university of groningen

faculty of science and engineering

mathematics and applied mathematics

Model Reduction Error Analysis of Multi-Agent Systems

Bachelor's Project Mathematics

July 2017

Student: M. Ruiter

First supervisor: Prof.dr. M.K. Camlibel

Second assessor: Dr. M.A. Grzegorczyk

Abstract

This thesis gives a numerical method for computing the model reduction error of a multi-agent system associated with an undirected graph, given a leader set and an arbitrary partition. One can determine this model reduction error easily, if one makes use of a so called almost-equitable partition, but for other instances this (most likely) cannot be done analytically. This model reduction error is expressed in terms of the $\mathcal{H}_2$-norm, for which the Lyapunov equation had to be solved. This can only be done when the system is stable, so we make use of similarity of systems. We have experimented with different graph partition and may conclude the effect of the leader agents on the resulting error.

Contents

Chapter 1

Introduction

Multi-agent systems capture the behaviour and dynamics of connected systems in many different fields. For example these can be technological networks, such as the Internet (data connections between computers and related devices), or power grids. But these can also be biological networks. We can for example take a neural network, a system of interconnecting neurons. More background and different examples can be found in [8].

Because most multi-agent systems are quite large (the average human brain for example consists of more than 80 billion neurons) analysing and controlling of these systems can take a great deal of computational power. Therefore it is beneficial to use reduced order models, which have the same properties and spatial structure as the original network. This thesis closely follows Nima Monshizadeh's PhD thesis [6] who has described these multi-agent systems, has given a method to reduce them and has derived a simple expression for the model reduction error.

A multi-agent system is a network consisting of interconnected *agents*, which we can represent as a graph. In this graph the agents are indicated as *vertices* and the communication links between the agents are represented as *edges*. In this thesis we are working with networks that are associated with a leader-follower set up. These are networks in which some agents, called leaders, receive an external input (or disturbance) and the remaining agents, called followers, only receive information from their neighbours in the graph. The model reduction technique that is used in this thesis is called the *Petrov-Galerkin projection*, which is quite a general projection as many other reduction methods are specific cases of this projection.

When the Petrov-Galerkin projection is applied directly, the network structure will in general collapse. In fact, the reduced order model may not even refer to a multi-agent system anymore. Because we want to preserve as much properties of the original system as possible, graph partitions are introduced.

A graph partition is a collection of clustered individual graph nodes. When graph partitions are used to obtain the reduced order system, every cell of the partition, so every cluster of the original graph, relates to a single node in the reduced order graph.

A special class of graph partitions can be distinguished, the almost-equitable partitions. Almost-equitable partitions are a collection of clusters of nodes that are connected to the graph in a similar way. When given a set of vertices, a leader set and an almost equitable partition, the error between the original and reduced order system can be determined easily, using an expression derived by Monshizadeh [6]. This error expression is given in terms of the $\mathcal{H}_2$-norm [1]. Although it can be derived easily, no further conclusions can be drawn from this model reduction error for an almost equitable partition. For example, it is not implied that an almost equitable partition leads to a smaller model reduction error (or a larger error for that matter) than an arbitrary partition.

There is unfortunately no such simple error expression for the model reduction error associated with an arbitrary (so not necessarily almost equitable) partition. Monshizadeh and van der Schaft [7] have used almost equitable partitions to derive an explicit expression for the model reduction error for graphs associated with weights both to the edges and to the vertices. Interesting enough, this error does not depend on the weights associated to the edges, but only on the weights associated to the vertices of the graphs. Jongsma et al. [5] have given an upper bound for the model reduction error for an arbitrary partition, by approximating the original system by a new system, for which the chosen partition is almost equitable, and using the triangle inequality. Jongsma [4] also introduced a new method for model reduction, which is based on removing edges that close cycles. For this method an explicit error expression and upper bound in terms of the eigenvectors of the edge Laplacian matrices (of the original and reduced system) is given.

Because there is not much known about the model reduction error for systems associated with arbitrary partitions, in this thesis we are interested in this error. An algorithm is written to numerically obtain this model reduction error. Because the $\mathcal{H}_2$-norm can only be derived for stable systems, similarity is used to eliminate the unobservable part of our systems. Similarity does not change transfer functions, so the similar system can be used to determine the $\mathcal{H}_2$-norm and hence the model reduction error.

First, given a graph, a leader set and an almost equitable partition, some observations about the model reduction error are made when a single edge is removed. When this edge is not contained in an individual cell (or cluster), the (originally almost equitable) partition becomes non almost equitable. Then an algorithm is given to compute the partition that leads to the smallest model reduction error for a given graph, leader set and number of nodes in the reduced order model.

Chapter 2

Preliminaries and defining the system

In this chapter some preliminaries are provided that are needed further on. We recall some notions about graph theory and describe the multi-agent system that is used in this thesis.

2.1 Graph theory

A multi-agent system, or a network, can be seen as a graph. This is a collection of vertices joined by edges, denoted as the triple $G = (V, E, A)$. In this triple $V = \{1, 2, \ldots, n\}$ is the set of n vertices (or agents), E is the set of k edges (the connections between individual agents) and A is the adjacency matrix. For *unweighted* graphs the adjacency matrix A is the square n-by-n matrix with elements a_{ij} such that

$$a_{ij} = \begin{cases} 1 & \text{if there is an edge between vertices } i \text{ and } j \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

Unweighted graphs are associated with networks that have simple on/off connections between vertices; so there is a connection between agents, or there is not. However, sometimes it can be useful to represent the edges with a strength, or a weight. Therefore we can distinguish networks associated with a *weighted* graph. For weighted graphs the adjacency matrix A has elements a_{ij} such that a_{ij} is the corresponding weight of the edge between vertices i and j. In this thesis we only consider *simple* graphs, so graphs without multiple edges or self loops.

A *directed graph* is a network in which each edge has a direction from vertex i to vertex j. A directed graph is called *symmetric* if whenever (i, j) is an edge, also (j, i) is (but weights the corresponding a_{ij} and a_{ji} can be distinct). Remark that an undirected graph can be seen as a symmetric directed graph where for any i, j the equality $a_{ij} = a_{ji}$ holds.

Both for directed and undirected graphs the degree matrix of G can be formed. This degree matrix is a diagonal matrix $D = diag(d_1, d_2, \ldots, d_n)$, with

$$d_i = \sum_{j=i}^{n} a_{ij}. \tag{2.2}$$

The Laplacian matrix can now be defined as $L = D - A$. This Laplacian matrix is an important matrix, because it tells us much about the network structure. For directed graphs the incidence matrix R can be defined, which has elements r_{ij} such that

$$r_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ is the head of arc } j \\ -1 & \text{if vertex } i \text{ is the tail of arc } j \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

Even for undirected graphs an incidence matrix can be obtained, if we assign an arbitrary orientation to the edges. When given the set of k edges we can indicate the diagonal matrix

$$W = diag(w_1, w_2, \ldots, w_k). \tag{2.4}$$

Where w_j specifies the weight corresponding to the j^{th} edge. For undirected graphs there is now the following relation between the graph Laplacian and the incidence matrix

$$L = RWR^T \tag{2.5}$$

2.2 Leader-follower multi-agent systems

We can consider a standard multi-agent system Σ as

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.6}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input and $y \in \mathbb{R}^p$ is the output of the system. Remark that in this system matrix A is a variable matrix and therefore different from the adjacency matrix defined earlier. A shorthand notation for this system Σ in (2.6) is:

$$\Sigma = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right]. \tag{2.7}$$

