university of groningen

faculty of science and engineering

Symmetries and Clustering in Nonlinear Network Systems

Master Project Mathematics

January 2021

Student: J. Norden

First supervisor: Dr. ir. B. Besselink

Second supervisor: Dr. A. E. Sterk

Abstract

This thesis treats network systems of diffusively coupled nonlinear oscillators and in particular, model order reduction by means of node clustering. The question whether there are any natural choices of nodes that may be clustered together such that the reduced-order model has similar dynamical features as the original one is addressed for an example network. Results on passivity-based cluster synchronization theory are reviewed. Based on this and by exploiting symmetries present in the network, a criterion is formulated which identifies groups of nodes that can be considered natural candidates for clustering. In the case of a modular directed ring network, it is demonstrated that this criterion predicts preservation of the emergence of a modular travelling wave when comparing the original and reduced-order systems. In the case that the modular travelling wave is preserved, it is demonstrated that its local stability region with respect to the system parameters is reproduced almost exactly. The results support the hypothesis that emergent dynamical features are related to the symmetries present in the network and that this can be exploited for purposes of model order reduction by node clustering.

Contents

Chapter 1

Introduction

With the technological revolution that had its roots in the early 20th century came the rise of network systems: power grids, telephone networks and, not least, the Internet are examples of systems in which individual agents communicate with one another. But not only human-made systems can be understood in terms of their network structure. Many natural systems, such as systems of neurons or gene regulatory mechanisms in biological cells, can be modelled as network systems. While the importance of network structure has been recognized for a long time, only the advancements of the past decades have brought the necessary computational capacities to describe, study and control the behaviour of complex network systems.

In particular, the phenomenon of *synchronization* in network systems has gained substantial attention. Be it networks of pacemaker cells of the heart (see Mirollo & Strogatz [16]), power grids (see Motter et al. [18]) or robotic systems (see Nair & Leonard [20]), the study of synchronization has proved invaluable for applications. A particular variant of synchronization is referred to as *cluster synchronization*. While (full) synchronization is characterized by uniform behaviour of *all* subsystems, cluster synchronization describes the case in which synchronization occurs for clusters of subsystems, rather than for the whole system in its entirety. The mechanisms that lead to the various types of synchronization have been studied by many authors, among them Pogromsky et al. [22, 25, 26], Sorrentino et al. [30] and Aminzare et al. [1].

Since many network systems are very large in size, the modelling of such systems at full scale is often untenable. However, methods of model order reduction have proven to be effective in deriving reduced-order models which still capture the overall behaviour of the original network system (see Antoulas [2] for an overview). One approach to arrive at a reduced-order model is by the clustering of subsystems: a number of subsystems in the original network are clustered together and a new, less complex network is derived from it. Along these lines, significant progress has been made in describing input-output relations for network systems with linear dynamics (see Monshizadeh et al. [17]). In linear network systems, a typical phenomenon of interest is consensus dynamics, i.e. the convergence of all trajectories of the subsystems to a common equilibrium state. This phenomenon can be regarded as a special kind of synchronization. Results of clustering-based model order reduction have been applied successfully to preserve consensus dynamics for linear systems. Much less is known about nonlinear systems and in particular the relationship between network topology and emergent dynamical phenomena such as the (dis)appearances of "non-equilibrium" attractors in the system.

In this thesis, we will study nonlinear network systems and address the question whether there is a way of identifying node clusters that, upon clustering-based model order reduction, yield reduced-order models which are good approximations of the original network. How good or bad a reduced-order model is will be evaluated on the basis of how well the bifurcation structure is preserved. In its full generality, the problem of judging how well the bifurcation structure is preserved under model order reduction is extremely complicated. Therefore, the scope of this thesis is restricted to the exploration of this problem for two example networks of nonlinear oscillators in which the subsystems are modelled as FitzHugh-Nagumo neurons. The networks under consideration are a cube network of 8 subsystems and a modular ring network of variable size. The networks have been selected due to their simplicity and limited number of symmetries. Further, the modular ring network is a straightforward modification of a simple ring which can be regarded as a building block for more complicated network structures. The modular ring network features multi-stability of the fully synchronous solution together with a modular travelling wave solution. We study the preservation of the multi-stability region in parameter space under clustering-based model order reduction. The symmetries of the example networks are exploited to apply arguments from cluster synchronization theory. The intuition behind the approach is the following: if nodes in a certain cluster tend to synchronize, then representing this cluster by a single node in the reduced-order network should result in a good approximation of the original network. If nodes tend to synchronize in multiple clusters, we would like to reduce those clusters in which the nodes synchronize fastest. Aiming to formalize this intuition, upper bounds on the convergence rates of state trajectories associated with the different choices of clustering nodes are derived and compared. For the example networks, it is demonstrated that the spectral properties of a certain matrix associated with the network structure are related to these convergence rates. This leads to a rough classification of node clusterings according to how well-suited they are for purposes of clustering-based model order reduction.

The contents of this thesis are arranged as follows:

- Chapter 2 *Modelling* aims to introduce the problem in a more mathematically rigorous way. Relevant notions from graph theory will be reviewed and a precise characterization of network systems with diffusive coupling will be given. The FitzHugh-Nagumo (FHN) model for neuronal activity will be discussed briefly and two networks consisting of FHN oscillators with different topologies will be introduced. These two network system will serve as running examples throughout the thesis. Further, the notion of clustering-based model order reduction will be introduced in a precise manner. The problem statement will be reformulated in terms of the introduced concepts.

- In chapter 3 *Cluster Synchronization*, we will review results pertaining to cluster synchronization in network systems as described by Sorrentino et al. [30] and Pogromsky [22, 23]. To do so, we review the notion of symmetry of a graph and we will see how these symmetries give rise to invariant manifolds of the underlying network system. These invariant manifolds in turn are associated with cluster synchronization. Employing the notion of passivity, Pogromsky's argumentation results in a stability criterion that can be used to assess whether a network system displays cluster synchronization. From Pogromsky's more

general result, a variant which holds true for network systems of FHN oscillators is derived. This variant also allows for the computation of upper bounds on the convergence rates associated with convergence towards the mentioned invariant manifolds.

- As the name suggests, Chapter 4 *Application to Example Networks* deals with the application of the theory presented in chapter 3 to the example networks as introduced in chapter 2. The conditions that allow for the application of the results from chapter 3 will be verified explicitly. It is then demonstrated that in the case of the example networks, the theory on cluster synchronization can be used to predict which groups of nodes in a network are good candidates for clustering, or at least which groups of nodes are bad candidates.

- Finally, in chapter 5 *Conclusion and Discussion*, we recapitulate the contents of the thesis and conclude. Directions for further research will be outlined.

This thesis is largely self-contained, albeit with the occasional reference to one of the appendices which contain supplementary material. However, the reader is expected to be familiar with basic group theory, linear algebra and the theory of ordinary differential equations.

Chapter 2

Modelling

The aim of this chapter is to introduce all mathematical concepts needed to study clustering-based model order reduction for nonlinear network systems with diffusive coupling. In particular, this chapter aims to:

1. Review some basic graph theory and make precise the notion of network systems with diffusive coupling.
2. Introduce Steur's network and Pogromsky's network of FitzHugh-Nagumo oscillators. These example networks will be the focus of our investigations.
3. Familiarize the reader with the notion of clustering-based model order reduction.
4. Formulate a precise problem statement.

Let us begin with an exposition of basic graph theory.

2.1 Preliminaries from Graph Theory

We will consider both weighted undirected graphs as well as weighted directed graphs. A *weighted undirected graph* is an ordered triple $G = (V, E, A)$. The set V is referred to as the vertex set (or node set) and we usually consider $V = \{1, 2, \ldots, N\}$. The edge set E is a set of *unordered pairs* $\{i, j\}$ with $i \neq j$ and indicates which vertices are connected to one another. We say that nodes i and j are neighbors if $\{i, j\} \in E$. Finally, the adjacency matrix $A = [a_{ij}]$ has non-negative elements and the element a_{ij} indicates the weight of the connection between vertex i and j. Note that $a_{ij} = a_{ji}$ and $a_{ij} = 0$ if there is no edge between vertex i and j, or equivalently

$$a_{ij} > 0 \iff \{i, j\} \in E.$$

A *weighted directed graph* is an ordered triple $G = (V, E, A)$. To contrast against the case of undirected graphs, for directed graphs, we refer to E as the arc set. The arc set contains *ordered pairs* (i, j) with $i \neq j$. If $(i, j) \in E$ then we call node i an in-neighbour of node j. Likewise, we call node j an out-neighbour of node i. Given an arc $(i, j) \in E$ we say that i is the tail and j is the head of the arc. The adjacency matrix $A = [a_{ij}]$ is again similarly defined as before,

however, now a_{ij} is the weight of the arc from vertex j to i. In general $a_{ij} \neq a_{ji}$ and $a_{ij} = 0$, if there is no arc from j to i, or equivalently

$$a_{ji} > 0 \iff (i,j) \in E.$$

Remarks.

i) We consider *simple* graphs, that is graphs without multiple edges/arcs between vertices and no self-connections. Moreover, since we will be dealing with weighted graphs only, we omit the word "weighted" and simply refer to undirected and directed graphs.

ii) We call a directed graph symmetric whenever $(i,j) \in E \iff (j,i) \in E$. Note that in this case a_{ij} and a_{ji} may be distinct.

For both undirected and directed graphs, we define the *degree matrix* $D = \text{diag}(d_1, \ldots, d_N)$ with

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

In the case of an unweighted graph, or equivalently a weighted graph with all weights equal to one, the quantity d_i is simply the number of neighbours of vertex i. In general, the number d_i conflates information about the number of neighbours of vertex i and the associated edge weights. In the case of a directed graph, the matrix D is also referred to as the *in-degree matrix* as only arcs which have vertex i as their head contribute to d_i. We define the *Laplacian matrix*

$$\mathcal{L} = D - A.$$

As we will soon see, the Laplacian matrix is a very useful mathematical object in the description of network systems.

Remarks.

i) Any undirected graph can be associated with a symmetric directed graph with $a_{ij} = a_{ji}$.

ii) For undirected graphs, the Laplacian matrix is always symmetric and positive semi-definite, as can be concluded from Gershgorin's circle theorem.

iii) If a matrix M is positive semi-definite, we will denote this as $M \succcurlyeq 0$. Similar notation is used if M is negative semi-definite ($M \preccurlyeq 0$), positive definite ($M \succ 0$) or negative definite ($M \prec 0$).

We are now equipped with the basic graph theory necessary to start our investigation of diffusively coupled networks. For further reading on graph theory, see for example Bollobás [4] and Egerstedt & Mesbahi [15]. As we move along, we will eventually make use of additional concepts from graph theory but the introduction of more advanced machinery will be postponed until later sections.

2.2 Diffusively Coupled Networks

In 1976, Smale [29] studied a model of concentrations of chemicals in two identical cells which are separated by a membrane and interact via diffusion. While concentrations in each individual cell by itself would approach an equilibrium, introducing a diffusive interaction between the cells allowed the coupled system to exhibit more complicated phenomena such as oscillatory motion. Inspired by these observations, networks in which the individual systems are coupled via diffusion-like terms, e.g. based on output differences, have been studied extensively. Let us introduce the notion of a *diffusively coupled network* more formally.

We consider systems of the form

$$\begin{cases} \dot{x}_j = f(x_j) + Bu_j, \\ y_j = Cx_j, \end{cases} \tag{2.1}$$

with $j = 1, 2, \ldots, N$, where $x_j(t) \in \mathbb{R}^n$ is the state, $u_j(t) \in \mathbb{R}^m$ is the input and $y_j(t) \in \mathbb{R}^m$ is the output of the j-th system. We require that $f(0) = 0$ and B, C are constant matrices of appropriate dimension. Further, we assume that the matrix CB is positive definite. The N subsystems are interconnected via some underlying graph $G = (V, E, A)$. We think of individual subsystems as nodes in the graph G and the interaction between two subsystems is represented by the edges/arcs between the corresponding nodes in G. This interconnection structure is reflected in the feedback terms

$$u_j = -\sum_{i=1}^{N} a_{ji}(y_j - y_i), \tag{2.2}$$

where a_{ij} are the elements of the adjacency matrix associated with G. Note how subsystem j interacts with the other subsystems solely based on *output differences*. For this reason, we call this kind of coupling *diffusive*. Exploiting the fact that we are dealing with identical subsystems, we can simplify equations (2.1) and (2.2) considerably by introducing more convenient notation: recall that the Laplacian matrix associated with G was defined as

$$\mathcal{L} = D - A = \begin{bmatrix} \sum_{i=1}^{N} a_{1i} & -a_{12} & \ldots & -a_{1N} \\ -a_{21} & \sum_{i=1}^{N} a_{2i} & \ldots & -a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{N1} & -a_{N2} & \ldots & \sum_{i=1}^{N} a_{Ni} \end{bmatrix}.$$

We now define

$$\begin{aligned} x &= \mathrm{col}(x_1, \ldots, x_N) \in \mathbb{R}^{Nn}, \\ y &= \mathrm{col}(y_1, \ldots, y_N) \in \mathbb{R}^{Nm}, \\ u &= \mathrm{col}(u_1, \ldots, u_N) \in \mathbb{R}^{Nm}, \\ F(x) &= \mathrm{col}(f(x_1), \ldots, f(x_N)) \in \mathbb{R}^{Nn}. \end{aligned}$$

where $\mathrm{col}(\cdot, \ldots, \cdot)$ denotes the "vertical stacking" of vectors. Our original system (2.1), together with the feedback defined in (2.2), can now be written as

$$\begin{cases} \dot{x} = F(x) + (I_N \otimes B)u, \\ y = (I_N \otimes C)x, \end{cases} \tag{2.3}$$

with feedback

$$u = -(\mathcal{L} \otimes I_m)y.$$

Remark. Here we have used the Kronecker product (denoted by $\otimes$) to enable compact notation. Given some $p \times q$ matrix Q and $r \times s$ matrix R, their Kronecker product is the $pr \times qs$ matrix defined by

$$Q \otimes R = \begin{bmatrix} Q_{11}R & Q_{12}R & \ldots & Q_{1q}R \\ Q_{21}R & Q_{22}R & \ldots & Q_{2q}R \\ \vdots & \vdots & \ddots & \vdots \\ Q_{p1}R & Q_{p2}R & \ldots & Q_{pq}R \end{bmatrix}.$$

With this notation, the closed-loop system now takes the form

$$\dot{x} = F(x) - (\mathcal{L} \otimes BC)x. \tag{2.4}$$

Note how in equation (2.4), the two terms nicely separate the contribution coming from the dynamics of the individual subsystems and the contribution coming from the network structure.

2.3 Introduction to Example Networks

Throughout this thesis, we will keep coming back to two example networks. On the basis of observations about these examples, we will try to extract more general principles. As already pointed out, diffusively coupled networks with *identical* subsystems will be considered. Since we are interested in networks of artificial neurons, each subsystem will be a so-called FitzHugh-Nagumo oscillator; a common choice in modelling neuronal activity in recent literature.

2.3.1 The FitzHugh-Nagumo Oscillator

The FitzHugh-Nagumo oscillator (short: FHN oscillator) is a two-dimensional simplification of the famous Hodgkin-Huxley model for action potentials in neurons. The model is named after Richard FitzHugh [7], who first proposed the model in 1961, and Jin-Ichi Nagumo [19] who built a corresponding electrical circuit in 1962. The FHN oscillator has recently gained popularity

in the study of networks of nonlinear systems due to its simplicity: the qualitative behaviour of solutions can largely be described by studying the z and y nullclines. The FHN oscillator is given by the following set of coupled ordinary differential equations

$$\begin{cases} \dot{z} = \alpha(y - \beta z), \\ \dot{y} = y - \gamma y^3 - z + u, \end{cases} \tag{2.5}$$

where y represents the membrane potential of the neuron, z a recovery variable and u an external stimulus. A common choice for the parameters is

$$\alpha = \frac{8}{100}, \quad \beta = \frac{8}{10}, \quad \gamma = \frac{1}{3}.$$

Let us have a first look at what solutions of the FHN oscillator look like. The typical behaviour of a single FHN oscillator without any external stimulation ($u \equiv 0$) is summarized in Figure 2.1.

