. van der Woude. *Mathematical Systems Theory.* Delft University Press, 1998.

[20] P.J. Patil and M.D. Patil. Model order reduction of high order lti system using balanced truncation approximation. In *Process Automation, Control and Computing (PACC), 2011 International Conference on*, pages 1 –6, july 2011.

[21] J. Phillips, L. Daniel, and L. Miguel Silveira. Guaranteed passive balancing transformations for model order reduction. In *Design Automation Conference, 2002. Proceedings. 39th*, pages 52 – 57, 2002.

[22] Rostyslav V. Polyuga. *Model Reduction of Port-Hamiltonian Systems.* Phd thesis, Rijksuniversisteit Groningen, april 2010.

[23] Amirreza Rahmani, Meng Ji, Mehran Mesbahi, and Magnus Egerstedt. Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM J. Control Optim.*, 48:162–186, February 2009.

[24] Maryam Saadvandi. Passivity preserving model reduction and selection of spectral zeros. Master of science thesis, Stockholm, 2008.

[25] Arjan van der Schaft. Characterization and partial synthesis of the behavior of resistive circuits at their terminals. pages 423–428, 2010.

[26] J.M.A. Scherpen and A.J. van der Schaft. Balanced model reduction of gradient systems. pages 1–6, October 2011.

[27] A. van der Schaft. Balancing of lossless and passive systems. *IEEE Transactions on Automatic Control*, 53(9):2153 –2157, oct. 2008.

[28] Wikipedia. Gram-schmidt process, 2011.

[29] Wikipedia. Graph theory, 2011.

[30] Wikipedia. Hankel singular value, 2011.

[31] Wikipedia. Kronecker product, 2011.

[32] Wikipedia. Laplacian matrix, 2011.