In a leader-follower multi-agent system only agents called leaders receive external input. That is defined in the following dynamical system

$$\dot{x}_i = \begin{cases} z_i & \text{if vertex } i \text{ is a follower} \\ z_i + u_l & \text{if vertex } i \text{ is a leader} \end{cases} \tag{2.8}$$

where $x_i \in \mathbb{R}$ is the state of the agent i, $u_l \in \mathbb{R}$ the external input applied to leader i and z_i denotes the coupling variable for agent i and is given by

$$z_i = \sum_{j=1}^{n} a_{ij}(x_j - x_i). \tag{2.9}$$

Given a weighted undirected graph $G = (V, E, A)$ and given m agents that act as leaders, the set of leaders $V_L \subseteq V$ is denoted as $V_L = \{v_1, v_2, \dots, v_m\}$. From this leader set the matrix $M \in \mathbb{R}^{n \times m}$ can be defined as

$$M_{il} = \begin{cases} 1 & \text{if } i = v_l \\ 0 & \text{otherwise.} \end{cases} \tag{2.10}$$

The input/state system of the leader-follower multi-agent system associated with weighted undirected graph G can now be written as

$$\dot{x} = -Lx + Mu \tag{2.11}$$

where M is defined as in (2.10) and L is the Laplacian matrix of G as in (2.5).

Since the differences of the states of communicating agents are embedded in the incidence matrix (2.3), output variables y can be chosen as

$$y = W^{\frac{1}{2}} R^T x \tag{2.12}$$

Combining (2.11) and (2.12) gives us the following input/state/output model for the leader-follower multi-agent system

$$\begin{aligned} \dot{x} &= -Lx + Mu \\ y &= W^{\frac{1}{2}} R^T x \end{aligned} \tag{2.13}$$

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$ and output $y \in \mathbb{R}^p$.

Example 2.2.1. *We can give an example of a leader-follower multi-agent system associated with a weighted undirected graph G, with $n = 10$ vertices, Laplacian matrix L, and $m = 2$ leaders: $V_L = \{2, 7\}$.*

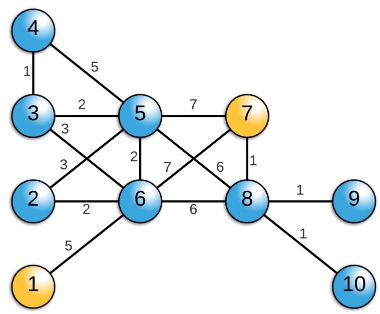

Figure 2.1: Network graph of example 2.2.1

$$L = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & -5 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & -3 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & -1 & -2 & -3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 6 & -5 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & -2 & -5 & 25 & -2 & -6 & -7 & 0 & 0 \\ -5 & -2 & -3 & 0 & -2 & 25 & -6 & -7 & 0 & 0 \\ 0 & 0 & 0 & -6 & -6 & -1 & 15 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -7 & -7 & -1 & 15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \qquad M = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Chapter 3

Model reduction

In the previous chapter we have defined the leader-follower multi-agent system which indicates the original system. In this section a reduced order system is obtained. First the model reducing technique called the Petrov-Galerkin projection is explained and after that a technique for model reduction through graph partitioning is proposed.

3.1 Petrov-Galerkin projection

The model reduction technique we use to derive the reduced order model is defined in the following way

Definition 3.1.1. *Consider the standard input/state/output sytem as in* (2.6). *If we let* $\boldsymbol{W}$, $\boldsymbol{V} \in \mathbb{R}^{n \times r}$ *such that* $\boldsymbol{W}^T \boldsymbol{V} = I$, *then the matrix* $\Gamma = \boldsymbol{V}\boldsymbol{W}^T$ *is called the Petrov-Galerkin projector. The reduced order model*

$$\begin{aligned} \dot{\hat{x}} &= \boldsymbol{W}^T A \boldsymbol{V} \hat{x} + \boldsymbol{W}^T B u \\ \hat{y} &= C \boldsymbol{V} \hat{x} \end{aligned} \tag{3.1}$$

with state $\hat{x} \in \mathbb{R}^r$ *is called the Petrov-Galerkin projection of the original system.*

This projection technique is a general projection, many other model reduction techniques are similar to this projection, only with specific choices for matrices $\mathbf{V}$ and $\mathbf{W}$. When applied directly, this projection will in most cases destroy the spatial structure of the network. We wish to preserve the properties and structure of the original network, therefore a model reduction technique based on *graph partitions* is proposed.

3.2 Graph partitioning

In this section we will explain graph partitions and a special class of graph partitions called almost equitable partitions. From a given graph partition the characteristic matrix can be formed, which can be used in the Petrov-Galerkin projection to obtain the reduced order system.

Given a graph G, a nonempty subset $C \subseteq V$ is called a *cell* or a *cluster* of vertex set V. A partition π of V is a collection of disjoint cells, whose union equals vertex set V. This can be defined as

Definition 3.2.1. *Given an undirected graph G with vertex set V, a collection of cells given by $\pi = \{C_1, C_2, \dots, C_r\}$ is called a partition of V if $\bigcup_{i=1}^{r} C_i = V$ and $C_i \bigcap C_j = \emptyset$ whenever $i \neq j$*

When given a vertex set $V = \{1, 2, \dots, n\}$, for a cell $C \subseteq V$ the characteristic vector of C is the column vector $p(C) \in \mathbb{R}^n$ with

$$p_i(C) = \begin{cases} 1 & \text{if } i \in C \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

Now for a partition $\pi = \{C_1, C_2, \dots, C_r\}$ the characteristic matrix P can defined as:

$$P(\pi) = \begin{bmatrix} p(C_1) & p(C_2) & \cdots & p(C_r) \end{bmatrix} \tag{3.3}$$

This can be illustrated with an example.

Example 3.2.2. *Take the system from example 2.2.1. A partition consisting of 5 cells can be $\pi = \{\{1, 2, 3, 4\}, \{5, 6\}, \{7\}, \{8\}, \{9, 10\}\}$. The corresponding characteristic matrix P is then given by:*

$$P(\pi) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For a partition $\pi = \{C_1, C_2, \ldots, C_r\}$, the degree of any node $i \in C_k$ relative to C_s is denoted as the sum of the weights of all edges from i to $j \in C_s$. So given the adjacency matrix (2.1), the degree of i relative to C_s is defined as $d(i, C_s) = \sum\limits_{j \in C_s} a_{ij}$.

A special class of graph partitions can be distinguished, almost equitable partitions. Each cell of an almost equitable partition contains agents that are connected to the network in approximately the same way. This can be formally defined as following.

Definition 3.2.3. *Let G be a graph with set of vertices V. We call π an almost equitable partition of G if for all distinct ordered pairs of cells (C_k, C_s) it holds that $d(i, C_s) = d(j, C_s)$ for all $i, j \in C_k$.*

This means that for unweighted graphs every vertex in C_k has an equal amount of neighbours in C_s and that holds for any pair (C_k, C_s) with $k \neq s$. For weighted graphs for any pair (C_k, C_s) every vertex in C_k has an equal total sum of the weights of edges to C_s. Observe that the partition defined in example 3.2.2 is an almost equitable partition.

An important property of an almost equitable partition π is that the image of the *characteristic matrix* $P(\pi)$ is invariant under the Laplacian matrix L. So L im $P(\pi) \subseteq$ im $P(\pi)$.