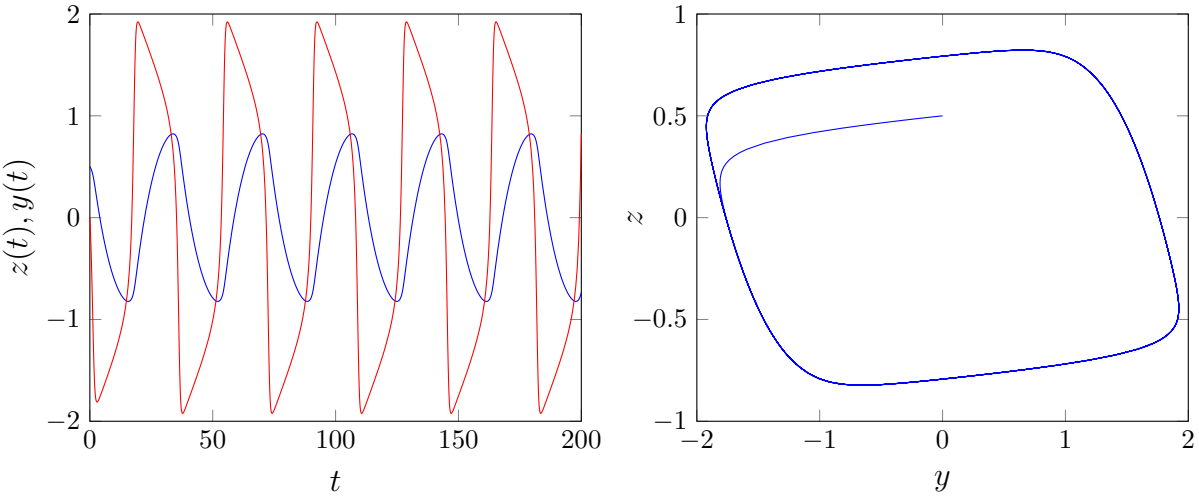

Figure 2.1: Left panel: Solutions $z(t)$ (blue) and $y(t)$ (red) of a single FHN oscillator as time series. Right panel: Solutions $z(t)$ and $y(t)$ plotted together in state space. In both plots the initial condition is $(z(0), y(0)) = (0.5, 0)$ and the external stimulus is $u \equiv 0$.

As long as the initial condition is chosen to be different from the equilibrium at the origin, we will observe oscillatory behaviour as in Figure 2.1. There is a lot to be said about the dynamics of a single FHN oscillator when considering external stimuli and how this represents the excitation of a physical neuron. However, this is not in the scope of this thesis and the interested reader is referred to Rocşoreanu et al. [28] for a comprehensive review and the original papers by FitzHugh [7] and Nagumo [19].

Let us now see what the equations for a network of such oscillators look like. As derived in section 2.2, the equation for a diffusively coupled network takes the form

$$\dot{x} = F(x) - (\mathcal{L} \otimes BC)x.$$

What are $F(\cdot), B$ and C in the case of FHN oscillators? Letting $x_j = \mathrm{col}(z_j, y_j)$ we see that

equations (2.5) can be written as

$$\begin{cases} \dot{x}_j = Ax_j - \gamma y_j^3 B + Bu_j, \\ y_j = Cx_j, \end{cases}$$

with

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

This means that

$$f(x_j) = Ax_j - \gamma y_j^3 B,$$

and so with $x = \mathrm{col}(x_1, \ldots, x_N)$ we have

$$F(x) = (I_N \otimes A)x - \gamma(I_N \otimes B)\Phi((I_N \otimes C)x),$$

where $\Phi : \mathbb{R}^N \to \mathbb{R}^N$ is defined by component-wise cubing, i.e.

$$\Phi(y) = \begin{bmatrix} y_1^3 & y_2^3 & \ldots & y_N^3 \end{bmatrix}^\top.$$

It follows that a network of diffusively coupled FHN oscillators is described by the equation

$$\begin{aligned} \dot{x} &= F(x) - (\mathcal{L} \otimes BC)x \\ &= (I_N \otimes A)x - \gamma(I_N \otimes B)\Phi((I_N \otimes C)x) - (\mathcal{L} \otimes BC)x. \end{aligned}$$

Having chosen the FHN oscillator as the basic element of our nonlinear network, the function F together with matrices B and C are now determined. The only remaining unknown is the Laplacian matrix $\mathcal{L}$ of the underlying network structure. We proceed with an exposition of two example networks that we will be analyzing throughout this thesis.

2.3.2 Steur's Network

In their research, Steur et al. [8, 9, 31] study networks of coupled nonlinear oscillators. Among other things, they investigate the influence of the network topology on certain dynamical features. In particular, they study two very basic network types: The directed chain and the directed cycle. These are of special interest as they can be thought of as basic building blocks for more complicated network structures. Generalizing from directed chains and cycles, Steur and colleagues then consider networks with so-called modular structure.

A module of size $\bar{M}$ is a symmetric directed graph with equal arc weights, i.e. for any two nodes i and j in the module, we have $(i,j),(j,i) \in E$ with $a_{ij} = a_{ji}$. So the de facto structure of a module is that of a complete graph of $\bar{M}$ nodes. A modular network comprises a number of modules of potentially different sizes. In each module, only a single node is connected to nodes from other modules but apart from that, the inter-modular coupling structure is not subject to further restrictions. We will now introduce a particular modular network for which the inter-modular coupling structure is that of a directed ring. This network will henceforth be referred to as *Steur's network*.

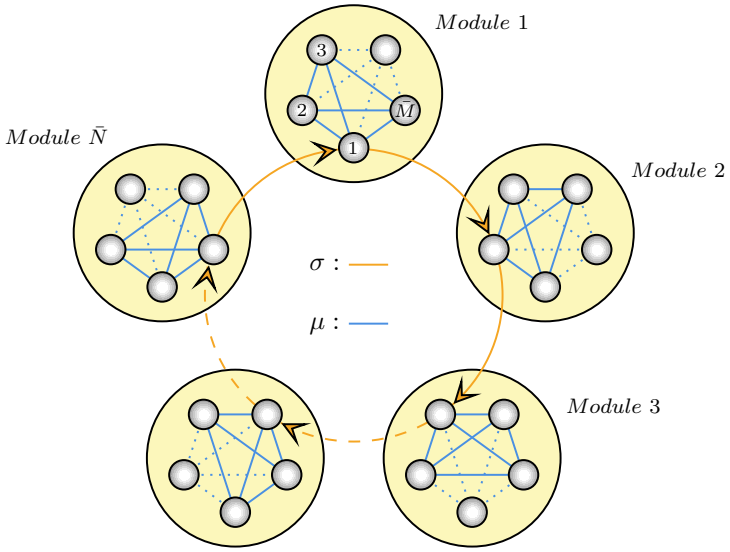

Figure 2.2: Steur's network. For purposes of visual clarity, the arcs within the modules are illustrated as undirected edges.

Let us derive the graph Laplacian of Steur's network. As is depicted in Figure 2.2, Steur's network is a modular network with $N = \bar{N}\bar{M}$ nodes consisting of $\bar{N}$ modules where each module contains $\bar{M}$ nodes. The inter-module coupling structure is that of a directed cycle where only the first node in each module is connected to the next/previous module as depicted. The graph Laplacian $\mathcal{L}_{\text{module}}$ for a single module is given by

$$\mathcal{L}_{\text{module}} = \begin{bmatrix} \bar{M}-1 & -1 & \dots & -1 \\ -1 & \bar{M}-1 & \dots & -1 \\ \vdots & \ddots & \ddots & \vdots \\ -1 & -1 & \dots & \bar{M}-1 \end{bmatrix} \in \mathbb{R}^{\bar{M}\times\bar{M}},$$

while the graph Laplacian $\mathcal{L}_{\text{ring}}$ for the directed ring structure is given by

$$\mathcal{L}_{\text{ring}} = \begin{bmatrix} 1 & 0 & 0 & \dots & -1 \\ -1 & 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & -1 & 1 & 0 \\ 0 & 0 & \dots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{\bar{N}\times\bar{N}}.$$

Note that the matrix $\mathcal{L}_{\text{module}}$ is symmetric while $\mathcal{L}_{\text{ring}}$ is a circulant matrix. In order to combine these Laplacians and arrive at the Laplacian of Steur's network, we introduce the matrix

$$B_{\mathcal{L}} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix},$$

with $B_{\mathcal{L}} \in \mathbb{R}^{\bar{M}\times\bar{M}}$ and again make use of the Kronecker product. It is readily checked that the

graph Laplacian of Steur's network is given by

$$\mathcal{L}_{\text{ring}} \otimes B_{\mathcal{L}} + I_N \otimes \mathcal{L}_{\text{module}},$$

where $I_{\bar{N}}$ is the $\bar{N} \times \bar{N}$ identity matrix. We go one step further and introduce an *intra-module coupling strength* μ and an *inter-module coupling strength* σ. Depending on these two parameters, the Laplacian of Steur's network becomes a function of σ and μ, i.e.

$$\mathcal{L}_S(\sigma, \mu) = \sigma \mathcal{L}_{\text{ring}} \otimes B_{\mathcal{L}} + \mu I_N \otimes \mathcal{L}_{\text{module}}.$$

To understand the influence of σ and μ, let us consider two extreme cases: In the case of $\sigma = 0$ the Laplacian $\mathcal{L}_S$ is simply that of $\bar{N}$ disconnected modules, each of which is essentially a complete graph. On the other hand, if $\mu = 0$, then $\mathcal{L}_S$ represents a simple directed ring network together with $(\bar{M} - 1)\bar{N}$ disconnected subsystems. Varying both σ and μ continuously means to consider a network whose characteristics lie somewhere between these two extremes.

Having found the Laplacian, the dynamics of Steur's network of FHN oscillators is then described by the equation

$$\dot{x} = F(x) - (\mathcal{L}_S(\sigma, \mu) \otimes BC)x. \tag{2.6}$$

Let us take a moment to have a look at what solutions to the equation (2.6) might look like. We consider Steur's network with $\bar{M} = 5$ nodes per module and $\bar{N} = 5$ modules in total. This

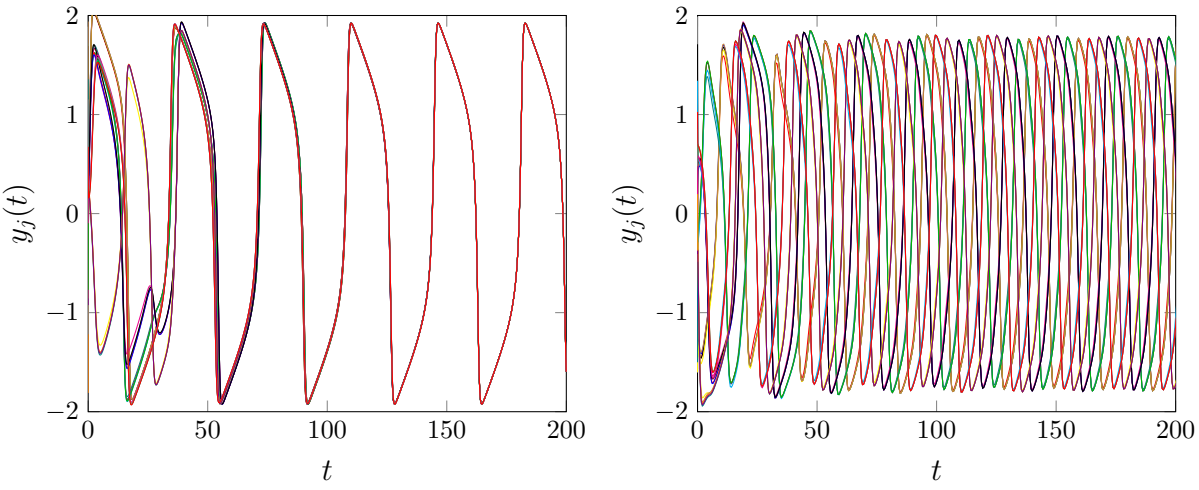

Figure 2.3: Time series for Steur's network with $\bar{M} = \bar{N} = 5$ and $\sigma = \mu = 1$. Left panel: Full synchronization of oscillators in Steur's network. Starting off from distinct initial conditions, the oscillators soon fully synchronize. Right panel: Cluster synchronization of oscillators in Steur's network. After an initial transient, some nodes synchronize while others do not.

yields a network with a total of $N = 25$ coupled oscillators, each having two components, y_j and z_j. If we wanted to understand the motion in the 50-dimensional state space, we might run into some trouble visualizing this. As an alternative, we will simply plot the solutions of $y_1, \ldots, y_{25}$ as functions of time. The results are summarized in Figure 2.3, Figure 2.4 and Figure 2.5.

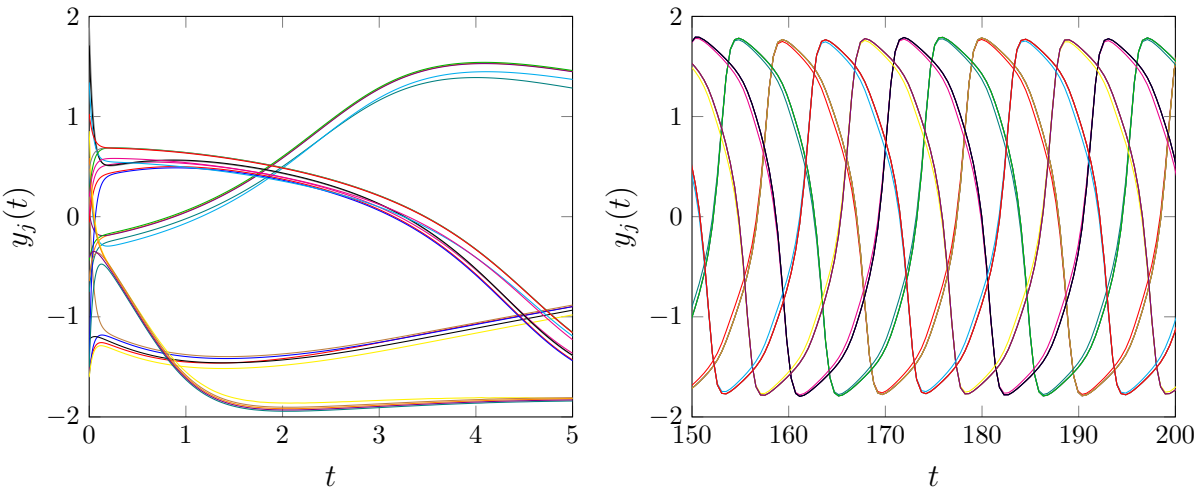

Figure 2.4: Zoomed version of the solution displaying cluster synchronization as illustrated in the right panel of Figure 2.3. Left panel: The membrane potentials of oscillators in each module quickly converge on one another. Right panel: After the initial transit, the solution starts to exhibit a periodic character.

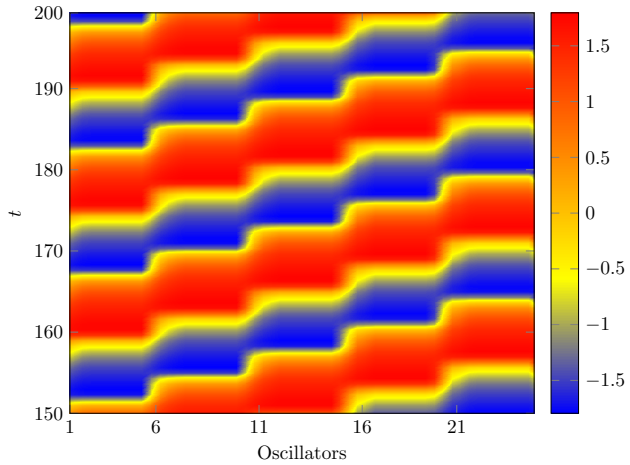

Figure 2.5: Hovmöller diagram of the solution displayed in the right panel of Figure 2.3. The membrane potentials $y_j(t)$ of the oscillators are indicated by colour. Oscillators in the modules of Steur's network synchronize and a modular travelling wave emerges.

Overall, we find two distinct behaviours of solutions: full synchronization and a solution which displays cluster synchronization. As the name suggest, in the full synchronization regime, all oscillators synchronize after some time as illustrated in the left panel of Figure 2.3. For the solution characterized by cluster synchronization, the Hovmöller diagram in Figure 2.5 reveals that this solution is in fact a modular travelling wave. The modular travelling wave solution is characterised by cluster synchronization of the oscillators in each module. However, the individual modules then do not synchronize but rather converge to a solution where each module has the same period with a phase shift. Figure 2.4 shows the same modular travelling wave as the right panel of Figure 2.3, plotted for different time windows. Both solution types (full synchronization and modular travelling wave) correspond to periodic solutions in state space. Which behaviour is to be observed depends on the values of the parameters σ and μ, as well as the chosen initial condition. As was shown in earlier work by Raven [27], there is a region in the parameter plane for which both full synchronization and modular travelling waves can occur

and another region in which only the full synchronization behaviour exists. We will return to this later on.

In conclusion, there are a number of reasons that make Steur's network an interesting object of study: its modular structure leads us to natural choices of nodes to be clustered together for model order reduction as we will soon see. Even though the network has a fairly simple structure, complex behaviour such as bi-stability of both full synchronization and modular travelling wave solutions have been observed in previous work. For now, let us move on and introduce one more example network.

2.3.3 Pogromsky's Network

The next example network has also been studied by Pogromsky [22] and we will therefore refer to it as *Pogromsky's network*. The network has a cube structure, as is illustrated in Figure 2.6.