We are now able to use the characteristic matrix $P(\pi)$ of (3.3) as a choice for the Petrov-Galerkin projector in(3.1). Let matrices $\mathbf{V}$ and $\mathbf{W}$ such that

$$\begin{aligned} \mathbf{W} &= P(\pi)(P^T(\pi)P(\pi))^{-1} \\ \mathbf{V} &= P(\pi). \end{aligned} \tag{3.4}$$

It can be easily verified that indeed $\mathbf{W}^T\mathbf{V} = I$ holds. Note that this projection is only possible when $P^T(\pi)P(\pi)$ is non-singular. Since the columns of $P(\pi)$ are orthogonal, $P^T(\pi)P(\pi)$ is a diagonal matrix and hence invertible.

Recall the original system Σ:

$$\begin{aligned} \dot{x} &= -Lx + Mu \\ y &= W^{\frac{1}{2}}R^T x. \end{aligned} \tag{3.5}$$

If we apply the Petrov-Galerkin projection with matrices $\mathbf{W}$ and $\mathbf{V}$ from (3.4) above, we obtain the following reduced order system $\hat{\Sigma}$

$$\begin{aligned} \dot{\hat{x}} &= -\hat{L}\hat{x} + \hat{M}u \\ \hat{y} &= W^{\frac{1}{2}}\hat{R}^T. \end{aligned} \tag{3.6}$$

where, if we abbreviate $P(\pi)$ simply as P, matrices $\hat{L}$, $\hat{M}$ and $\hat{R}$ are denoted as

$$\begin{aligned}\hat{L} &= (P^TP)^{-1}P^TLP \\ \hat{M} &= (P^TP)^{-1}P^TM \\ \hat{R} &= P^TR.\end{aligned} \tag{3.7}$$

This reduced order model results in a new multi-agent system with a weighted symmetric directed graph (recall that our original system is associated with a weighted undirected graph). Because of the relation between L and $\hat{L}$ the rate of convergence of the reduced order model is at least as fast as the rate of convergence of the original system.

This can be clarified in an example.

Example 3.2.4. *Take the system with matrices L and M from example 2.2.1. Using the partition $\pi = \{\{1,2,3,4\},\{5,6\},\{7\},\{8\},\{9,10\}\}$ the following reduced order system with matrices $\hat{L}$ and $\hat{M}$ is obtained*

$$\hat{L} = \begin{bmatrix} 5 & -5 & 0 & 0 & 0 \\ -10 & 23 & -6 & -7 & 0 \\ 0 & -12 & 15 & -1 & -2 \\ 0 & -14 & -1 & 15 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} \qquad \hat{M} = \begin{bmatrix} 0.25 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

This results in the following reduced order model.

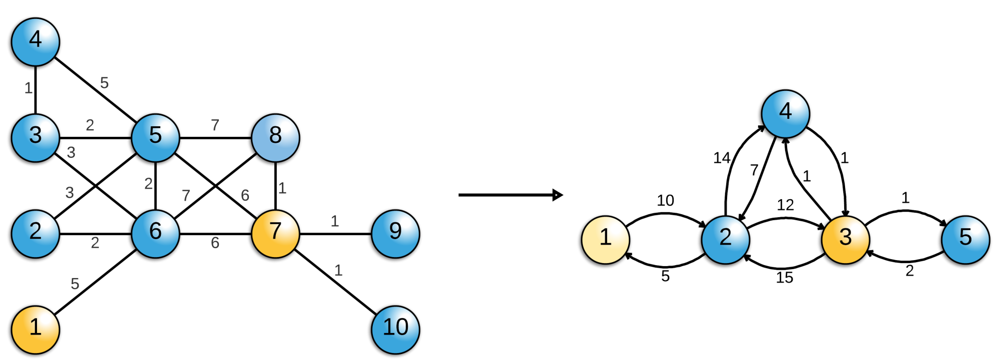

Figure 3.1: Reduced order graph from example 3.2.4

From this example it can be observed that every cluster from the original network corresponds to a node in the reduced order system. The weights of the arcs from node i to node j in the reduced system equal the total of weights corresponding to edges from cell C_i to C_j divided by the number of nodes in C_j. Also, the number of cell mates of a leader in the original system affect the leader in the reduced system. We have for for matrix $\hat{M} \in \mathbb{R}^{r \times m}$

$$\hat{M}_{pj} = \begin{cases} \frac{1}{|C_p|} & \text{if } v_j \in C_p \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

where $|C_p|$ is the cardinality of C_p (the number of nodes in C_p) and $v_1, v_2, \ldots, v_m$ are agents that are defined as leaders.

So observe that the cluster $\{1, 2, 3, 4\}$ contains agent 1, which is a leader for the system. This cluster corresponds to node 1 in the reduced system, which receives $\frac{1}{4}^{th}$ of the external input that agent 3 (which is deduced from node 7 in the original system, which is its own cluster) receives.

Chapter 4

Error computation

In this chapter we are interested in the error between the original and the reduced order system. This error is represented in terms of the transfer matrices of the systems and the $\mathcal{H}_2$-norm. When an almost equitable partition is used to reduce the system, the model reduction error can be easily analytically derived. There is however no such error expression for arbitrary partitions, so then the $\mathcal{H}_2$-norm has to be computed. Since this can only be done for stable systems, and that is not the case for our systems, we make use of similar systems.

For an input/state/output system $\Sigma = (A, B, C, 0)$ (in the form (2.6)) we denote the transfer matrix S from input u to output y as [3]

$$S(s) = C(sI - A)^{-1}B. \tag{4.1}$$

Recall that the trace of a square n-by-n matrix A is equal to the sum of the elements on the diagonal. So $tr(A) = \sum_{i=1}^{n} a_{ii}$.

Antoulas [1] proposes a formula to obtain the $\mathcal{H}_2$-norm for the system $\Sigma = (A, B, C, D)$:

$$\|\Sigma\|_{\mathcal{H}_2} = \sqrt{tr(B^T Q B)}. \tag{4.2}$$

This norm is bounded only when $D = 0$ and Σ is stable. So this norm is bounded if every eigenvalue of A has a strictly negative part (in other words: matrix A is Hurwitz).

When Σ is stable the matrix Q is a unique solution of the Lyapunov equation:

$$A^T Q + QA = -C^T C \tag{4.3}$$

When using partition π to obtain the reduced order model, we can now represent the model reduction error $\Xi(\pi)$ in terms of the $\mathcal{H}_2$-norm:

$$\Xi(\pi) = \frac{\|S - \hat{S}\|_2^2}{\|S\|_2^2} \tag{4.4}$$

where S denotes the transfer function of the original system Σ and $\hat{S}$ denotes the transfer function of the reduced order system $\hat{\Sigma}$. Note that $\Xi(\pi)$ is normalized, so $0 \leq \Xi(\pi) \leq 1$.

4.1 Almost equitable partitions

For almost equitable partition the model reduction error can be determined easily, without needing to compute the $\mathcal{H}_2$-norm. Monshizadeh [6] presents following theorem:

Theorem 4.1.1. *Let G be a weighted undirected graph, and assume G is connected. Let $\pi = \{C_1, C_2, ..., C_r\}$ be an almost equitable partition of G. Suppose that the reduced order multi-agent system* (3.6) *is obtained from* (2.13) *using partition π. Also, let S and $\hat{S}$ denote the transfer matrices from u to y in* (2.13) *and* (3.6) *respectively. Then, we have*

$$\frac{\|S - \hat{S}\|_2^2}{\|S\|_2^2} = \frac{\sum_{i=1}^{m}(1 - \frac{1}{|C_{k_i}|})}{m(1 - \frac{1}{n})} \tag{4.5}$$

Here m represents the number of leaders and n equals the amount of agents (or vertices).

So for the reduced order system from example 3.2.4 the relative error between the original and reduced order network is $\Xi(\pi) = 0.4167$. Note that no further claims are made about almost equitable partitions. It is for example not the optimal partition one can chose, as it does not have to result in the smallest possible model reduction error. Therefore, we are interested in the relative error obtained from a partition.