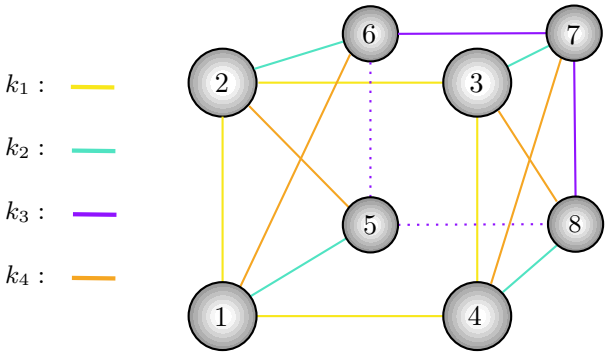

Figure 2.6: Pogromsky's network.

Pogromsky's network is undirected and there are 4 coupling parameters, k_1, k_2, k_3 and k_4, which govern the, very symmetric, coupling relations in this network. Denoting

$$k = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix}^\top,$$

the graph Laplacian is given by

$$\mathcal{L}_P(k) = \begin{bmatrix} \Delta_1 & -k_1 & 0 & -k_1 & -k_2 & -k_4 & 0 & 0 \\ -k_1 & \Delta_1 & -k_1 & 0 & -k_4 & -k_2 & 0 & 0 \\ 0 & -k_1 & \Delta_1 & -k_1 & 0 & 0 & -k_2 & -k_4 \\ -k_1 & 0 & -k_1 & \Delta_1 & 0 & 0 & -k_4 & -k_2 \\ -k_2 & -k_4 & 0 & 0 & \Delta_2 & -k_3 & 0 & -k_3 \\ -k_4 & -k_2 & 0 & 0 & -k_3 & \Delta_2 & -k_3 & 0 \\ 0 & 0 & -k_2 & -k_4 & 0 & -k_3 & \Delta_2 & -k_3 \\ 0 & 0 & -k_4 & -k_2 & -k_3 & 0 & -k_3 & \Delta_2 \end{bmatrix}, \tag{2.7}$$

with

$$\Delta_1 = 2k_1 + k_2 + k_4,$$
$$\Delta_2 = 2k_2 + k_3 + k_4.$$

Pogromky's network of FHN oscillators is then governed by

$$\dot{x} = F(x) - (\mathcal{L}_P(k) \otimes BC)x. \tag{2.8}$$

We will again have a brief look at what typical solutions of this network system might be. As before, rather than trying to make sense of motion in 16 dimensions, we will simply plot the time series $y_j(t)$ of the membrane potential for each oscillator in the network, given some initial condition. From various trials with randomized parameter values and randomized initial

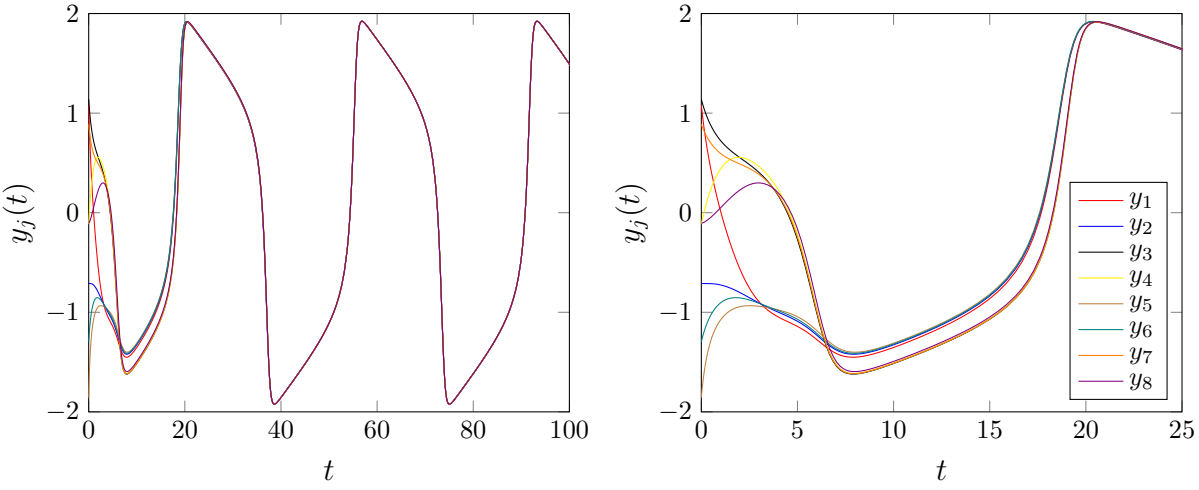

Figure 2.7: Left panel: Solution of Pogromsky's network with $k_1 = k_2 = k_3 = k_4 = 0.3$. The oscillators fully synchronize eventually. Right panel: Zoomed version of the solution in the left panel. Cluster synchronization sets in before the oscillators fully synchronize.

conditions, it appears that the situation as presented in Figure 2.7 is the most generic, if not the only one: in a short transient period, groups of oscillators first partially synchronize until, eventually, full synchronization sets in. As can be seen in the right panel of Figure 2.7, in this particular case, oscillators 1, 2, 5 and 6 synchronize early. The same holds for oscillators 3, 4, 7 and 8. Only after a while, these two groups of four oscillators each, then synchronize and the overall system attains full synchronization. Recall that synchronization of groups of oscillators has also been observed in Steur's network.

Pogromsky's network has been chosen as a second example for two reasons: Firstly, the network has an overseeable amount of symmetry which is conveniently regulated by the four system parameters. Secondly, the small number of nodes allows for numerical simulations with short running times. Having introduced the two example networks and being more familiar with the type of solutions they generate, we turn to the question of how one could approximate the behaviour of network systems.

2.4 Model Order Reduction Techniques

To begin with, we should answer the following more general questions:

1. What is model order reduction?
2. Why are we interested in this?

As alluded to in the introduction, complex and large networks are not only the products of technological advancements, examples are power grids, computer chips and the Internet, but life itself also tends to generate them, think of physical neural networks or gene regulatory networks. The study of network systems has many important applications and the networks involved tend to be massive in size. Very large networks become difficult to analyze simply because of their size. Consider, for example, Pogromsky's network with 8 model neurons. We can give a nice visual representation of this network and our attention and memory capacity is sufficient to keep all the nodes and connections of the network in mind at once. Numerically solving the associated equations can take up to a few seconds with a standard home computer but this is still sufficiently fast for us to get an idea of what the solutions might look like. If we consider the adult human brain with roughly *86 billion neurons* (see Azevedo et al. [3]), it is utterly hopeless to try and make intuitive sense of such a network, not to speak of running any full-scale simulations. To be able to analyze large networks nonetheless, we require model order reduction techniques.

Model order reduction is the process of obtaining a simpler model from a more complicated one. Ideally, the reduced model then still captures characteristics of the object to be modelled while being significantly simpler. This usually results in a trade off between model complexity and the degree to which the object is faithfully represented. Since we are interested in network systems, the model complexity is simply the number of nodes in the network. The notion of "faithful representation" is a more complicated matter and we will return to the issue at the end of this chapter. Having outlined what model order reduction is and why we need it, let us now return to the study of Steur's and Pogromsky's network.

In the introduction to the example networks it has become clear that certain oscillators in the networks tend to synchronize while others do not. The phenomenon of synchronization depends on the network structure and in particular also on the coupling strengths. We will see in chapter 3 that these different modes of synchronization can be linked to the symmetries which are present in the graph of the underlying network. For now, let us explore how these observations could relate to model order reduction: For diffusively coupled networks, in which certain subsystems tend to synchronize, we could argue that these subsystems have a "similar function" in the context of the full network. After an initial transit, solutions of the full network become restrained to a lower dimensional subspace as the individual subsystems synchronize. Clustering-based model order reduction aims to exploit this behaviour: instead of modelling the full network, we "cluster together" subsets of nodes, i.e. we consider a network with fewer nodes in which each node represents a group of nodes from the original network. We then use solutions of the smaller network to approximate solutions of the full network. This immediately raises two questions:

1. How can we obtain a reduced network from the original one?

2. Given the reduced network, what are the governing equations?

To answer these questions, we review the basic ideas of projection-based model order reduction with clustering-based model order reduction as a special case.

2.4.1 Projection-Based Model Order Reduction

We consider the general case of a system with state and output equations

$$\begin{cases} \dot{x} = g(x,t), \\ y = h(x,t), \end{cases} \tag{2.9}$$

where $x \in \mathbb{R}^N, y \in \mathbb{R}^m$ and $x(0) = x_0$. If such a system has a high-dimensional state space (i.e. N is large), it is often the case that, as time goes to infinity, its solutions tend towards trajectories which are restricted to some lower-dimensional subspace $\mathcal{S}$ with $\dim(\mathcal{S}) = k_{\mathcal{S}} < N$. After a sufficiently long transient time, we can then approximate

$$x(t) \approx \xi_1(t)v_1 + \ldots + \xi_{k_{\mathcal{S}}}(t)v_{k_{\mathcal{S}}},$$

where the vectors v_i form a basis for $\mathcal{S}$ and the functions $\xi_i : \mathbb{R} \to \mathbb{R}$ describe the component wise time evolution. Introducing $\xi(t) = \mathrm{col}(\xi_1(t), \ldots, \xi_{k_{\mathcal{S}}}(t))$ and

$$V_{\mathcal{S}} = \begin{bmatrix} v_1 & \ldots & v_{k_{\mathcal{S}}} \end{bmatrix},$$

we can write

$$x(t) \approx V_{\mathcal{S}}\xi(t).$$

However, in general, the basis for $\mathcal{S}$, represented by the matrix $V_{\mathcal{S}}$, is unknown as well and so we need to approximate it with some matrix $V \in \mathbb{R}^{N\times k}$ where k is potentially different from $k_{\mathcal{S}}$. A candidate for V can often be arrived at with an empirical approach. We then approximate

$$x(t) \approx V\xi(t). \tag{2.10}$$

Substitution of approximation (2.10) into equations (2.9) yields

$$\begin{cases} V\dot{\xi} = g(V\xi(t),t) + r(t), \\ y(t) = h(V\xi(t),t), \end{cases} \tag{2.11}$$

where $r(t)$ is a residual term accounting for the approximation. We now try to recover an expression for the dynamics of ξ which does not involve the residual term $r(t)$: left-multiplying the state equation in system (2.11) by the transpose of an arbitrary matrix $W \in \mathbb{R}^{N\times k}$ yields

$$W^\top V\dot{\xi} = W^\top g(V\xi(t),t) + W^\top r(t).$$

If we now solve

$$W^\top V\dot{\xi} = W^\top g(V\xi(t),t), \tag{2.12}$$

we enforce that $\xi(t)$ and $r(t)$ are such that

$$W^\top r(t) = 0, \quad \forall t.$$

In the case that $W^\top V$ is invertible, we can express the equations of the reduced system explicitly as

$$\begin{cases} \dot{\xi} = (W^\top V)^{-1} W^\top g(V\xi(t), t), \\ y(t) = h(V\xi(t), t). \end{cases} \tag{2.13}$$

For $V = W$, this method is referred to as *Galerkin projection* and in the general case $V \neq W$, it is called *Petrov-Galerkin projection*. Knowing the general framework for projection-based model order reduction techniques, we now consider the special case of clustering-based model order reduction for network systems. To do so, we need to introduce a few more graph theoretic tools.

2.4.2 Graph Partitions

In order to introduce clustering-based model order reduction, we review the notion of graph partitions (see chapter 2.5 in Egerstedt & Mesbahi [15]).

Definition 2.4.1. Given some undirected or directed graph G with vertex set $V = \{1, 2, \ldots, N\}$, we call a nonempty subset $\mathcal{C} \subseteq V$ a *cell* of V. A collection of cells $\pi = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ is called a *partition* if

$$\bigcup_{i=1}^{K} \mathcal{C}_i = V \quad \text{and} \quad \mathcal{C}_i \cap \mathcal{C}_j = \emptyset \text{ for } i \neq j.$$

With a slight abuse of terminology, we will say that π is a "partition of G" and mean that it is a partition of the associated vertex set.

Definition 2.4.2. Given a cell $\mathcal{C}$ belonging to some partition π, we define the *characteristic vector* $p(C) \in \mathbb{R}^N$ by

$$(p(C))_i = \begin{cases} 1, & \text{if } i \in \mathcal{C}, \\ 0, & \text{otherwise.} \end{cases}$$

Given a partition $\pi = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$, the *characteristic matrix* $P(\pi)$ is then defined as

$$P(\pi) = \begin{bmatrix} p(\mathcal{C}_1) & p(\mathcal{C}_2) & \ldots & p(\mathcal{C}_K) \end{bmatrix}.$$

Example 1. Consider the graph of Pogromsky's network as introduced in section 2.3. The set

$$\pi = \{\{1, 3\}, \{2, 4\}, \{5, 6, 7, 8\}\}$$

is a partition of this graph and the associated characteristic matrix is

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

We now have a framework that allows us to talk about node groupings of a graph in a very general and abstract way. The next step is to combine this with projection-based model order reduction to arrive at clustering-based model order reduction.

2.4.3 Clustering-Based Model Order Reduction

Clustering-based model order reduction for a network system is essentially Petrov-Galerkin projection with a special choice of V and W: Suppose we are given some diffusively coupled network with equations

$$\begin{cases} \dot{x} = g(x(t), t) = F(x) - (\mathcal{L} \otimes BC)x, \\ y = h(x(t), t) = (I_N \otimes C)x \end{cases} \tag{2.14}$$

and G is the graph associated with the Laplacian $\mathcal{L}$. Further, assume we are given some graph partition π of G with K cells where $K \leq N$. This means $P(\pi) \in \mathbb{R}^{N \times K}$. For convenience, we will drop the argument in $P(\pi)$ from now on. To ensure that the reduced system can again be thought of as a network of (now fewer) oscillators, we set

$$\bar{P} = P \otimes I_n$$

and consider a Galerkin projection with

$$V = W = \bar{P}.$$

In the case $K = N$, we have $\bar{P} = I_{Nn}$ and no reduction takes place.

What can we say about the reduced system in the general case? Note that the columns of P are orthogonal and so the matrix $P^\top P$ is diagonal where $[P^\top P]_{ii}$ is the number of nodes in $\mathcal{C}_i$. This means that $P^\top P$ is invertible and hence $\bar{P}^\top \bar{P}$ is invertible as well. Invoking the derivations which culminated in the arrival at equations (2.11), it follows that the reduced system is governed by

$$\begin{cases} \dot{\xi} = (\bar{P}^\top \bar{P})^{-1} \bar{P}^\top (F(P\xi) - (\mathcal{L} \otimes BC)\bar{P}\xi), \\ \zeta = (I_N \otimes C)\bar{P}\xi. \end{cases}$$

A direct computation shows that the reduced state equation takes the form

$$\dot{\xi} = \bar{F}(\xi) - (\bar{\mathcal{L}} \otimes BC)\xi,$$

where $\bar{\mathcal{L}} = (P^\top P)^{-1} P^\top \mathcal{L} P$ and we use that

$$F(\bar{P}\xi) = \bar{P}\bar{F}(\xi),$$

with $\bar{F}(\xi) = \mathrm{col}(f(\xi_1), \ldots, f(\xi_K)) \in \mathbb{R}^{Kn}$ analogously defined to $F(\cdot)$. Finally, we conclude that the reduced system is of the form

$$\begin{cases} \dot{\xi} = \bar{F}(\xi) - (\bar{\mathcal{L}} \otimes BC)\xi, \\ \zeta = (P \otimes C)\xi, \end{cases} \tag{2.15}$$

which is very similar in its form to the original system given by equations (2.14).

Let us pause and take a moment to understand what happened here. Using a partition π, we approximated the state of the full network by $x \approx \bar{P}\xi$. Using the properties of $P(\pi)$, we could explicitly determine the governing equations for the reduced state ξ. As shown, we have

$$\dot{\xi} = \bar{F}(\xi) - (\bar{\mathcal{L}} \otimes BC)\xi$$

and so the equation for the reduced state has precisely the same form as the state of the original network. The differences are that we are now dealing with a lower dimensional analogue $\bar{F}(\cdot)$ of $F(\cdot)$ and the Laplacian matrix transformed according to

$$\bar{\mathcal{L}} = (P^\top P)^{-1} P^\top \mathcal{L} P. \tag{2.16}$$

What is this new Laplacian and how does the corresponding network relate to our original one? By left-multiplication of $\mathcal{L}$ with $P^\top$ and right-multiplication with P, we add together all the connections of nodes in a cell and combine them into a single node. As pointed out, $P^\top P$ contains the cardinalities of the cells on its diagonal and so by left-multiplication with $(P^\top P)^{-1}$ we divide each row by the number of nodes in the associated cell. This means that $\bar{\mathcal{L}}$ is the Laplacian of the network in which each node represents a cell from the original network and the connections between the nodes are the averaged connections between the cells of the original network.