4.2 Arbitrary partitions

If the partition used is not necessarily almost equitable we do have to determine the $\mathcal{H}_2$-norm. We have to compute this norm for the transfer matrix of the original system S and of the difference between the transfer matrix of the original system and the transfer matrix of the reduced order system $S - \hat{S}$. In order to do the latter we want to write $S - \hat{S}$ as a single error system $\bar{S}$, where $\bar{S} = \bar{C}(sI - \bar{A})^{-1}\bar{B}$.

So

$$\begin{aligned}
\bar{S} &= S - \hat{S} \\
&= C(sI-A)^{-1}B - \hat{C}(sI-\hat{A})^{-1}\hat{B} \\
&= \begin{bmatrix} C(sI-A)^{-1} & -\hat{C}(sI-\hat{A})^{-1} \end{bmatrix} \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \\
&= \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} (sI-A)^{-1} & 0 \\ 0 & (sI-\hat{A})^{-1} \end{bmatrix} \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \\
&= \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} (sI-A) & 0 \\ 0 & (sI-\hat{A}) \end{bmatrix}^{-1} \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \\
&= \begin{bmatrix} C & -\hat{C} \end{bmatrix} (sI - \begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix})^{-1} \begin{bmatrix} B \\ \hat{B} \end{bmatrix}.
\end{aligned} \tag{4.6}$$

We have obtained our desired form

$$\bar{S} = \bar{C}(sI - \bar{A})^{-1}\bar{B} \tag{4.7}$$

where

$$\bar{A} = \begin{bmatrix} A & 0 \\ 0 & \hat{A} \end{bmatrix}, \qquad \bar{B} = \begin{bmatrix} B \\ \hat{B} \end{bmatrix}, \qquad \bar{C} = \begin{bmatrix} C & -\hat{C} \end{bmatrix}. \tag{4.8}$$

Recall the original system Σ and reduced multi-agent system $\hat{\Sigma}$:

$$\begin{aligned}
\dot{x} &= -Lx + Mu \\
y &= W^{\frac{1}{2}}R^T x \\
\\
\dot{\hat{x}} &= -\hat{L}\hat{x} + \hat{M}u \\
\hat{y} &= W^{\frac{1}{2}}\hat{R}^T
\end{aligned} \tag{4.9}$$

Since the $n \times n$ Laplacian matrix is obtained as $L = D - A$, we have $L\mathbb{1}_n = \mathbb{0}_n$. Therefore zero is an eigenvalue of the Laplacian matrix, with corresponding eigenvector $\mathbb{1}_n$. All the eigenvalues of the Laplacian matrix are non-negative [2]. Therefore $-L$ and $-\hat{L}$ each possess a single zero eigenvalue, and the remaining eigenvalues of $-L$ and $-\hat{L}$ are all strictly negative. Due to the zero eigenvalues we are not able to compute the $\mathcal{H}_2$-norm of the systems. Therefore we can use the Kalman decomposition:

Theorem 4.2.1. *If for the system $\Sigma = (A, B, C, 0)$ the pair (A, C) is unobservable, then there exists a non-singular T such that*
$$\left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right] \sim \left[\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & 0 \end{array}\right] = \left[\begin{array}{cc|c} A_{11} & 0 & B_1 \\ A_{21} & A_{22} & B_2 \\ \hline C_1 & 0 & 0 \end{array}\right]$$
where (A_{11}, C_1) is observable

Since similarity does not change the transfer function, we may use the transfer function of the similar system. Using the Banachiewicz-Schur form to compute the inverse of a partitioned matrix [9] we can derive the transfer function of the similar system as follows:

$$\begin{aligned} S &= C(sI - A)^{-1}B \\ &= CT(sI - T^{-1}AT)^{-1}T^{-1}B \\ &= \begin{bmatrix} C_1 & 0 \end{bmatrix} \begin{bmatrix} sI - A_{11} & 0 \\ -A_{21} & sI - A_{22} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \\ &= \begin{bmatrix} C_1 & 0 \end{bmatrix} \begin{bmatrix} sI - A_{11}^{-1} & 0 \\ (sI - A_{22})^{-1}A_{21}(sI - A_{11})^{-1} & sI - A_{22}^{-1} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \\ &= C_1(sI - A_{11})^{-1}B_1. \end{aligned} \tag{4.10}$$

Remark that the pair (C_1, A_{11}) is observable, so if we find matrices T and $\hat{T}$ for our original system Σ and reduced order system $\hat{\Sigma}$ we are able to compute the $\mathcal{H}_2$-norm of the original system and error system, and hence compute the model reduction error.

Since 0 is the only unobservable eigenvalue for the Laplacian matrix, we use for T the corresponding eigenvector $\mathbb{1}$. So $T = [\tilde{T}\ \mathbb{1}]$ and choose $\tilde{T}$ such that T is non singular. Therefore for our original system we can use for T and T^{-1}:

$$T = \begin{bmatrix} I_{n-1} & \mathbb{1}_{n-1} \\ 0_{1\times n-1} & 1 \end{bmatrix}, \qquad T^{-1} = \begin{bmatrix} I_{n-1} & -\mathbb{1}_{n-1} \\ 0_{1\times n-1} & 1 \end{bmatrix}. \tag{4.11}$$

For the reduced order system, where r is the number of cells in partition π and hence the number of agents in the reduced order system:

$$\bar{T} = \begin{bmatrix} I_{r-1} & \mathbb{1}_{r-1} \\ 0_{1\times r-1} & 1 \end{bmatrix}, \qquad \bar{T}^{-1} = \begin{bmatrix} I_{r-1} & -\mathbb{1}_{r-1} \\ 0_{1\times r-1} & 1 \end{bmatrix}. \tag{4.12}$$

This means that for the leader follower multi-agent system in this thesis

$$\left[\begin{array}{c|c} -L & M \\ \hline C & 0 \end{array}\right] \sim \left[\begin{array}{c|c} -T^{-1}LT & T^{-1}M \\ \hline CT & 0 \end{array}\right] \tag{4.13}$$

where we set $C = W^{\frac{1}{2}}R^T$ in order to make the equations somewhat more readable and for the similar system

$$-T^{-1}LT = \begin{bmatrix} -L_{11} & 0 \\ -L_{21} & -L_{22} \end{bmatrix} \qquad T^{-1}M = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \qquad CT = \begin{bmatrix} C_1 & 0 \end{bmatrix} \tag{4.14}$$

and since $(-L_{11}, C_1)$ is observable we can solve the Lyapunov equation:

$$-{L_{11}}^T Q - QL_{11} = -C_1^T C_1 \tag{4.15}$$

and use Q to obtain the $\mathcal{H}_2$-norm:

$$\|\Sigma\|_{H_2} = \sqrt{tr(M_1^T Q M_1)} \tag{4.16}$$

We can do the same for the error system $\bar{S}$ of (4.7). This gives us the following expression for the model reduction error:

$$\frac{\|S - \hat{S}\|_2^2}{\|S\|_2^2} = \frac{tr(\bar{M}_1^T \bar{\mathbb{Q}} \bar{M}_1)}{tr(M_1^T \mathbb{Q} M_1)} \tag{4.17}$$

Chapter 5

Experiments with arbitrary partitions

Since we are now able to compute the model reduction error, given a graph, a leader set and an arbitrary partition, we would like to do some experiments with partitions that are not almost equitable. Recall that a partition is almost equitable if for all distinct ordered pairs of cells (C_r, C_s) every vertex in C_r has the same number of neighbours in C_s. For weighted graphs this means the degree should be equal.

First, we look at partitions that are close to almost equitable. We do this by removing single edges from a given graph and an almost equitable partition, on condition that this leaves the graph connected. Then we would like to do some experiments with arbitrary partitions and investigate which partitions lead to the smallest model reduction error. Since we have obtained a method for determining the model reduction error, given a graph and a leader set we can numerically try al possible partitions and hence attain the partition that leads to the smallest model reduction error.

5.1 Removing edges

So first we investigate errors of partitions that are close to almost equitable. For this we start with a given graph and an almost equitable partition and then remove a single edge from the network graph. This is done for every edge, on condition that the resulting network graph is connected as well. Note that if we remove an edge between two vertices contained in the same cell, the partition remains almost equitable and hence the model reduction error does not change.

Recall the system from example 2.2.1 which possesses the almost equitable partition $\pi = \{\{1, 2, 3, 4\}, \{5, 6\}, \{7\}, \{8\}, \{9, 10\}\}$. This system is associated with the network graph depicted in figure 5.1.

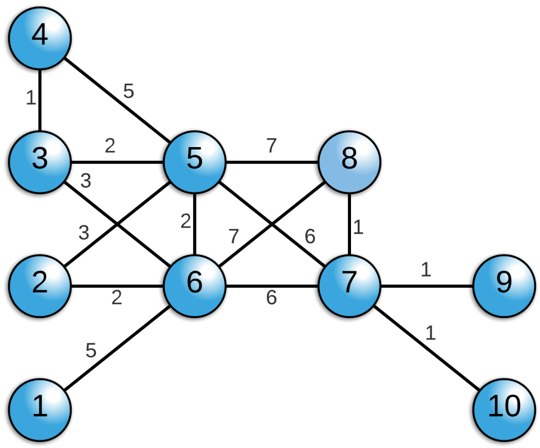

Figure 5.1: Weighted, undirected graph

Let us start with assigning agents 1 and 7 as leaders. For the model reduction error for this system and partition we obtain for removed edges.

Removed edge	Error
-	0.4167
(2,6)	0.4034
(2,5)	0.4078
(3,6)	0.3963
(3,5)	0.4109
(3,4)	0.4167
(4,5)	0.3984
(5,6)	0.4167
(5,8)	0.4106
(5,7)	0.4389
(6,7)	0.4559
(6,8)	0.4253
(7,8)	0.4167

Table 5.1: Relative errors for leader set (1,7)

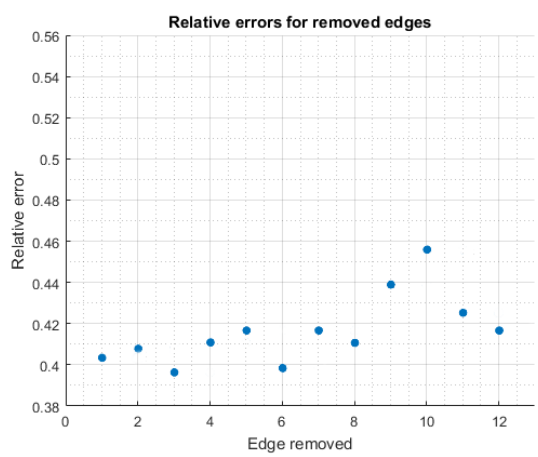

Figure 5.2: Relative error for removed edges

Although there are no substantial differences between the errors, it seems that removing an edge that is connected to a leader results in a larger model reduction error. To test this we can add edges (1,2) and (9,10) with corresponding weights equal to 1. This yields in a network graph where more edges attached to a leader can be removed without a resulting disconnected graph. These extra edges do not change the almost equitable partition.

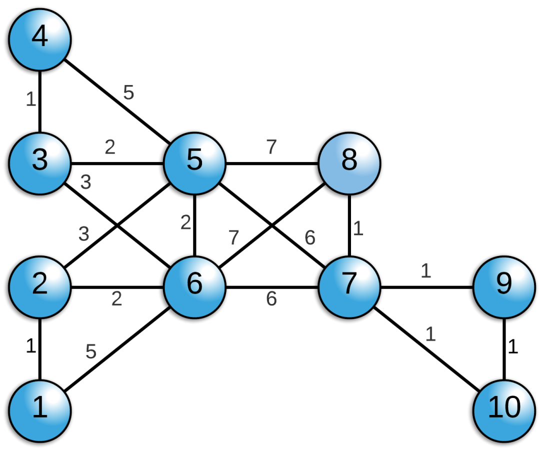

Figure 5.3: Weighted, undirected graph with added edges

Removed edge	**Error**
-	0.4167
(1,2)	0.4167
(1,6)	0.5519
(2,6)	0.4072
(2,5)	0.4131
(3,6)	0.3969
(3,5)	0.4105
(3,4)	0.4167
(4,5)	0.3976
(5,6)	0.4167
(5,8)	0.4114
(5,7)	0.4401
(6,7)	0.4546
(6,8)	0.4245
(7,8)	0.4167
(7,9)	0.4222
(7,10)	0.4222
(9,10)	0.4167

Table 5.2: Relative errors for leader set (1,7)

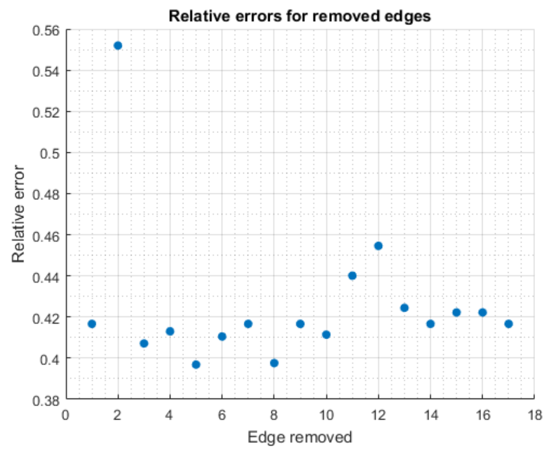

Figure 5.4: Relative error for removed edges

This seems to support the premise that removing an edge attached to a leader results in a larger model reduction error. We now take the system from figure 5.3 and assign the leaders to $V_L = \{1, 8\}$.

Removed edge	Error
-	0.4167
(1,2)	0.4167
(1,6)	0.5532
(2,6)	0.4073
(2,5)	0.4135
(3,6)	0.3973
(3,5)	0.4107
(3,4)	0.4167
(4,5)	0.3989
(5,6)	0.4167
(5,8)	0.4581
(5,7)	0.4117
(6,7)	0.4262
(6,8)	0.4711
(7,8)	0.4167
(7,9)	0.4213
(7,10)	0.4213
(9,10)	0.4167

Table 5.3: Relative errors for leader set (1,8)

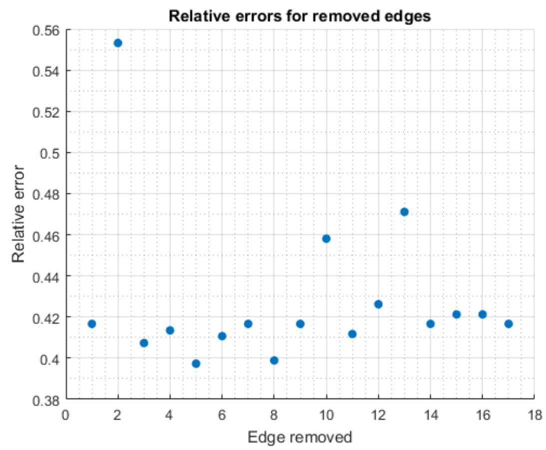

Figure 5.5: Relative error for removed edges

Now let us take the same system from figure 5.3 and add an extra leader and set $V_L = \{1, 5, 7\}$.