Example 2. We consider Steur's network with $\bar{M} = 5$ and $\bar{N} = 3$ with unspecified σ and μ. The set

$$\pi = \{\{1\}, \{2,3,4,5\}, \{6\}, \{7,8,9,10\}, \{11\}, \{12,13,14,15\}\}$$

is a partition with associated characteristic matrix $P = I_3 \otimes \hat{P}$ where

$$\hat{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix}^\top.$$

Transforming the Laplacian according to equation (2.16) yields

$$\bar{\mathcal{L}}(\sigma, \mu) = \begin{bmatrix} \sigma + 4\mu & -4\mu & 0 & 0 & -\sigma & 0 \\ -\mu & \mu & 0 & 0 & 0 & 0 \\ -\sigma & 0 & \sigma + 4\mu & -4\mu & 0 & 0 \\ 0 & 0 & -\mu & \mu & 0 & 0 \\ 0 & 0 & -\sigma & 0 & \sigma + 4\mu & -4\mu \\ 0 & 0 & 0 & 0 & -\mu & \mu \end{bmatrix}.$$

We see that the row sums are still 0, so this matrix can indeed be interpreted as the Laplacian of a (weighted) directed graph, namely of the graph as depicted on the right in Figure 2.8. After clustering, we are left with a network of similar structure. While the connections associated with σ "stayed the same", all connections associated with μ are now split into two parts: $\mu_1 = \mu$ and $\mu_2 = 4\mu$. Note that the first node has two incoming arcs, one from node 2 with weight $\mu_2 = 4\mu$ and one from node 5 with weight σ. Node 2 of the reduced network represents the

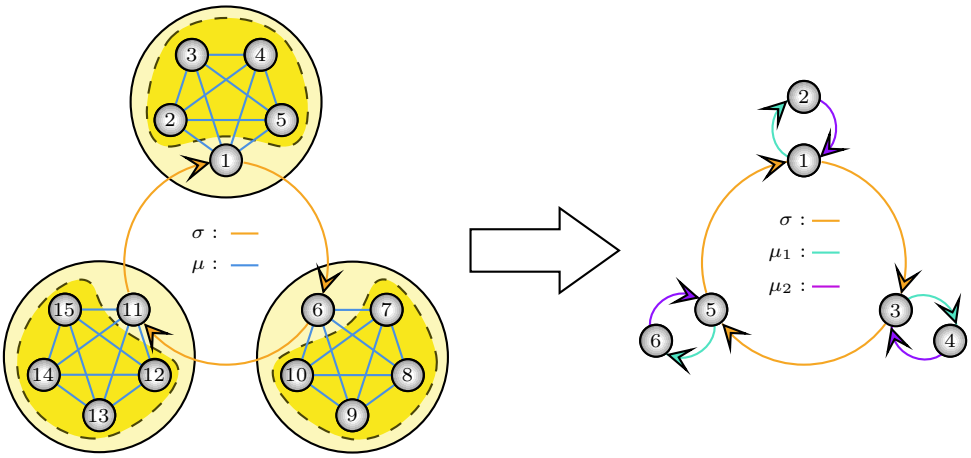

Figure 2.8: Clustering-based model order reduction on Steur's network.

clustered cell $\{2, 3, 4, 5\}$ of the original network and so has an associated weight of 4μ, since the original weights are all μ and there are 4 nodes in this cell. This explains the difference between μ_1 and μ_2. Node 5 of the reduced network corresponds to node 11 of the original one. Since this cell contains only one element, the associated weight is simply σ. This explains why the inter-modular coupling σ stayed the same. In this manner, we see how the reduced network represents the averaged connections between the cells of the original network.

2.5 Summary and Problem Statement

In this chapter, we laid the groundwork for the investigation of clustering-based model order reduction in diffusively coupled network systems. We reviewed some basic graph theoretical notions and used these to define precisely what is meant when we talk about diffusively coupled network systems. To model dynamics at the individual nodes of our network, we chose the FitzHugh-Nagumo (FHN) oscillator. The FHN oscillator is a common choice in modelling neuronal activity. Having established the individual node dynamics, we introduced two example networks which we will analyse throughout this thesis. On the one hand, we have Steur's network, which is essentially a directed modular ring network. On the other hand, we have Pogromsky's network, which has a cube-like character with very symmetric coupling which is controlled by 4 system parameters. For both systems, we observed both full and cluster synchronization, i.e. depending on parameter configuration and initial conditions, some oscillators in the network would synchronize while others would not. Finally, we introduced clustering-based model order reduction, a model order reduction technique for network systems in which we approximate the behaviour of a given network by a reduced network with fewer nodes. Nodes in this reduced network represent groups of nodes from the original network. We are now able to give a precise and concise problem statement.

Problem Statement

The purpose of this thesis is to study the performance of clustering-based model order reduction for network systems of nonlinear oscillators with diffusive coupling. Different choices of node

clusterings in the network will be investigated with respect to how well the associated reduced-order system approximates the behaviour of the original system. The question then becomes: how does one evaluate the performance of a choice of node clustering? For selected dynamical phenomena, such as the travelling wave in Steur's network, the (dis)similarity of bifurcation plots of the full and reduced-order system will be invoked to evaluate the choice of node clustering. This approach reduces our investigation to a case-by-case analysis.

In Steur's network and Pogromsky's network, cluster synchronization phenomena have been observed, and it seems like certain nodes tend to synchronize "more easily" and quicker than others. Motivated by these observations, this thesis aims to provide formal argumentation to support the intuition that clustering together nodes which are likely to synchronize (quickly) will result in a reduced-order system that approximates the original system well. However, the treatment of this problem in full generality is out of the scope of this thesis and so the analysis is restricted to the study of this problem for Steur's network and Pogromsky's network.

It is clearly not useful to consider any arbitrary choice of node clustering for model order reduction and so this leaves three overall tasks:

1. Reduce the number of candidates for node clustering to be considered for clustering-based model order reduction.

2. Investigate what kind of node clustering produces the best approximations.

3. Predict which node clusterings will generate the best approximations to the original system.

In summary, the purpose of this thesis is to study Steur's network and Pogromsky's network, where the subsystems are identical FitzHugh-Nagumo oscillators, to find out if certain node clusterings are better suited for clustering-based model order reduction than others, and if so, why.

As it turns out, the symmetries of a network system are intimately related to the occurrence of cluster synchronization phenomena. This means that node clusterings associated with these symmetries make for a great starting point of our investigations. The next chapter dives further into the theoretical underpinnings of this relationship.

Chapter 3

Cluster Synchronization

In chapter 2, we started our investigation of diffusively coupled networks and saw that, after the introduction of more convenient notation, the dynamics for such a network system takes the form

$$\begin{cases} \dot{x} = F(x) + (I_k \otimes B)u, \\ y = (I_k \otimes C)x, \end{cases} \tag{3.1}$$

with feedback

$$u = -(\mathcal{L} \otimes I_m)y.$$

The closed-loop form of this system is then given by

$$\dot{x} = F(x) - (\mathcal{L} \otimes BC)x. \tag{3.2}$$

This chapter revolves entirely around equation (3.2). Motivated by our observations from chapter 2, we will try to understand the mechanism that makes certain groups of nodes in the network synchronize and others not. We refer to this phenomenon as *cluster synchronization*, which gives this chapter its name. Naturally, the interconnection structure of the network plays a key role in understanding cluster synchronization and we will see that the symmetries of a network are intimately related to cluster synchronization.

The contents of this chapter are arranged as follows: In section 3.1 we will review the concept of symmetry of a graph and how symmetries give rise to linear invariant manifolds of system (3.2). Further, we explore a group-theoretic approach to finding all valid node configurations in which cluster synchronization can occur. In section 3.2, we explore an approach for obtaining a sufficient condition for global asymptotic stability of the symmetry-associated linear invariant manifolds as presented in Pogromsky [22, 23]. Then, in section 3.3, we derive a modified sufficient condition that is applicable for network systems of FHN oscillators.

3.1 Graphs and Symmetries

In this first section, we will review the notion of symmetry of a graph and a method that allows to identify all groupings of nodes in which cluster synchronization may occur. We will see that with each such grouping, there is associated a linear invariant manifold of the network system.

To study cluster synchronization is to study these linear invariant manifolds and, in particular, their stability. For now, let us get started with the mathematical description of symmetries of a graph.

3.1.1 Symmetries of a Graph

We consider both undirected and directed graphs $G = (V, E, A)$ and wish to make precise the notion of a symmetry of this graph.

Definition 3.1.1. A *symmetry of the directed graph* G is a permutation $g : V \to V$ that acts on the set of vertices such that $(i, j) \in E$ if and only if $(g(i), g(j)) \in E$ with $a_{ij} = a_{g(i)g(j)}$.

The notion of symmetry for undirected graphs is defined analogously. Definition 3.1.1 essentially says that if g is a symmetry of G, then rearranging the vertices according to g does not change the structure of the graph. An example to illustrate this notion is in order.

Example 3. Consider Steur's network with $\bar{M} = 3$ and $\bar{N} = 3$ and in particular the first module, so nodes $1, 2$ and 3, as illustrated in Figure 3.1. Let $g : V \to V$ be the permutation that transposes vertices 2 and 3 and leaves the other vertices unchanged, i.e.

$$g(2) = 3, \quad g(3) = 2, \quad g(i) = i \text{ for } i \in V \backslash \{2, 3\}.$$

In a single module of Steur's network, we have $(i, j) \in E$ for distinct $i, j \in \{1, 2, 3\}$ and so clearly

$$(g(i), g(j)) \in E \text{ for distinct } i, j \in \{1, 2, 3\}.$$

Further, note that $a_{ij} = a_{ji} = \mu$ for distinct $i, j \in \{1, 2, 3\}$ and so, by Definition 3.1.1, g is indeed a symmetry.

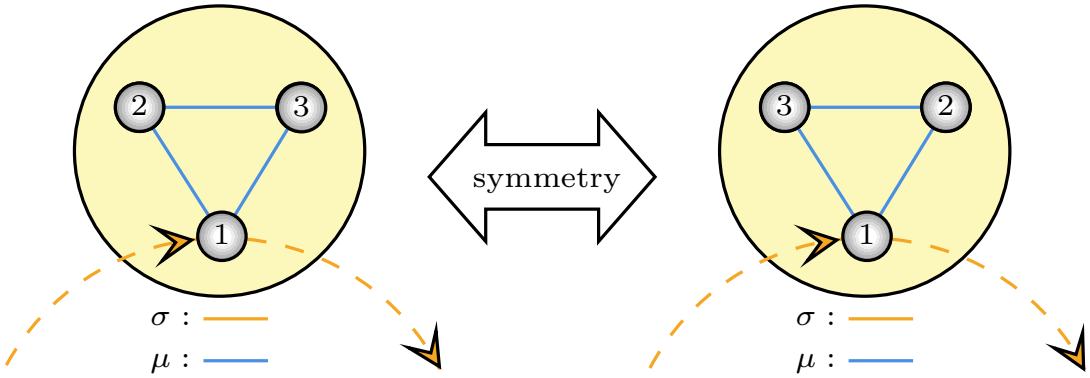

Figure 3.1: A simple symmetry of Steur's network.

Remark. There are different notions of the concept of symmetry to be found in the literature. For example, Pogromsky et al. [26] refer to the kind of symmetry given in Definition 3.1.1 as *global symmetries*. They contrast this against the notion of *internal symmetries*, by which they mean symmetries which are present in the individual subsystems of the network. In the context of this thesis, the distinction between global and internal symmetries is not necessarily useful and so we restrict our attention to global symmetries which shall henceforth simply be referred to as symmetries.

The symmetries of a graph come with their own convenient structure: Suppose a graph has two symmetries g_1 and g_2, then also their composition $g_1 \circ g_2$ is a symmetry. It should not come as a surprise that the set of all symmetries of a graph has a group structure.

Definition 3.1.2. The *symmetry group of* G, denoted by $\mathcal{G}$, is the group of all symmetries of G.

We now have made precise what is meant when we talk about the symmetry of a graph and also recalled that the symmetries of a graph form a group. These abstract notions are useful, however we also need to establish how they enter the equations that we use to describe the dynamics of network systems.

If we consider the dynamics in equation (3.2) then rather obviously, the underlying graph structure enters this equation through the associated Laplacian $\mathcal{L}$. As is well known from spectral graph theory, the graph Laplacian contains a lot of information about the topology of the underlying network: The dimension of the square matrix $\mathcal{L}$ is equal to the number of individual subsystems in the network. The number of disconnected components of the network is equal to the dimension of the subspace $\ker(\mathcal{L})$. The number of subsystems connected to a specific other subsystem, as well as coupling strengths are represented by the nonzero elements found in a specific row or column of $\mathcal{L}$. Lastly, also symmetries that are present in the network graph can be found by studying $\mathcal{L}$.

3.1.2 Symmetries and Linear Invariant Manifolds

Suppose we are given a graph G with associated Laplacian $\mathcal{L}$. Now we would like to find out if the symmetries of G somehow tell us something about system (3.2). So far, we considered the symmetries of G in a rather abstract sense, so let us introduce a more tangible way of thinking about these symmetries. We will use matrices to represent permutations.

Let g be a permutation of the set $V = \{1, 2, \ldots, N\}$ and

$$\Pi_g \in \mathbb{R}^{N \times N},$$

be such that the columns of Π_g are given by $e_{g(1)}, e_{g(2)}, \ldots, e_{g(N)}$ where $e_1, e_2 \ldots, e_N$ are the columns of the identity matrix I_N. Note that the set of permutation matrices forms a group with respect to matrix multiplication. This group is isomorphic to the group of permutations. In addition, note that permutation matrices are unitary, i.e. we have

$$\Pi_g^\top \Pi_g = I_N.$$

Rather than thinking about symmetries of G in terms of abstract permutations on the set of nodes, we now consider them as permutation matrices acting on the node space. We then need to establish how the notion of "symmetry of a graph" (Definition 3.1.1) translates to permutation matrices.

Lemma 3.1.1. *Let $G = (V, E, A)$ be a graph, $\mathcal{L}$ be the associated Laplacian matrix and $\mathcal{G}$ be*

the symmetry group of G. Then for any $g \in \mathcal{G}$, it holds that

$$\Pi_g \mathcal{L} = \mathcal{L} \Pi_g. \tag{3.3}$$

The proof of this lemma can be found in Appendix A. With a slight abuse of terminology, we will say that a permutation matrix Π is a symmetry of the graph G (with Laplacian $\mathcal{L}$) if $\Pi\mathcal{L} = \mathcal{L}\Pi$.

Let us now return to the question of how symmetries of the underlying graph lead to linear invariant manifolds of the system

$$\dot{x} = F(x) - (\mathcal{L} \otimes BC)x. \tag{3.4}$$

We recall the definition of a linear invariant manifold.

Definition 3.1.3. Given a system of the form (3.4) and some matrix $M \in \mathbb{R}^{Nn \times Nn}$, the linear manifold

$$\mathcal{A}_M = \{x \in \mathbb{R}^{Nn} | Mx = 0\}$$

is *invariant* if $M\dot{x} = 0$ whenever $Mx = 0$.

What this definition says is that if at some time t_0 some trajectory lies on $\mathcal{A}_M$, then the trajectory will remain on the manifold for all time, i.e.

$$x(t_0) \in \mathcal{A}_M \implies x(t) \in \mathcal{A}_M, \quad \forall t \geq t_0.$$

Given $F(\cdot)$ and $(\mathcal{L} \otimes BC)$, identifying linear invariant manifolds of the system (3.4) then boils down to finding matrices M such that

$$MF(x) - M(\mathcal{L} \otimes BC)x = 0, \tag{3.5}$$

for all x such that $Mx = 0$. Unfortunately, there is no general approach on how to find such matrices M, however, it is possible to exploit the symmetries of the underlying graph to find a class of matrices that solve (3.5). We will make use of the following lemma.

Lemma 3.1.2. *(Pogromsky et al. [26]) Given a permutation matrix Π such that*

$$\Pi\mathcal{L} = \mathcal{L}\Pi,$$

the set

$$\mathcal{S}_\Pi := \ker(I_{Nn} - \Pi \otimes I_n)$$

is a linear invariant manifold of system (3.4).

Proof. For brevity of notation, let $\Sigma = \Pi \otimes I_n$. Assume that $x \in \mathcal{S}_\Pi$, i.e.

$$(I_{Nn} - \Sigma)x = 0. \tag{3.6}$$

Recall that $F(\cdot)$ has a special structure, namely

$$F(x) = \text{col}(f(x_1), \ldots, f(x_N)),$$

and thus $\Sigma F(x) = F(\Sigma x)$. By assumption, and making use of the properties of the Kronecker product, we then have

$$\Sigma(\mathcal{L} \otimes BC) = (\Pi\mathcal{L} \otimes BC) = (\mathcal{L}\Pi \otimes BC) = (\mathcal{L} \otimes BC)\Sigma.$$

A direct computation shows

$$(I_{Nn} - \Sigma)\dot{x} = 0,$$

since we assumed that equation (3.6) holds. So indeed, $\mathcal{S}_\Pi$ is a linear invariant manifold of system (3.4). ■

This lemma shows how symmetries of the underlying graph give rise to linear invariant manifolds in the network system. However, it is a possibility that there are permutation matrices with $\Pi\mathcal{L} \neq \mathcal{L}\Pi$ and yet the set $\mathcal{S}_\Pi$ still turns out to be invariant. Consider the following extension of Lemma 3.1.2.

Lemma 3.1.3. *(Pogromsky [22]) Given a permutation matrix* Π *such that*

$$(I_N - \Pi)\mathcal{L} = X(I_N - \Pi), \tag{3.7}$$

has a solution X*, the set* $\mathcal{S}_\Pi$ *is a linear invariant manifold of system (3.4).*

The proof is analogous to the proof of Lemma 3.1.2.

We now know how symmetries of the underlying graph give rise to linear invariant manifolds. In the case that a certain permutation matrix Π commutes with the Laplacian $\mathcal{L}$, we associate with Π the invariant set $\mathcal{S}_\Pi$. As shown in Lemma 3.1.3, permutation matrices which correspond to symmetries of the graph are not necessarily the only ones for which $\mathcal{S}_\Pi$ is invariant. Before moving on to the investigation of the stability properties of these invariant sets, we will study an approach to find *all* permutation matrices for which $\mathcal{S}_\Pi$ is invariant.

3.1.3 Cluster Synchronization Patterns

We previously saw that for permutation matrices Π such that equation (3.7) can be solved, the set

$$\mathcal{S}_\Pi := \ker(I_{Nn} - \Pi \otimes I_n)$$

is a linear invariant manifold of system (3.4). Given such a matrix Π, we will refer to the set of pairs of indices (i, j) for which

$$x_i - x_j = 0$$

whenever $x \in \mathcal{S}_\Pi$, as the *cluster synchronization (CS) pattern* associated with Π and denote it as $\mathcal{I}_\Pi$, i.e.

$$\mathcal{I}_\Pi = \{(i, j) \,|\, x \in \mathcal{S}_\Pi \implies x_i - x_j = 0\}.$$

In this way, a natural way to partition the vertex set arises: Given a permutation matrix Π, we define the *orbit partition* π_Π associated with Π. The defining feature of this partition is that nodes i and j belong to the same cell if and only if $(i, j) \in \mathcal{I}_\Pi$. The following example

illustrates the relationship between a permutation matrix Π, the associated orbit partition and the characteristic matrix of the orbit partition.

Example 4. Let

$$\Pi = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

We see that this permutation transposes nodes 1 and 2, while leaving node 3 unchanged. Further, nodes 4, 5 and 6 are all cyclically shifted by one place ($4 \to 5, 5 \to 6, 6 \to 4$). The associated orbit partition is thus

$$\pi_\Pi = \{\{1, 2\}, \{3\}, \{4, 5, 6\}\}$$

and the characteristic matrix of the orbit partition is

$$P(\pi_\Pi) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Remark. The notation $P(\pi_\Pi)$ seems quite cumbersome and to involve too many variants of the letter P. However, **p**ermutation matrices as well as **p**artitions are commonly denoted by Π and π respectively. Recalling from section 2.4 that we use the characteristic matrix of a partition for purposes of **p**rojection, this kind of notation does, after all, seem fitting.

We saw earlier that we need to make a fundamental distinction between CS patterns which arise as a direct consequence of symmetries present in the graph, i.e. for which $\Pi\mathcal{L} = \mathcal{L}\Pi$, and those for which

$$(I_N - \Pi)\mathcal{L} = X(I_N - \Pi),$$

can merely be solved for $X \neq \mathcal{L}$. We will refer to the clusters giving rise to these fundamentally different patterns as *symmetry clusters* and *Laplacian clusters* respectively. The term "symmetry cluster" is self-explanatory, while the term "Laplacian cluster" has been introduced by Sorrentino et al. [30, 21].

In their work, Sorrentino et al. outline an approach that allows to identify all admissible CS patterns. They distinguish two types of network coupling, one involving the adjacency matrix of the underlying network and one involving the Laplacian matrix. In the case of adjacency coupling, they show that all admissible CS patterns stem from the symmetries of the underlying network. So in this case, given the adjacency matrix A, all admissible CS patterns can be obtained by finding all permutation matrices such that $\Pi A = A\Pi$. Even though the problem of finding these matrices Π has not been shown to be solvable in polynomial-time, there are discrete algebra routines available that allow for solving this problem on graphs with relatively many

nodes (e.g. $N = 22,322$ nodes for the Internet at the level of Autonomous Systems, MacArthur et al. [14]).

In the case of Laplacian coupling, which is the type of coupling we are concerned with in this thesis, the CS patterns due to symmetry are still valid. This is because, given an adjacency matrix A and corresponding Laplacian $\mathcal{L}$, we have

$$\Pi A = A\Pi \iff \Pi\mathcal{L} = \mathcal{L}\Pi,$$

as the row sums of A are not affected by the permutation operations. As already discovered, Laplacian coupling allows for more CS patterns than just those arising by symmetry, hence the name "Laplacian clusters" for the (possibly) additional clusters. It is not immediately obvious how these patterns can be obtained. Let us see how Sorrentino et al. [30] use the symmetry patterns to construct all valid Laplacian patterns.

Given a graph G, it is possible to decompose the symmetry group $\mathcal{G}$ into subgroups $\mathcal{H}_i, i = 1, \ldots, v$ where each of the subgroups acts only on some subset of clusters but not on any of the other clusters (see MacArthur et al. [14]). We will refer to the subgroups $\mathcal{H}_i$ as *cluster groups*. Note that $\mathcal{G}$ can be written as a direct product

$$\mathcal{G} = \mathcal{H}_1 \times \mathcal{H}_2 \times \ldots \times \mathcal{H}_v.$$

Uniqueness (up to isomorphism) of this decomposition is shown in MacArthur et al. [14]. Decomposing each $\mathcal{H}_i$ into all of its possible subgroups yields all valid CS patterns for symmetry clusters. Finally, to construct all valid Laplacian patterns, Sorrentino et al. devised the following three-step method:

1. From each cluster group $\mathcal{H}_i$ choose either the full cluster group or a subgroup thereof. Together this yields a subgroup $\mathcal{G}'$ of the original symmetry group $\mathcal{G}$.

2. Merge some of the clusters/sub-clusters associated with the selected subgroups from the first step. These merged clusters are candidates that might yield an additional CS pattern.

3. Given a merged cluster, set all x_i that belong to the new cluster equal to one another and check whether the equations of motions are identical in which case the linear manifold associated with the new CS pattern is flow-invariant.

Remarks.

i) In step 2: Note that the merged clusters cannot arise due to the original symmetries of the network, yet they may yield CS patterns that are dynamically valid in the case of Laplacian coupling.

ii) In step 3: For larger networks, this step of checking the equations of motion can become intractable when done by hand. Making again use of available discrete algebra tools, Sorrentino et al. provide an elegant way of implementing this last step computationally. The interested reader is referred to the original article [30] for details.

iii) A word on computational complexity of the proposed method: If κ is the number of clusters and sub-clusters, then the number of tests that need to be performed is bounded above by, the κth Bell number B_κ where

$$B_0 = 1, \quad B_{\kappa+1} = \sum_{i=0}^{\kappa} \binom{\kappa}{i} B_\kappa.$$

In general, this behaviour is of course very undesirable. We conclude that the application of this algorithm is feasible for networks with few nodes or networks with few symmetries, i.e. $\kappa << N$.

The following example illustrates this procedure.

Example 5. Consider a graph G as depicted in Figure 3.2.

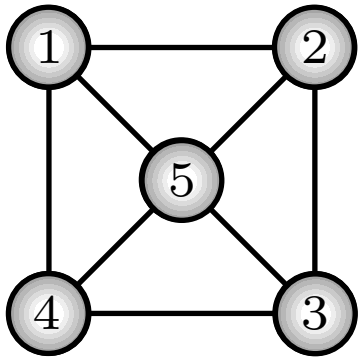

Figure 3.2: Undirected example graph. All edges have a weight of 1.

Decomposing the symmetry group of G as proposed gives $\mathcal{G} = \mathcal{H}_1 \times \mathcal{H}_2$ where $\mathcal{H}_1$ is the set of permutations acting on $\{1, 2, 3, 4\}$ and $\mathcal{H}_2$ simply contains the identity on the set $\{5\}$. Note that the permutations in $\mathcal{H}_1$ indeed only act on the subset $\{1, 2, 3, 4\}$ and not on $\{5\}$ and vice versa for the identity element in $\mathcal{H}_2$. Examples of partitions with symmetry patterns would be $\pi_1 = \{\{1, 2, 3, 4\}, \{5\}\}$ and $\pi_2 = \{\{1, 3\}, \{2, 4\}, \{5\}\}$. Let us now apply the mentioned three-step method to arrive at partitions with Laplacian patterns:

1. We have the decomposition $\mathcal{G} = \mathcal{H}_1 \times \mathcal{H}_2$. We simply select $\mathcal{H}_1$ and $\mathcal{H}_2$ to arrive at $\mathcal{G}'$ which is in this case simply $\mathcal{G}$, so $\mathcal{G}' = \mathcal{G}$.

2. We now consider $\{1, 2, 3, 4\} \cup \{5\} = \{1, 2, 3, 4, 5\}$ as a candidate for a Laplacian cluster.

3. Indeed, setting $x_1 = x_2 = \ldots = x_5$ the coupling term disappears and the associated set is invariant.

A more complicated situation is the following:

1. Decompose $\mathcal{G} = \mathcal{H}_1 \times \mathcal{H}_2$ and choose $\bar{\mathcal{H}}_1 \subseteq \mathcal{H}_1$ and $\mathcal{H}_2$ where $\bar{\mathcal{H}}_1$ is the subgroup generated by the permutation transposing both nodes 1 and 3 as well as 2 and 4. This yields $\mathcal{G}' = \bar{\mathcal{H}}_1 \times \mathcal{H}_2 \subseteq \mathcal{G}$ (here $\subseteq$ denotes "is a subgroup of").

2. The partition with associated symmetry pattern is $\pi = \{\{1, 3\}, \{2, 4\}, \{5\}\}$. Candidates for Laplacian clusters in this case are $\{1, 3\} \cup \{5\} = \{1, 3, 5\}$ and $\{2, 4\} \cup \{5\} = \{2, 4, 5\}$.

3. Checking the equations of motion, we see that setting $x_1 = x_3 = x_5$ and $x_2 = x_4$ is dynamically valid, i.e. the associated manifold is indeed invariant. The same holds for $x_1 = x_3$ and $x_2 = x_4 = x_5$. This yields two new partitions with Laplacian patterns, namely $\{\{1,3,5\},\{2,4\}\}$ and $\{\{1,3\},\{2,4,5\}\}$.

This illustrates how Laplacian clusters can be formed by combining symmetry clusters.

Before moving on to matters of stability analysis, let us briefly review the contents of this section: We recalled the concept of the symmetry group $\mathcal{G}$ associated with a graph G. Given some permutation $g \in \mathcal{G}$, we saw that the associated permutation matrix Π_g commutes with the Laplacian $\mathcal{L}$ of the underlying graph. For a network system of diffusively coupled oscillators with dynamics

$$\dot{x} = F(x) - (\mathcal{L} \otimes BC)x,$$

the symmetries of the underlying graph give rise to linear invariant manifolds. However, network symmetries are not the only way to identify linear invariant manifolds: There might be additional linear invariant manifolds which arise due to diffusive coupling and the appearance of the Laplacian matrix in the coupling term. We distinguish between these two cases by referring to the associated clusters as *symmetry clusters* and *Laplacian clusters*. Finally, we explored an algorithm that allows for the construction of all valid CS patters, involving both symmetry and Laplacian clusters.

3.2 A Passivity-Based Approach

We previously saw how a group-theoretic approach can be used to identify a class of linear invariant manifolds of the system (3.2), i.e. of the system

$$\dot{x} = F(x) - (\mathcal{L} \otimes BC)x.$$

These linear invariant manifolds are associated with (generalized) symmetries of the underlying graph. We now turn to the stability analysis for these linear invariant manifolds. In this, the central result which we make use of is presented in an article by Pogromsky [22]. The result essentially states that for diffusively coupled networks with strictly semi-passive subsystems, an arbitrary linear invariant manifold associated with a symmetry of the underlying graph becomes globally asymptotically stable provided the overall coupling strength is sufficiently large.

Before diving into the details, let us briefly outline the overall strategy that Pogromsky follows in proving his result:

1. Establish that there are linear invariant manifolds of system (3.2) which are associated with symmetries of the underlying graph (see section 3.1 of this thesis).

2. Introduce the notion of passive systems to arrive at a sufficient condition that guarantee that solutions of the network system exist and are ultimately bounded.

3. Introduce the notion of convergent systems and state a sufficient condition which guarantees that a given system is in fact convergent.

4. State and prove a sufficient condition which guarantees global asymptotic stability of a linear invariant manifold associated with a particular symmetry.

All important notions and steps will be presented in detail and where appropriate, the reader will be referred to the original material for further detail.

3.2.1 Passivity and Boundedness of Solutions

As was introduced in chapter 2, we consider network systems where the the individual subsystems are described by equations of the form

$$\begin{cases} \dot{x} = f(x) + Bu, \\ y = Cx, \end{cases} \tag{3.8}$$

with $x \in \mathbb{R}^n, u \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$. Further, we required that $f(0) = 0$ and B, C are constant matrices of appropriate dimension.

The notion of (semi-)passivity is commonly employed in the control theory literature to arrive at sufficient conditions which guarantee the existence and ultimate boundedness of solutions of systems such as (3.8). Therefore, the notion of semi-passivity will play a key role in the stability analysis we are about to undertake.

Definition 3.2.1. We call system (3.8) *passive*, if there exists a differentiable and non-negative function $S : \mathbb{R}^n \to \mathbb{R}$ such that $S(0) = 0$ and, along the solutions of (3.8), the function satisfies

$$\dot{S}(x, u) := \frac{\partial S}{\partial x}(x)(f(x) + Bu) \leq y^\top u. \tag{3.9}$$

We refer to the function S as the *storage function* and inequality (3.9) as the *dissipation inequality*.

What is the intuition behind this definition?

In physical systems such as mechanical or electrical ones, the function S can often be chosen to be the energy stored in the system. If we consider the solutions of the system on some time interval $[0, t]$ then the dissipation inequality implies that

$$S(x(t)) \leq S(x(0)) + \int_0^t y^\top(\tau)u(\tau)d\tau,$$

which essentially means that the energy stored in the system at some time point t is always less than or equal to the energy at the initial time $t = 0$ plus whatever energy has been supplied to the system through the external input u. Passive systems thus do not "produce energy" internally, where the notion of energy is to be understood in a more abstract way rather than in the conventional physical one. In this sense, we can already get an idea of how passivity may become useful when considering the problems of existence and boundedness of solutions. Let us consider a handy variation of passivity.

Definition 3.2.2. We call system (3.8) *semi-passive*, if there exists a differentiable and non-

negative function $S : \mathbb{R}^n \to \mathbb{R}$ such that, along the solutions of (3.8), the function satisfies

$$\dot{S}(x, u) \leq y^\top u - H(x),$$

where the function $H : \mathbb{R}^n \to \mathbb{R}$ has the property that there exists $r > 0$ and a continuous and non-negative function ρ, defined for $|x| \geq r$, such that

$$|x| \geq r \implies H(x) \geq \rho(x). \tag{3.10}$$

If, moreover, inequality (3.10) holds for some positive function ρ, then we refer to system (3.8) as *strictly semi-passive*.

Pogromsky [22] gives a concise characterization of passive-systems: "In brief, a semi-passive system behaves like a passive system for sufficiently large $|x|$". This conceptualization is particularly illuminating.

Regarding the problem of existence and boundedness of solutions for system (3.8), if we assume that the system is strictly semi-passive and the storage function S is radially unbounded, i.e.

$$|x| \to \infty \implies S(x) \to \infty,$$

then any choice of feedback of the form

$$y^\top \phi(y) \leq 0,$$

will render solutions of the closed loop system ultimately bounded. This can be shown by considering S as a Lyapunov function candidate and applying standard arguments from stability theory (see e.g. chapter 4 of Khalil [11]). We can derive the following lemma.

Lemma 3.2.1. *(Pogromsky et al. [24]) We consider a diffusively coupled network*

$$\dot{x} = F(x) - (\mathcal{L} \otimes BC)x, \tag{3.11}$$

as introduced at the beginning of this chapter. If each individual system of the form (3.8) is strictly semi-passive and if the Laplacian $\mathcal{L}$ is such that $\mathcal{L} + \mathcal{L}^\top$ is positive semi-definite, then the solutions of (3.11) exist for all $t \geq 0$ and are ultimately bounded.