Removed edge	**Error**
-	0.4630
(1,2)	0.4630
(1,6)	0.5397
(2,6)	0.4489
(2,5)	0.4752
(3,6)	0.4389
(3,5)	0.4681
(3,4)	0.4630
(4,5)	0.4763
(5,6)	0.4630
(5,8)	0.4877
(5,7)	0.5045
(6,7)	0.4659
(6,8)	0.4434
(7,8)	0.4630
(7,9)	0.4682
(7,10)	0.4682
(9,10)	0.4630

Table 5.4: Relative errors for leader set (1,5,7)

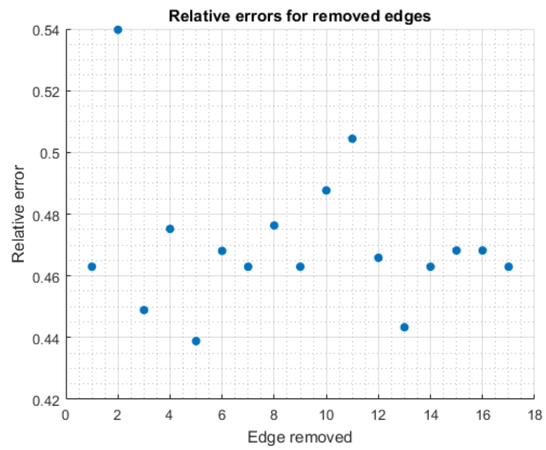

Figure 5.6: Relative error for removed edges

These results confirm the proposition that a removed edge from a leader results in a larger model reduction error as well.

5.2 Changing partitions

Now that we have investigated model reduction errors obtained from partitions close to almost equitable, we are interested in finding the smallest possible model reduction error for a given system. This can be done by computing the error as in the previous chapter for all possible graph partition for a given graph. In this section we consider a network associated with a weighted graph and a network associated with an unweighted graph as well. Experiments with a different number of leaders and different number of agents in the reduced system are performed on these networks and we draw conclusions about this.

5.2.1 Weighted graphs

Recall once again the system defined in example 2.2.1:

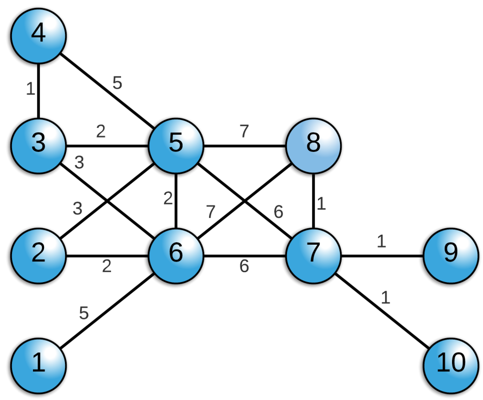

Figure 5.7: Weighted, undirected graph

For this graph we wish to determine the smallest possible model reduction error for a given number of agents in the reduced network and a given leader set. This results in the following.

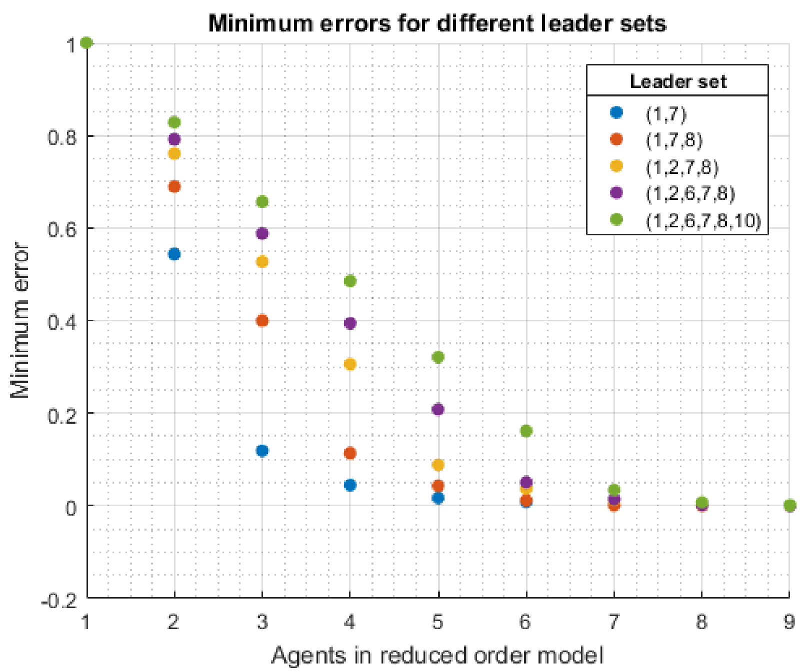

Figure 5.8: Minimum errors for the system from figure 5.7

Table 5.5: The mimimal relative errors for the graph from figure 5.7

		Leaders				
	r	(1,7)	(1,7,8)	(1,2,7,8)	(1,2,6,7,8)	(1,2,6,7,8,10)
$\boldsymbol{\Xi(\pi)}$	1	1	1	1	1	1
	2	0.54372	0.68945	0.76078	0.79158	0.82812
	3	0.11886	0.40003	0.5274	0.58815	0.65694
	4	0.044731	0.11373	0.30608	0.39438	0.48531
	5	0.016874	0.042981	0.088316	0.20789	0.32109
	6	0.0084646	0.011935	0.037095	0.050867	0.16188
	7	0.0017718	0.0011812	0.012905	0.014562	0.034268
	8	0.00011168	7.4452×10^{-5}	0.00088353	0.001937	0.007606
	9	-3.7007×10^{-16}	-2.4672×10^{-16}	-3.084×10^{-16}	-3.2073×10^{-16}	0.0016141

Table 5.6: Some partitions corresponding to table 5.5

		Leaders	
	r	(1,7)	(1,7,8)
$\boldsymbol{\pi}$	1		
	2	(1,2,3,4,5,6,8,9,10),(7)	(1,2,3,4,5,6,8,9,10),(7)
	3	(1),(2,3,4,5,6,8,9,10),(7)	(1),(2,3,4,5,6,8,9,10),(7)
	4	(1),(2,3,4,5,8,9,10),(6),(7)	(1),(2,3,4,5,6,9,10),(7),(8)
	5	(1),(2,3,4,9,10),(5,8),(6),(7)	(1),(2,3,4,9,10),(5,6),(7),(8)
	6	(1),(2,3,4),(5,8),(6),(7),(9,10)	(1),(2,3,4,9,10),(5),(6),(7),(8)
	7	(1),(2,3,4),(5),(6),(7),(8),(9,10)	(1),(2,3,4),(5),(6),(7),(8),(9,10)
	8	(1),(2,3),(4),(5),(6),(7),(8),(9,10)	(1),(2,3),(4),(5),(6),(7),(8),(9,10)

This plot has the relative errors and best possible partitions above, where **r** denotes the number of agents in the reduced system. What stands out is that small model reduction errors are possible, smaller errors than obtained from an almost equitable partition, or close to an almost equitable partition. Another thing that can be concluded we can observe that when leaders are clustered in their own cell (so when they do not have cell mates) the model reduction error becomes relatively small (<0.15).