A proof of a slightly more general result is given in Pogromsky et al. [24].

Let us briefly recapitulate, what has been discussed: we introduced the notion of passive systems and in particular the notion of strictly semi-passive systems. With this notion, we can formulate a sufficient condition which guarantees the existence and ultimate boundedness of solutions of diffusively coupled networks. This sufficient condition decouples into two parts: on the one hand, we require that the individual subsystems within the network are strictly semi-passive, on the other hand, we need the symmetrized Laplacian to be positive semi-definite. As we will soon see, Lemma 3.2.1 is a key element that allows for the stability analysis of synchronized states in network systems with diffusive coupling.

3.2.2 Convergent Systems

We consider systems of the form

$$\dot{z} = q(z, w(t)), \tag{3.12}$$

with $z \in \mathbb{R}^l$ and $w(t)$ a bounded external signal.

Definition 3.2.3. We call system (3.12) *convergent* if for any bounded function $w(t)$ defined for $t \in (-\infty, \infty)$ there is a unique, bounded and globally asymptotically stable solution $\bar{z}(t)$, defined also for $t \in (-\infty, \infty)$, such that

$$\lim_{t\to\infty} |z(t) - \bar{z}(t)| = 0,$$

for all initial conditions.

The existence of the unique limit mode $\bar{z}(t)$ which only depends on the external signal and not on the initial condition makes convergent systems closely related to the study of cluster synchronization: Consider two trajectories z_1 and z_2 of system (3.12) with potentially different initial conditions. For their difference we have

$$\lim_{t\to\infty} |z_1(t) - z_2(t)| \leq \lim_{t\to\infty} (|z_1(t) - \bar{z}(t)| + |\bar{z}(t) - z_2(t)|) = 0.$$

In order to be able to apply this notion of convergent systems to our stability analysis, we consider the following sufficient condition as formulated by Pogromsky et al. [26] as a variation on the original result by Demidovich [6].

Theorem 3.2.1 (Pogromsky et al. [26]/Demidovich [6])**.** *Suppose there exists a positive definite symmetric $l \times l$ matrix P such that all eigenvalues $\lambda_i(Q(z, w))$ of the symmetric matrix*

$$Q(z, w) = \frac{1}{2}\Big[P\Big(\frac{\partial q}{\partial z}(z, w)\Big) + \Big(\frac{\partial q}{\partial z}(z, w)\Big)^\top P\Big] \tag{3.13}$$

are negative and separated from zero, i.e. there exists a $\delta > 0$ such that

$$\lambda_i(Q(z, w)) \leq -\delta < 0, \tag{3.14}$$

for $i = 1, \ldots, l$ and for all $z, w \in \mathbb{R}^l$, then system (3.12) is convergent and the quadratic function $W(\zeta) = \zeta^\top P \zeta$ satisfies

$$\frac{\partial W(z_1 - z_2)}{\partial \zeta}[q(z_1, w) - q(z_2, w)] \leq -\bar{\alpha}|z_1 - z_2|^2 \tag{3.15}$$

for some $\bar{\alpha} > 0$.

We are now ready to state and prove the main theorem of this section.

3.2.3 Stability Analysis

To show stability of the set $\mathcal{S}_\Pi$, we will try to find a Lyapunov function. In particular, we will construct this Lyapunov function as the sum of a term which depends on input-output relations

of the individual subsystems and another term which depends on the interconnection structure. To do this, it will be useful to apply a change of coordinates

$$x \mapsto \mathrm{col}(z, y) \tag{3.16}$$

such that only the equation for y involves the interconnection structure. Recall again the dynamics of the subsystems as in system (3.8), i.e.

$$\begin{cases} \dot{x} = f(x) + Bu, \\ y = Cx. \end{cases} \tag{3.17}$$

Differentiating y with respect to time yields

$$\dot{y} = Cf(x) + CBu.$$

It is not immediately obvious, but it can be shown that if CB is non-singular, then we can find a set of $n - m$ complementary variables z such that system (3.17) takes the form

$$\begin{cases} \dot{z} = q(z, y), \\ \dot{y} = a(z, y) + CBu, \end{cases} \tag{3.18}$$

with a and q vector functions. This transformation is globally defined due to the linear input-to-output relation and an explicit construction of such a transformation can be found in Pogromsky et al. [24]. From the equation for z in system (3.18), it now also becomes apparent how the notion of convergent systems becomes applicable.

For a diffusively coupled network system with associated Laplacian $\mathcal{L}$ and with equations

$$\begin{cases} \dot{x}_j = f(x_j) + Bu_j, \\ y_j = Cx_j, \end{cases} \tag{3.19}$$

and feedback

$$u_j = -\sum_{i=1}^{N} a_{ji}(y_j - y_i),$$

we consider the following sufficient condition for global asymptotic stability.

Theorem 3.2.2 (Pogromsky [22])**.** *Suppose that*

1. *Each system (3.19) in the network is strictly semipassive with respect to the input u_j and output y_j with a radially unbounded storage function.*
2. *There exists a positive definite matrix P such that equation (3.14) holds with some $\delta > 0$ for the matrix Q as defined in equation (3.13) for q as in equation (3.18).*
3. *CB is positive definite.*
4. *The symmetrized Laplacian $\mathcal{L} + \mathcal{L}^\top$ is positive semi-definite.*

5. *There is an $N \times N$ matrix solution X of the following linear equation*

$$(I_N - \Pi)\mathcal{L} = X(I_N - \Pi).$$

Let λ' be the largest number such that the inequality

$$\frac{1}{2}\xi^\top (I_N - \Pi)^\top (X + X^\top)(I_N - \Pi)\xi \geq \lambda' \xi^\top (I_N - \Pi)^\top (I_N - \Pi)\xi$$

holds for all $\xi \in \mathbb{R}^N$.
Then all solutions of system (3.2) are ultimately bounded and there exists a positive threshold $\bar{\lambda}$ such that if $\lambda' > \bar{\lambda}$, then the set $\ker(I_{Nn} - \Pi \otimes I_n)$ contains a globally asymptotically stable compact subset.

Remark. In the case that $X + X^\top$ commutes with Π, the number λ' is the minimal eigenvalue of $\frac{1}{2}(X + X^\top)$ under the restriction that the eigenvectors of $\frac{1}{2}(X + X^\top)$ are taken from the set $\mathrm{im}(I_N - \Pi)$.

Proof. We present a brief outline of the proof. For the full derivation, the reader is referred to Pogromsky [22].

First note that, strictly speaking, we require the matrix CB to be only *similar to a positive definite matrix*. However, in this case we could simply apply the similarity transformation to input and output of system (3.18) and rename our variables. Using assumption 3, we define

$$z = \mathrm{col}(z_1, \ldots, z_N),$$
$$y = \mathrm{col}(y_1, \ldots, y_N),$$

where $x_i \mapsto \mathrm{col}(z_i, y_i)$ is such that the equations governing the individual subsystems take the form (3.18). We consider the Lyapunov function candidate

$$V(z, y) = V_1(z) + V_2(y),$$

with

$$\begin{aligned} V_1(z) &= \sum_{(i,j)\in\mathcal{I}_\Pi} W(z_i - z_j), \\ V_2(y) &= \frac{1}{2} y^\top (I_{Nm} - \Pi \otimes I_m)^\top (I_{Nm} - \Pi \otimes I_m) y \\ &= \frac{1}{2} \sum_{(i,j)\in\mathcal{I}_\Pi} |y_i - y_j|^2. \end{aligned}$$

By assumption 2, $W(\xi)$ can be chosen such that inequality (3.15) is satisfied. Assumption 5 guarantees that $\mathcal{S}_\Pi$ is invariant and by construction

$$V(z, y) = 0, \quad \forall x \in \mathcal{S}_\Pi,$$

so it remains to show that

$$\dot{V}(z,y) < 0, \quad x \notin \mathcal{S}_\Pi,$$

whenever λ' exceeds a certain threshold value. Assumptions 1 and 4 together with Lemma 3.2.1 guarantee that the solutions of the network system exist, are ultimately bounded and thus enter some compact region Ω in finite time. Straightforward computation of the derivative shows that on this compact region Ω, we have

$$\dot{V}(z,y) \leq -\bar{\alpha} \sum_{(i,j)\in\mathcal{I}_\Pi} |z_i - z_j|^2 + V_2(y)(C_1 - 2\lambda'\beta) + (C_2 + C_3) \sum_{(i,j)\in\mathcal{I}_\Pi} |z_i - z_j||y_i - y_j|, \quad (3.20)$$

where $\bar{\beta}$ is the minimal eigenvalue of CB and C_1, C_2 and C_3 are positive constants such that on Ω we have, respectively,

$$\begin{aligned} |a(z_i,y_i) - a(z_i,y_j)| &\leq C_1|y_i - y_j|, \quad \forall (i,j) \in \mathcal{I}_\Pi, \\ |a(z_i,y_j) - a(z_j,y_j)| &\leq C_2|z_i - z_j|, \quad \forall (i,j) \in \mathcal{I}_\Pi, \\ \left| \frac{\partial W(z_i - z_j)}{\partial \zeta} (q(z_j,y_i) - q(z_j,y_j)) \right| &\leq C_3|z_i - z_j||y_i - y_j|, \quad \forall (i,j) \in \mathcal{I}_\Pi. \end{aligned}$$

From equation (3.20) it follows that

$$\lambda' > \bar{\lambda} = \frac{1}{\bar{\beta}} \Big[\frac{C_1}{2} + \frac{(C_2 + C_3)^2}{4\bar{\alpha}} \Big] \implies \dot{V}(z,y) < 0, \quad x \notin \mathcal{S}_\Pi,$$

which proves the statement. ■

Let us take a moment to ponder the implications of this theorem. Suppose we are given the Laplacian $\mathcal{L}_0$ of some network and assume that all conditions of Theorem 3.2.2 are met. We consider the scaled Laplacian of the same network

$$\mathcal{L}(s) = s\mathcal{L}_0, \quad s > 0,$$

where s is to be considered a bifurcation parameter. Let $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_N$ be the eigenvalues of $\frac{1}{2}(\mathcal{L}(s) + \mathcal{L}^\top(s)$. If we consider permutation matrices Π such that $\Pi\mathcal{L} = \mathcal{L}\Pi$ then we can take $X = \mathcal{L}$ and for each Π, we have $\lambda' \in \{\lambda_2, \ldots, \lambda_N\}$. Note that the value of λ' depends on Π and so we will denote $\lambda' = \lambda'(\Pi)$ for the moment. Theorem 3.2.2 predicts that there is a threshold value which only depends on the dynamics of the individual subsystems. This threshold is upper bounded by $\bar{\lambda}$. It can thus happen that for two distinct permutation matrices Π_1 and Π_2, we have that $\lambda'(\Pi_1)$ exceeds the actual threshold while $\lambda'(\Pi_2)$ does not. In this case, we observe cluster synchronization associated with Π_1 but not with $\lambda'(\Pi_2)$. This means that, as s grows, we can expect a maximum of $N - 1$ bifurcations in which different cluster synchronization manifolds become globally asymptotically stable, one after the other. This is the case since λ' always takes its values from the set $\{\lambda_2, \ldots, \lambda_N\}$.

Suppose again that s is such that $\lambda'(\Pi_1)$ exceeds the threshold value while $\lambda'(\Pi_2)$ does not and we keep increasing s until also $\lambda'(\Pi_2)$ exceeds the threshold. In this case the cluster synchronization manifolds associated with both Π_1 and Π_2 are globally asymptotically stable,

which can only be the case if their intersection is globally asymptotically stable. As s increases, $\lambda'(\Pi)$ surpasses the threshold value for an ever-growing number of permutation matrices Π until the intersection of all cluster synchronization manifolds is globally asymptotically stable. In this sense, the theorem could be useful in discerning symmetries for which the associated nodes synchronize "more easily", from those for which larger coupling strengths would be required.

Practically speaking, this could be done as follows: Given some Laplacian matrix $\mathcal{L}$ we may find the symmetries of the underlying network graph. Using Theorem 3.2.2, the symmetries can be categorized into a maximum of $(N-1)$ groups. Each group is characterized by the minimal eigenvalue λ' of the symmetrized Laplacian under the restriction that the eigenvectors lie in $\mathrm{im}(I_N - \Pi)$. Note that the set $\mathrm{im}(I_N - \Pi)$ might be identical for a number of permutation matrices Π. We would then prefer groups of symmetries with a larger λ' in the context of clustering-based model reduction, as cluster synchronization associated with these symmetries sets in for smaller coupling strengths.

Before moving on to the application of Theorem 3.2.2 to the example networks, we will make a slight modification to the theorem: Theorem 3.2.2 is formulated for network systems where subsystems are of the general form as given in equation 3.17 and its proof relies on "worst case" estimates for the behaviour of the nonlinearities involved. It can therefore be expected that additional information about the subsystems would lead to less conservative results in terms of the required coupling strength to ensure cluster synchronization. In the following section, we derive a modified sufficient condition for cluster synchronization for networks of FHN oscillators.

3.3 A Modified Result for the FHN Oscillator

By explicitly incorporating the available information about the FHN oscillator, we derive a modified sufficient condition from Theorem 3.2.2.

In the general case of the passivity-based approach, as presented in Theorem 3.2.2, the proof involves the Lyapunov function candidate

$$V(x) = V_1(z) + V_2(y), \tag{3.21}$$

with

$$\begin{aligned} V_1(z) &= \sum_{(i,j)\in\mathcal{I}_\Pi} W(z_i - z_j), \\ V_2(y) &= \frac{1}{2} y^\top (I_{Nm} - \Pi \otimes I_m)^\top (I_{Nm} - \Pi \otimes I_m) y. \end{aligned}$$

In the case of a network system consisting of FHN oscillators, we have $n = 2$ and $m = n - m = 1$. Further, $W(\zeta) = p\zeta^\top\zeta$ is a valid choice for any positive number p. This will be shown explicitly in chapter 4. It follows that we can write

$$\begin{aligned} V_1(z) &= p z^\top (I_N - \Pi)^\top (I_N - \Pi) z, \\ V_2(y) &= \frac{1}{2} y^\top (I_N - \Pi)^\top (I_N - \Pi) y. \end{aligned}$$

We will now introduce a Lyapunov function candidate which is less restricted than the one proposed in the proof of Theorem 3.2.2: Let R be a diagonal and positive definite matrix, then the functions

$$V_1(z) = pz^\top Mz,$$
$$V_2(y) = \frac{1}{2}y^\top My,$$

with

$$M = (I_N - \Pi)^\top R(I_N - \Pi), \tag{3.22}$$

are again such that

$$z \in \ker(I-\Pi) \implies V_1(z) = 0 \quad \text{and} \quad z \notin \ker(I-\Pi) \implies V_1(z) > 0,$$
$$y \in \ker(I-\Pi) \implies V_2(y) = 0 \quad \text{and} \quad y \notin \ker(I-\Pi) \implies V_2(y) > 0.$$

We now compute $\dot{V} = \frac{\partial V}{\partial x}\dot{x}$ and derive a sufficient condition which guarantees that $\dot{V} < 0$. To cancel out cross terms zy, we let $p = \frac{1}{2\alpha}$ and find

$$\dot{V}(x) = -\beta z^\top Mz + \frac{1}{2}y^\top M(I-\mathcal{L})y + \frac{1}{2}y^\top(I-\mathcal{L})^\top My - \gamma y^\top M\Phi(y).$$

For the FHN oscillator, we have $\Phi(y) = \mathrm{col}(\phi(y_1), \dots, \phi(y_N))$ with $\phi(s) = s^3$. Using the following lemma, we can make use of the monotonicity of ϕ.

Lemma 3.3.1. *Let Π be an arbitrary permutation matrix. If R is diagonal and positive definite and Φ is (component-wise) monotonically increasing, then*

$$y^\top(I-\Pi)^\top R(I-\Pi)\Phi(y) \geq 0, \quad \forall y.$$

The proof of Lemma 3.3.1 can be found in Appendix A. Using Lemma 3.3.1, we see that

$$-\gamma y^\top M\Phi(y) \leq 0$$

and hence

$$\dot{V}(x) \leq y^\top M(I-\mathcal{L})y, \tag{3.23}$$

since evidently $-\beta z^\top Mz \leq 0$. The following lemma gives a sufficient condition for the negative definiteness of the matrix $M(I-\mathcal{L})$.

Lemma 3.3.2. *Let P be the characteristic matrix associated with the orbit partition of some $\Pi \neq I$ with $\mathcal{L}\Pi = \Pi\mathcal{L}$. Suppose there exists a $\rho \in \mathbb{R}$ such that*

$$2R - R\mathcal{L} - \mathcal{L}^\top R - \rho PP^\top \prec 0. \tag{3.24}$$

Then the matrix $M(I-\mathcal{L})$, with M as defined in equation (3.22), is negative definite.

The proof of Lemma 3.3.2 can be found in Appendix A. From Lemma 3.3.2 and equation 3.23, it is clear that if there exists a diagonal and positive definite matrix R together with some $\rho \in \mathbb{R}$

such that linear matrix inequality (LMI) (3.24) is satisfied, then $\dot{V} < 0$ and global asymptotical stability of the cluster synchronization manifold ensues. The issue of finding such R and ρ can be formulated as a LMI feasibility problem with respect to LMI (3.24). Software to solve such LMI problems is widely available and MATLAB also comes with a number of built-in functionalities to solve LMI feasibility and optimization problems.

Not only does this approach give us a modified criterion to check for cluster synchronization, in the case that cluster synchronization is guaranteed, we can also find an upper bound on the convergence rate with which the solution converges to the associated linear invariant manifold.

Lemma 3.3.3. *Let P be the characteristic matrix associated with the orbit partition of some $\Pi \neq I$ with $\mathcal{L}\Pi = \Pi\mathcal{L}$. If global asymptotical stability of the set $\mathcal{S}_\Pi$ can be shown by means of Lemma 3.3.2, then an upper bound for the convergence rate of the Lyapunov function, as defined in equation (3.21), along any trajectory is given by*

$$\delta_{\text{eff}} = \begin{cases} -2\alpha\beta\delta_0, & 0 < \delta_0 \leq 1, \\ -2\alpha\beta, & \delta_0 > 1, \end{cases}$$

where δ_0 is determined as

$$\delta_0 = \max\left\{\delta > 0 \,|\, (1+\alpha\beta\delta)I - \frac{1}{2}(\mathcal{L} + \mathcal{L}^\top) - \rho P P^\top \prec 0, \quad \rho \in \mathbb{R}\right\}. \tag{3.25}$$

We will make use of this result in chapter 4, when discussing cluster synchronization in the example networks.

3.4 Summary

This chapter aimed to shed some light on the mechanisms that give rise to cluster synchronization in diffusively coupled network systems. We saw that with each symmetry of the network graph, we can associate a certain linear invariant manifold. Moreover, we discussed a method that allows to identify additional linear invariant manifolds which are not directly associated with the symmetries but arise as a consequence of diffusive/Laplacian coupling.

Finally, we reviewed a theorem which provides a sufficient conditions for cluster synchronization in the case of strictly semi-passive subsystems. From this theorem, we derived a modified sufficient condition for cluster synchronization which applies in the special case of the FHN oscillator. Moreover, in the case that global asymptotical stability can be shown by means of this modified sufficient condition, an estimate for the convergence rate has been derived as well. In the following chapter, we apply the discussed theory to the example networks as introduced in chapter 2.

Chapter 4

Application to Example Networks

This chapter aims to make a connection between the contents of chapter 2 and chapter 3. The stability results as derived in chapter 3 are applied to Steur's network and Pogromsky's network, as introduced in chapter 2. In particular, Theorem 3.2.2, which gives a sufficient condition for cluster synchronization, will be applied to the example networks. We will try to extract a general principle that allows to predict which nodes in a network are good candidates to be clustered together in the context of clustering-based model reduction.

The content of this chapter is arranged as follows: In section 4.1, we show that the FHN oscillator meets the conditions on the individual subsystems as posed in Theorem 3.2.2. We then discuss the remaining conditions of Theorem 3.2.2, which pertain to the underlying network structure, as well as the implications of this theorem. Pogromsky's network will be treated in section 4.2 and Steur's network in section 4.3. Finally, we conclude the chapter with a brief summary in section 4.4. For readability's sake, a number of proofs to results as presented in this chapter are omitted here but given in Appendix A.

4.1 Conditions on the Subsystems

In this section, we show that the FHN oscillator is a strictly semi-passive system with respect to its input and output with a radially unbounded storage function (condition 1 in Theorem 3.2.2). Furthermore, we show that in the model of the FHN oscillator, the dynamics of the recovery variable z can be considered convergent in the sense of Definition 3.12 (condition 2 in Theorem 3.2.2). Moreover, we show the FHN oscillator also meets condition 3, which is of a rather technical nature. Finally, we discuss the threshold value $\bar{\lambda}$ and find upper bounds on it.

Condition 1

Recall that the governing equations for the FHN oscillator are given by

$$\begin{cases} \dot{z} = \alpha(y - \beta z), \\ \dot{y} = y - \gamma y^3 - z + u, \end{cases} \tag{4.1}$$

and so the dimensions of the subsystem components are $n = 2$ and $n - m = m = 1$. The following lemma establishes strict semi-passivity of the FHN oscillator.

Lemma 4.1.1 (cf. Lemma 1, Steur et al. [31]). *The FHN oscillator is strictly semi-passive with respect to input u and output y with a radially unbounded storage function.*

The proof of this lemma can be found in Appendix A. Having checked the first condition of Theorem 3.2.2, we move on to conditions 2 and 3.

Conditions 2 and 3

Note that in the case of the FHN oscillator, we have $B = \begin{bmatrix} 0 & 1 \end{bmatrix}^\top$ and $C = \begin{bmatrix} 0 & 1 \end{bmatrix}$ and so $CB = 1 > 0$. The implications are twofold: on the one hand we see that condition 3 is met, on the other hand, the equations of the FHN oscillator are already of the form as prescribed in equation (3.18), i.e.

$$\begin{cases} \dot{z} = q(z,y) & = \alpha(y - \beta z), \\ \dot{y} = a(z,y) + CBu & = y - \gamma y^3 - z + u. \end{cases} \tag{4.2}$$

To show that the dynamics of z is convergent, we make use of Demidovich's criterion (Theorem 3.2.1). We now need to show that there exist a positive definite matrix P such that all eigenvalues of

$$Q(z,y) = \frac{1}{2}\Big[P\Big(\frac{\partial q}{\partial z}(z,y)\Big) + \Big(\frac{\partial q}{\partial z}(z,y)\Big)^\top P\Big]$$

are negative. However, note that in our case all involved quantities are scalar: we have $\frac{\partial q}{\partial z} = -\alpha\beta$ and thus

$$Q(z,y) = -\alpha\beta p,$$

where p is a $m \times m$ matrix with $m = 1$. It follows that p is a scalar and the conditions of Demidovich's criterion are met for any real number $p > 0$. We conclude that the dynamics of the recovery variable z is indeed convergent.

The Threshold Value

The threshold value $\bar{\lambda}$ plays a central role in the formulation of Theorem 3.2.2. Recall that

$$\bar{\lambda} = \frac{1}{\bar{\beta}}\Big[\frac{C_1}{2} + \frac{(C_2 + C_3)^2}{4\bar{\alpha}}\Big],$$

and that the involved quantities C_1, C_2 and C_3 are such that

$$\begin{aligned} |a(z_i, y_i) - a(z_i, y_j)| &\leq C_1 |y_i - y_j|, \quad \forall (i,j) \in \mathcal{I}_\Pi, \\ |a(z_i, y_j) - a(z_j, y_j)| &\leq C_2 |z_i - z_j|, \quad \forall (i,j) \in \mathcal{I}_\Pi, \\ \left| \frac{\partial W(z_i - z_j)}{\partial \zeta} (q(z_j, y_i) - q(z_j, y_j)) \right| &\leq C_3 |z_i - z_j||y_i - y_j|, \quad \forall (i,j) \in \mathcal{I}_\Pi, \end{aligned}$$

where the functions $a(z,y)$ and $q(z,y)$ are as in system (4.2) and $W(\zeta) = p\zeta^\top\zeta$ with p as in Demidovich's criterion. In the case of the FHN oscillator, we have

$$C_1 = \max_{x \in \Omega, (i,j) \in \mathcal{I}_\Pi} \left|1 - \gamma(y_i^2 + y_i y_j + y_j^2)\right| \tag{4.3}$$

where Ω is the region of absorption for the full network system and $\mathcal{I}_\Pi$ is the synchronization pattern associated with Π. Furthermore,

$$C_2 = 1, \quad C_3 = 2\alpha p, \quad \bar{\alpha} = 2p\alpha\beta, \quad \bar{\beta} = 1.$$

A direct computation shows that for $p > 0$, the value of $\bar{\lambda}$ is minimal for $p = \frac{1}{2\alpha}$. This yields

$$\bar{\lambda} = \frac{C_1}{2} + \frac{1}{\beta}.$$

We see that the threshold value $\bar{\lambda}$ not only depends on the kind of subsystems present in the network system but also on the network structure and the permutation matrix Π. It is however possible to obtain an upper bound for C_1 which only depends on the number of nodes in the network graph and the parameters associated with the FHN oscillator. In this way, we can arrive at an upper bound for $\bar{\lambda}$ which depends only on the type of subsystem and the number of subsystems. To do so, we simply estimate the absorption region Ω and maximize the expression in equation (4.3) over all pairs of i and j rather than only those in $\mathcal{I}_\Pi$. The following lemma and corresponding corollary give the desired results.

Lemma 4.1.2. *Assume that the Laplacian matrix $\mathcal{L}$ associated with a diffusively coupled network of N subsystems is such that $\mathcal{L} + \mathcal{L}^\top$ is positive semi-definite. If the subsystems are FHN oscillators, then trajectories of the network system are ultimately bounded and the region of absorption Ω is contained in the ball of radius $R = \sqrt{2N}\frac{(\alpha\beta+1)}{\sqrt{4\alpha^2\beta\gamma}}$.*

The following corollary is a direct consequence of Lemma 4.1.2.

Corollary 4.1.1. *Assume that all conditions of Theorem 3.2.2 are met. For the FHN oscillator, we have*

$$C_1 \leq \begin{cases} 1, & R \leq \sqrt{\frac{2}{\gamma}}, \\ \gamma R^2 - 1, & R > \sqrt{\frac{2}{\gamma}}. \end{cases}$$

where R is as in Lemma 4.1.2.

The proofs of Lemma 4.1.2 as well as Corollary 4.1.1 can be found in Appendix A.

For the chosen values of $\alpha = 8/100, \beta = 8/10$ and $\gamma = 1/3$, we have

$$\frac{(\alpha\beta + 1)}{\sqrt{4\alpha^2\beta\gamma}} \approx 12.9,$$

and thus $R > \sqrt{\frac{2}{\gamma}}$. Finally, it follows that

$$C_1 \leq \gamma R^2 - 1 = 2N\frac{(\alpha\beta + 1)^2}{4\alpha^2\beta} - 1.$$

However, numerical simulations suggest that this upper bound is very conservative. In fact, the modified approach of producing a suitable Lyapunov function, as described in section 3.3, suggests that even $C_1 = 1$ is a conservative estimate in the case of the FHN oscillator. We will address this issue with regard to the example networks in the upcoming sections.

Having verified that the FHN oscillator meets the conditions on the subsystems as posed in Theorem 3.2.2, we move on to discuss the conditions relating to the network structure.

4.2 Pogromsky's Network

In this section, we apply Theorem 3.2.2 and the derived modified sufficient condition in order to study cluster synchronization in Pogromsky's network as introduced in section 2.3. We previously saw that conditions 1, 2 and 3 of Theorem 3.2.2 are met for the FHN oscillator. Before moving on to discuss the stability of the linear invariant manifolds associated with the symmetries of the network, we verify that conditions 4 and 5 of Theorem 3.2.2 are indeed met as well.

Conditions 4 and 5

Since Pogromsky's network is undirected, it follows that the associated Laplacian matrix $\mathcal{L}_P$ is symmetric and positive semi-definite. It follows that $\mathcal{L}_P + \mathcal{L}_P^\top = 2\mathcal{L}_P$ is also positive semi-definite and hence condition 4 is met.

The last condition stated that, given some permutation matrix Π, there exists an $N \times N$ matrix solution X to

$$(I_N - \Pi)\mathcal{L} = X(I_N - \Pi). \tag{4.4}$$

At the end of section 3.1, we reviewed an approach on how to find permutation matrices Π such that equation (4.4) has a solution X. For now we restrict to the case in which $X = \mathcal{L}$, i.e. we consider the symmetries of Pogromsky's network.

4.2.1 Symmetries

Using numerical tools, we find that Pogromsky's network has a maximum of 48 symmetries. This is of course only the case if all parameters are identical, i.e. $k_1 = k_2 = k_3 = k_4$. Weakening this restriction, the number of symmetries in the network decreases. For example, in the case of $k_1 = k_3 \neq k_2 = k_4$, there are 16 symmetries which are always present for an arbitrary parameter configuration adhering to the imposed restriction. If we set $k_2 = k_4$ and impose no further conditions, then 8 symmetries remain.

In order to better understand the symmetries of Pogromsky's network, we consider the simpler version with fixed $k_2 = k_4$ which has at least 8 symmetries regardless of the values of k_1 and k_3. Let us denote these 8 symmetries $\Pi_1, \Pi_2, \ldots, \Pi_8$. The permutation matrices are explicitly given in Appendix B. It so happens that the characteristic matrices associated with the orbit partitions of these permutation matrices are all mutually distinct. If there had been any duplicates, we would simply discard all but one of them since we are interested in the different ways one may cluster together nodes and duplicate characteristic matrices imply that the associated reduced-order models would be identical.

There are 3 "types" of symmetries for $k_2 = k_4$: The first one is simply the identity permutation. The second kind leaves 4 nodes unchanged and permutes the remaining 4 nodes in pairs

of 2 (Π_2 and Π_3). The third type permutes all nodes in pairs of 2 ($\Pi_4, \Pi_5, \Pi_6, \Pi_7$ and Π_8). The respective network reduction situation for Π_2 is illustrated in Figure 4.1.

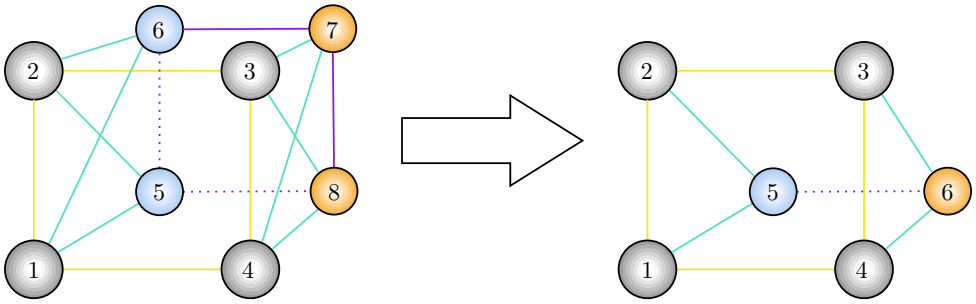

Figure 4.1: Model reduction when clustering according to the orbit partition of Π_2. Nodes of the same colour in the original network are clustered together in the reduced network.

4.2.2 Stability Regions

We consider Pogromsky's network with fixed $k_2 = k_4 = 0.5$ which leaves k_1 and k_3 as effective network parameters. Having checked all the conditions, we can finally apply Theorem 3.2.2 and the derived modified sufficient condition.

Recall that Theorem 3.2.2 involves the quantity C_1 which is a worst case estimate for the behaviour of the FHN oscillator in the region of absorption Ω. An approximation of this region resulted in the upper bound

$$C_1 \leq \gamma R^2 - 1 = 2N \frac{(\alpha\beta + 1)^2}{4\alpha^2\beta} - 1.$$

Pogromsky's network has $N = 8$ nodes and so

$$C_1 \leq 883.45$$

is a valid upper bound. As we will see in a moment, this estimate is extremely conservative. In fact, integration of the system suggests that trajectories actually do enter a region in which C_1 can be bounded above by 1 (see Corollary 4.1.1). Let us see how this compares to the modified sufficient condition which was derived in section 3.3.

By making explicit use of the dynamics of the FHN oscillator, we observed that the nonlinear term involved can be neglected when it comes to finding a suitable Lyapunov function (see section 3.3). Since a lot of conservatism in the predictions of Theorem 3.2.2 can be expected to be caused by accounting for potentially "very badly behaved" nonlinear terms, we expect the modified sufficient condition to be significantly less restrictive. This is indeed the case, as is demonstrated in the following numerical experiment.

Experimental Setup

We create a grid for a portion of the parameter plane with $k_1 = 0, 0.025, 0.05, ..., 1.5$ and $k_3 = 0, 0.025, 0.05, ..., 1.5$. For each value, we construct the associated Laplacian matrix and for a certain symmetry Π and check whether Theorem 3.2.2 or the derived modified sufficient condition

guarantee cluster synchronization as prescribed by Π. More explicitly, given Π, for each pair (k_1, k_3), we check the following two conditions:

- We calculate λ' which is the largest number such that the inequality

$$\frac{1}{2}\xi^\top (I_N - \Pi)^\top (X + X^\top)(I_N - \Pi)\xi \geq \lambda' \xi^\top (I_N - \Pi)^\top (I_N - \Pi)\xi$$

  holds for all $\xi \in \mathbb{R}^N$ and check if $\lambda' > \bar{\lambda} = \frac{1}{2} + \frac{1}{\beta}$ (i.e. $C_1 = 1$).