Recall the reduced order system obtained from an almost equitable partition with 5 cells from example 3.2.4. It can be determined that the partition that consists of 5 cells and results in the minimal model reduction error is $\pi = \{\{1\}, \{2, 3, 4, 9, 10\}, \{6\}, \{7\}\}$. When using this partition the model reduction error is 0.01687. The resulting matrices $\hat{L}$ and $\hat{M}$ are given below and the resulting reduced order network graph is depicted in figure 5.9.

$$\hat{L} = \begin{bmatrix} 5 & 0 & 0 & -5 & 0 \\ 0 & 3.4 & -2 & -1 & -0.4 \\ 0 & -5 & 13 & -4.5 & -3.5 \\ -5 & -5 & -9 & 25 & -6 \\ 0 & -2 & -7 & -6 & 15 \end{bmatrix} \qquad \hat{M} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

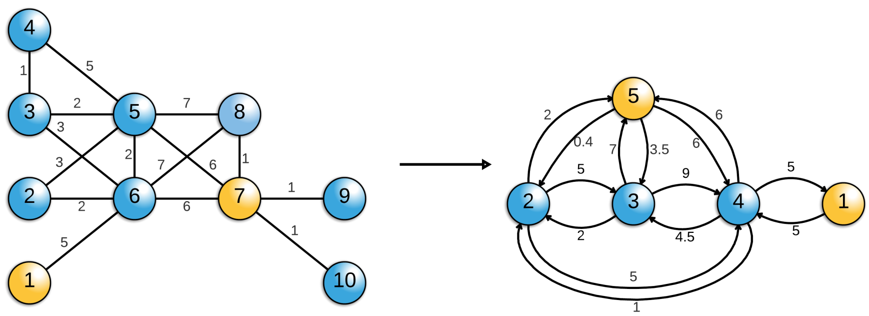

Figure 5.9: Reduced order graph obtained from the optimal partition

5.2.2 Unweighted graphs

Since we could also vary the weights of a graph, which results in a different network and a non almost equitable partition, let us now try the same for an unweighted graph. Take for example the graph given in figure 5.10.

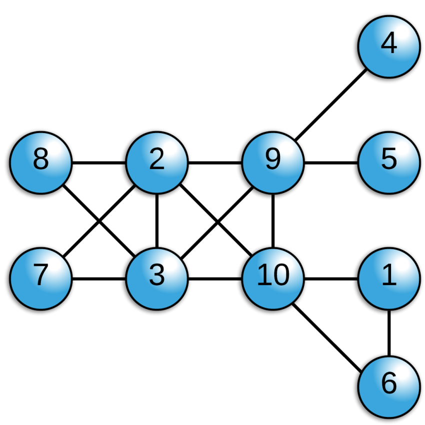

Figure 5.10: An unweighted, undirected graph

Observe that this network graph possesses almost equitable partition $\pi = \{\{1,6,4,5\},\{2,3\},\{7,8\},\{9,10\}\}$. Using this partition the model reduction error for different leader sets is given in table 5.7 below.

Table 5.7: Errors obtained from an almost equitable partition

Leaders	(1,7)	(1,7,8)	(1,2,7,8)	(1,2,7,8,10)	(1,2,6,7,8,10)
$\Xi(\pi)$	0.6944	0.6481	0.6250	0.6111	0.6481

For the graph depicted in figure 5.10 we wish to determine the smallest possible model reduction error for a given number of agents in the reduced network and a given leader set as well. This results in the following:

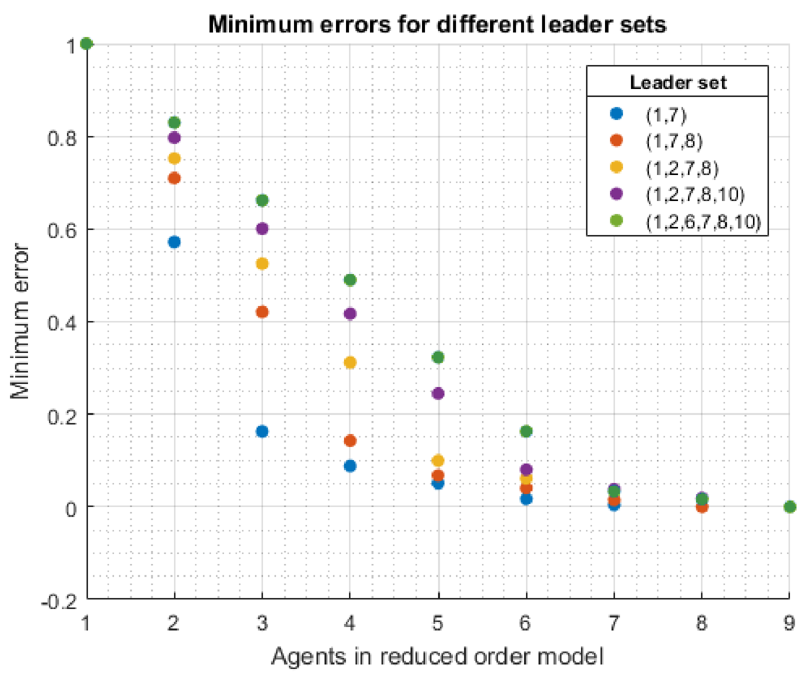

Figure 5.11: Minimum errors for the system from figure 5.10

Table 5.8: Minimum errors for the system from figure 5.10

	r	**Leaders** (1,7)	(1,7,8)	(1,2,7,8)	(1,2,6,7,8)	(1,2,6,7,8,10)
$\boldsymbol{\Xi(\pi)}$	1	1	1	1	1	1
	2	0.57186	0.70969	0.75253	0.79708	0.82981
	3	0.16281	0.42094	0.52517	0.60079	0.66205
	4	0.088079	0.14269	0.31196	0.41713	0.49005
	5	0.050894	0.067763	0.099777	0.24472	0.32288
	6	0.017548	0.040714	0.061395	0.080306	0.16308
	7	0.0044701	0.014966	0.03592	0.038029	0.033511
	8	-3.084×10^{-16}	-2.8784×10^{-16}	0.017906	0.019113	0.01666
	9	-3.084×10^{-16}	-2.8784×10^{-16}	3.084×10^{-17}	-7.7099×10^{-17}	-1.6705×10^{-16}

Table 5.9: Some corresponding partitions to table 5.8

	r	**Leaders** (1,7)	(1,7,8)
$\boldsymbol{\pi}$	1		
	2	(1,2,3,4,5,6,8,9,10),(7)	(1,2,3,4,5,6,7,9,10),(8)
	3	(1),(2,3,4,5,6,8,9,10),(7)	(1,2,3,4,5,6,9,10),(7),(8)
	4	(1),(2,3,4,5,8,9),(6,10),(7)	(1),(2,3,4,5,6,9,10),(7),(8)
	5	(1),(2,3),(4,5,8,9),(6,10),(7)	(1),(2,3),(4,5,6,9,10),(7),(8)
	6	(1),(2,3),(4,5,8,9),(6),(7),(10)	(1),(2,3),(4,5,9),(6,10),(7),(8)
	7	(1),(2,3),(4,5),(6),(7),(8,9),(10)	(1),(2,3),(4,5,9),(6),(7),(8),(10)
	8	(1),(2,3),(4,5),(6),(7),(8),(9),(10)	(1),(2,3),(4,5),(6),(7),(8),(9),(10)

This plot has the relative errors and best possible partitions above, where **r** denotes again the number of agents in the reduced system.

These results confirm the assumption we had in the previous section. Placing leaders in their own cell result in a small model reduction error. Also, an almost equitable is far from the best partition if one wishes to obtain a relative small model reduction error.

Chapter 6

Conclusions

In this thesis we examined the model reduction errors for multi-agent systems associated with a leader follower set up. These systems were reduced through graph partitioning, and if this partitioning is so called almost equitable, there exists a simple expression for the model reduction error. We examined partitions close to almost equitable by removing single edges from the original network graph. It can be observed that removing edges connected to a leader leads to a larger model reduction error.

Also, when given a system, a set of leaders and a number of agents for the reduced order system, this thesis provides an algorithm for determining the ideal partition for reducing the original system. It can be observed that placing the leaders of the system in their own cluster (so without any cell mates) generally leads to a smaller model reduction error. It can also be observed that an almost equitable partition is far from the best possible partition if one wants so keep the model reduction error relatively low.

This thesis only explored model reduction errors for small-scale systems. Since most multi-agents are usually quite large, investigating minimal model reduction errors for large-scale systems is a suggestion for further research.

Bibliography

[1] A.C. Antoulas. *Approximation of Large-scale Dynamical Systems*. Advances in Design and Control. Society for Industrial and Applied Mathematics, 2005. ISBN: 9780898715293.

[2] F.R.K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[3] J.P. Hespanha. *Linear Systems Theory*. Princeton, New Jersey: Princeton Press, 2009.

[4] H.J. Jongsma. "Robust Synchronization and Model Reduction of Multi-Agent Systems". PhD thesis. University of Groningen, 2017.

[5] H.J. Jongsma et al. "Model Reduction of Linear Multi-Agent Systems by Clustering and Associated $\mathcal{H}_2$-and $\mathcal{H}_\infty$-Error Bounds". In: *arXiv preprint arXiv:1610.02432* (2016).

[6] N. Monshizadeh. "Model Reduction and Control of Complex Systems". PhD thesis. University of Groningen, 2013.

[7] N. Monshizadeh and A.J. van der Schaft. "Structure-preserving model reduction of physical network systems by clustering". In: *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE. 2014, pp. 4434–4440.

[8] M. Newman. *Networks: an Introduction*. Oxford University Press, 2010.

[9] Y. Tian and Y. Takane. "Schur Complements and Banachiewicz-Schur Forms". In: *Electronic Journal of Linear Algebra* 13 (2005), pp. 405–418.

Chapter 7

Appendix

The MATLAB code used in this thesis to determine the partitions that result in the minimal error for a given number of nodes in the reduced order system.

```
% Algorithm for determining the minimal error and
   corresponding partition
% given a system and a leader set.

% Making use of the function partitions.m by Matt Fig
% (see
% https://nl.mathworks.com/matlabcentral/fileexchange
   /24185-partitions)

% Define the original system
v       =   [1 2 3 4 5 6 7 8 9 10];
s       =   [1 1  2 2 2 2  3 3 3 3  4 5 6  9 ];
t       =   [6 10 7 8 9 10 7 8 9 10 9 9 10 10];
weights =   [1 1  1 1 1 1  1 1 1 1  1 1 1  1 ];

G = graph(s,t,weights);
%plot(G,'EdgeLabel',G.Edges.Weight)

% Specify leaders
v_leader = [1 7]
%cells = 2;
nodenumber =  size(v,2);

%% determine the model reduction error for all
   possible partitions
for cells = 1:(nodenumber-1)
```

```
P_part = partitions(v,cells); % all possible
   partitions

% incidence matrix R
edgenumber = size(s,2);
nodenumber =  size(v,2);
R = zeros(nodenumber,edgenumber);
for i = 1:edgenumber
    R(s(i), i) = -1;
    R(t(i), i) = 1;
end
R ;

% diagonal matrix with weights on diagonal
W = diag(weights);

% adjacency matrix A (unweighted?)
A = adjacency(G);
A = full(A);

% degree matrix D
D = degree(G);
D = diag(D);

% laplacian matrix L
L = R*W*R';

% matrix C voor system (-L,M,C,0)
C = sqrt(W)*R.';

%% ------------------------------------------------
% for a given leader set 'v_leader' define matrix M
M = zeros(length(v),length(v_leader));
for c = 1:length(v_leader)
    for r = 1:length(v)
     if r == v_leader(c)
     M(r,c) = 1;
     end
    end
end
M;

    for j = 1:length(P_part)
% Given graph partition define characteristic matrix P
P = zeros(nodenumber,cells);
        for c = 1:cells % per kolom vullen
```

```
            for r = 1:nodenumber
              if ismember (r,P_part{j}{c})
              P(r,c)=1;
              end
            end
        end
P;

%% --------------------------
% projected system

% Petrov-Galerkin matrices W and V using graph
   partition P
W_proj    = P*inv(P.'*P);
V_proj    = P;

% Reduced order system 'hat'
L_hat   = W_proj.'*L*V_proj;
M_hat   = W_proj.'*M;

R_hat   = P.'* R ;
C_hat   = sqrt(W)*R_hat.';

%% define error system 'bar'
L_bar = blkdiag(L,L_hat);
M_bar = [M;M_hat];
C_bar = [C,-C_hat];

%% define matrices T for original, reduced system and
   error system
n            = length(L);
T            = [eye(n-1) ones(n-1,1); zeros(1,n-1) 1];
T_inv        = [eye(n-1) -ones(n-1,1); zeros(1,n-1) 1];

n_hat        = length(L_hat);
T_hat        = [eye(n_hat-1) ones(n_hat-1,1); zeros(1,
   n_hat-1) 1];
T_hat_inv    = [eye(n_hat-1) -ones(n_hat-1,1); zeros(1,
   n_hat-1) 1];

T_bar        = blkdiag(T,T_hat);
T_bar_inv    = blkdiag(T_inv,T_hat_inv);

%% original system, multiplication with T
L_tr         = T_inv*-L*T;
M_tr         = T_inv*M;
```

```
C_tr         = C*T;

L_block      = L_tr(1:n-1,1:n-1);

m            = size(M_tr,2);
M_block      = M_tr(1:n-1,1:m);

C_block      = C_tr(:,1:n-1) ;

%% H2 norm of original system (squared)
Q            = lyap(L_block.',(C_block.'*C_block));

norm_sys     = trace(M_block.'*Q*M_block);

%% error system multiplication with T_bar

L_bar_tr  = T_bar_inv*-L_bar*T_bar;
M_bar_tr  = T_bar_inv*M_bar;
C_bar_tr  = C_bar*T_bar;


L_bar_block = blkdiag( L_bar_tr(1:n-1,1:n-1), L_bar_tr
   (n+1:n+n_hat-1,n+1:n+n_hat-1));

M_bar_block  = [M_bar_tr(1:n-1,:) ; M_bar_tr(n+1:n+
   n_hat-1,:)];

C_bar_block = [C_bar_tr(:,1:n-1) , C_bar_tr(:,n+1:n+
   n_hat-1)];


%% H2 norm of error system (squared)
Q_bar = lyap(L_bar_block',(C_bar_block'*C_bar_block));

norm_error_sys = trace(M_bar_block'*Q_bar*M_bar_block)
   ;

%% Normalized error
relative_error = norm_error_sys/norm_sys;

errors(:,j)= relative_error;
    end

[minimum,I] = min(errors);
%partdisp(P_part(I));
```

```
[maximum,J] = max(errors);
%partdisp(P_part(J));

array_minimumerrors(cells) = minimum;
array_cells(cells) = cells;
disp(['The minimum error for a reduced order model
   with ',num2str(cells),' vertices is ',num2str(
   minimum)])
%partdisp(P_part(I))
end
array_minimumerrors;
array_cells;

hold on
grid on
grid minor
title('Minimum errors for different leader sets')
scatter(array_cells,array_minimumerrors,'filled')
xlabel('Agents in reduced order model')
ylabel('Minimum error')
lgd = legend('(1,7)','(1,7,8)','(1,2,7,8)','
   (1,2,7,8,10)','(1,2,6,7,8,10)');
title(lgd,'Leader set')
```