- We check whether there exist a diagonal matrix $R \succ 0$ and $\rho \in \mathbb{R}$ such that

$$2R - R\mathcal{L} - \mathcal{L}^\top R - \rho P P^\top \prec 0, \tag{4.5}$$

  where P is the characteristic matrix associated with the orbit partition of Π.

For each Π we thus find the region of guaranteed cluster synchronization. The results are summarized in Figures 4.2 and 4.3.

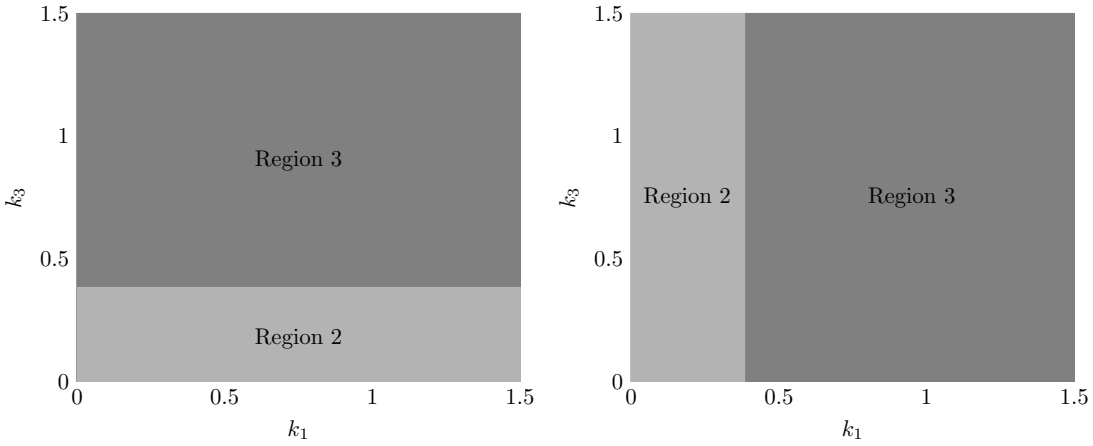

Figure 4.2: Left Panel: cluster synchronization regions for Π_2. Right panel: cluster synchronization regions for Π_3. In both panels region 2 corresponds to the parameter values for which the modified sufficient condition guarantees cluster synchronization and in region 3, both Theorem 3.2.2 and the modified sufficient condition guarantee cluster synchronization.

Remark. The criteria described are merely sufficient, so the actual stability regions contain the depicted regions. The converse is not necessarily true.

We note that in all cases, the modified criterion is less restrictive than Theorem 3.2.2 (even with $C_1 = 1$). This is in accordance with our predictions. In terms of the differences between the symmetries, we note that for symmetries Π_1, Π_2, Π_3 and Π_4, the modified condition guarantees cluster synchronization for all pairs (k_1, k_3), while for Π_5, Π_6, Π_7 and Π_8 cluster synchronization sets in only for comparatively larger values of k_1 and k_3. Let us have a closer look at what is going on.

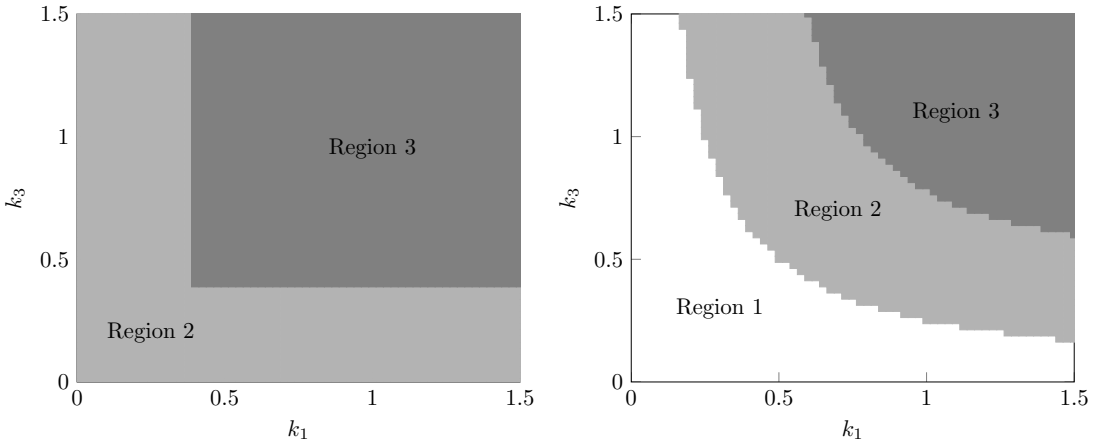

Figure 4.3: Left Panel: cluster synchronization regions for Π_4. Right panel: cluster synchronization regions for Π_5, Π_6, Π_7 and Π_8. Region 1 corresponds to parameter configurations for which neither Theorem 3.2.2 nor the modified sufficient condition guarantee cluster synchronization. In region 2, the modified sufficient condition guarantees cluster synchronization and in region 3, both Theorem 3.2.2 and the modified sufficient condition guarantee cluster synchronization.

The orbit partitions of Π_2 and Π_3 are

$$\pi_{\Pi_2} = \{\{1\},\{2\},\{3\},\{4\},\{5,6\},\{7,8\}\},$$
$$\pi_{\Pi_3} = \{\{1,2\},\{3,4\},\{5\},\{6\},\{7\},\{8\}\}.$$

According to Figure 4.2, the associated cluster synchronization manifolds are asymptotically stable for all values of k_1 and k_3. We can thus expect that nodes in the cells $\{1,2\},\{3,4\},\{5,6\}$ and $\{7,8\}$ will *always synchronize*. Note however, that the orbit partition of Π_4 is

$$\pi_{\Pi_4} = \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\}$$

and cluster synchronization for this symmetry is guaranteed only for larger values of k_1 and k_3 (see Figure 4.3). Furthermore, if we consider the orbit partitions of $\Pi_4, \ldots, \Pi_8$, given by

$$\pi_{\Pi_4} = \{\{1,2\},\{3,4\},\{5,6\},\{7,8\}\},$$
$$\pi_{\Pi_5} = \{\{1,3\},\{2,4\},\{5,7\},\{6,8\}\},$$
$$\pi_{\Pi_6} = \{\{1,3\},\{2,4\},\{5,8\},\{6,7\}\},$$
$$\pi_{\Pi_7} = \{\{1,4\},\{2,3\},\{5,7\},\{6,8\}\},$$
$$\pi_{\Pi_8} = \{\{1,4\},\{2,3\},\{5,8\},\{6,7\}\},$$

which all have the same region of guaranteed stability as shown in Figure 4.3, then we must conclude that nodes in the cells $\{1,2,3,4\}$ and $\{5,6,7,8\}$ synchronize as well. The existence of the associated linear invariant manifold is not explicitly predicted by our analysis of the symmetries. However, the approach of constructing Laplacian orbit partitions from symmetry orbit partitions as reviewed at the end of section 3.1 should prove useful in this regard and might be an interesting starting point for future research.

4.2.3 Convergence Rates

We established that Theorem 3.2.2 and the modified sufficient condition separate the symmetries into a number of groups according to their regions of guaranteed stability. We now return to the idea that in a model reduction setting, we would like to cluster together nodes that synchronize quickly.

Not only was demonstrated that the modified sufficient condition, as discussed in section 3.3, is less restrictive than Theorem 3.2.2 in terms of the regions of guaranteed cluster synchronization: Lemma 3.3.3 also gives us the means to estimate the convergence rate of the Lyapunov function which is employed to prove stability. This estimate of the convergence rate is an upper bound but it may be precise enough to say something about the convergence of trajectories as they approach the linear invariant manifolds. If there are significant differences in these convergence rates, the symmetries associated with quicker convergence might be better candidates for clustering.

To investigate whether there actually are meaningful differences between the symmetries, we study trajectories by means of straightforward integration of the system. We then approximate the rates at which trajectories converge to the symmetry associated linear invariant manifolds and compare them with the upper bounds as given by Lemma 3.3.3.

Experimental Setup

Given an initial condition x_0, we will denote the associated solution of

$$\dot{x} = F(x) - (\mathcal{L}_P \otimes BC)x, \quad x(0) = x_0,$$

as $x(t; x_0)$. We are interested in the distance between $x(t; x_0)$ and the subspace $\ker(I_{Nn} - \Pi \otimes I_n)$. Following the approach from chapter 3, we characterize this distance by means of the Lyapunov function

$$V(x) = \frac{1}{2\alpha} z^\top M z + \frac{1}{2} y^\top M y,$$

with $x = \mathrm{col}(z, y)$ and $M = (I_N - \Pi)^\top (I_N - \Pi)$. Based on this Lyapunov function, estimates for the convergence rate at which trajectories approach the associated subspaces have been derived. We will now investigate the convergence rate by means of straightforward integration and the inspection of plots for $V(x)$.

The distance of a trajectory $x(t; x_0)$ to the subspace associated with the symmetry Π is now characterized by $V(x(t; x_0))$. Of course, which subspace a given trajectory tends towards is also dependent on the initial condition. To still be able to say something about the overall convergence rates, rather than considering a single trajectory, we consider many and average over the number of initial conditions. Given a set of random initial conditions x_0^i, $i = 1, \ldots, s$, we introduce the quantity

$$d(t; \Pi) = \frac{1}{s} \sum_{i=1}^{s} V(x(t; x_0^i)).$$

In the case of full synchronization, we will observe $d(t; \Pi) \to 0$ as $t \to \infty$ for all permutation matrices Π, however, in principle the overall rate at which $d(t; \Pi) \to 0$ is expected to vary for

different choices of Π. How large we choose the individual parameters $k_1, \ldots, k_4$ will influence the converge rates. For example, by means of inspection of the network (see Figure 2.6), if we pick k_1 very large and the other parameters very small, we would expect the trajectories to quickly converge to the subspaces associated with synchronization involving nodes 1, 2, 3 and 4. To account for this, we again consider Pogromsky's network with maximal symmetry, i.e. with $k_1 = k_2 = k_3 = k_4 = k$. This yields 48 symmetries in total and from these, we can derive a set of 32 symmetries $\Pi_1, \ldots, \Pi_{32}$, which have mutually distinct characteristic matrices associated with them. For each of the symmetries, we plot $d(t; \Pi)$ with $s = 1000$ for $t \in [0, 100]$ and $k = 0.1, 0.2, 0.3, 0.4$. The results are summarized in Figure 4.4.

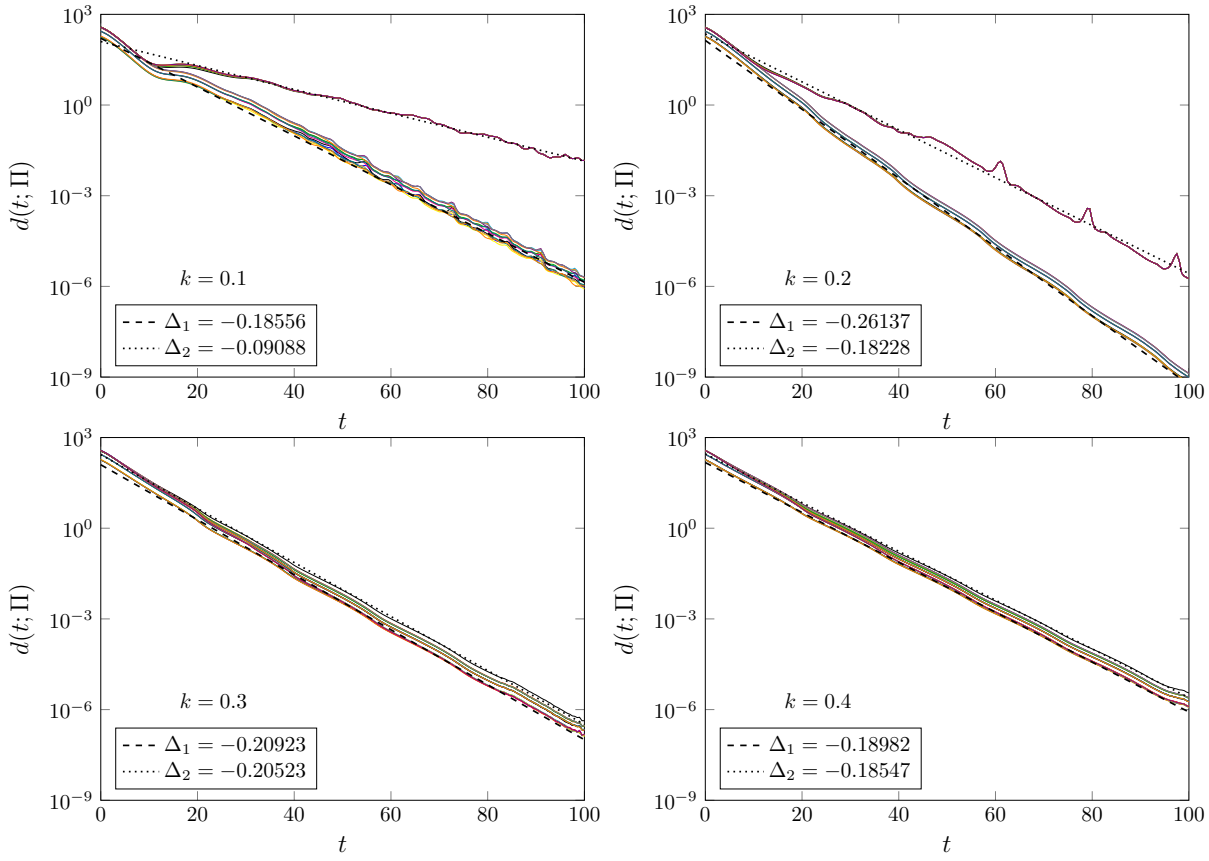

Figure 4.4: Each plot has the time t on the abscissa and the quantity $d(t; \Pi)$ on the ordinate. Initial conditions have been sampled uniformly such that $|z_j(0)| \leq 1$ and $|y_j(0)| \leq 2$ for all $j = 1, ..., 8$. The quantity $d(t; \Pi)$ is then computed with $s = 1000$ independently sampled initial conditions. Plotted is $d(t; \Pi_i)$ for the previously discussed 32 symmetries $\Pi_1, \ldots, \Pi_{32}$ of Pogromsky's network with all parameters equal. Each coloured solid line corresponds to $d(t; \Pi_i)$ for a different symmetry. A logarithmic scale for the ordinate is used in each plot. Read from the top left to the bottom right, the parameters are $k = 0.1, 0.2, 0.3$ and $k = 0.4$. In each plot, we use a linear fit to approximate the average slope of the curve with the highest and smallest value $d(100; \Pi_i)$. These linear fits correspond to the black and white dotted/dashed lines. The slopes of these lines are given by Δ_1 and Δ_2.

In Figure 4.4, we see that the symmetries again split into two groups. The curves appear to be roughly linear and the ones associated with symmetries from the same group have identical "overall slopes". Since we are using a logarithmically scaled ordinate, the slopes of the "lines" indicate the rate of exponential decay of $d(t; \Pi)$. For all selected values of k, one group of lines always has an overall smaller slope than the other group and the difference in slope decreases

as k increases from 0.1 to 0.4.

Following a similar argumentation as in the previous section, we now investigate the difference between symmetries belonging to the different groups in terms of cluster synchronization, if such a difference exists. Indeed, we find that the group with faster convergence is associated with the synchronization of nodes in the cells $\{1, 2, 5, 6\}$ and $\{3, 4, 7, 8\}$, while the other group contains two symmetries that respectively have orbit partitions

$$\{\{1, 4, 5, 6, 7, 8\}, \{2, 3\}\},$$
$$\{\{1, 2\}, \{3, 4, 5, 6\}, \{7, 8\}\}.$$

As cluster synchronization associated with the these two orbit partitions implies full synchronization, we conclude that the second, less steep group of lines is associated with full synchronization. Let us now compare these observations with the upper bounds as predicted by Lemma 3.3.3.

We select the two permutation matrices $\Pi_{\text{fast}} := \Pi_{25}$ and $\Pi_{\text{slow}} := \Pi_{32}$ to represent the two groups of symmetries and compare the observed approximations of the convergence rates Δ_1 and Δ_2 to the upper bounds δ_{eff} as predicted by Lemma 3.3.3. The values are summarized in Table 4.1.

		$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$
Π_{fast}	Δ_1	-0.186	-0.261	-0.209	-0.190
	δ_{eff}	-	-	-0.128	-0.128
Π_{slow}	Δ_2	-0.091	-0.182	-0.205	-0.185
	δ_{eff}	-	-	-	-

Table 4.1: Observed approximate convergence rates and upper bounds.

As can be seen in Table 4.1, for $k = 0.1$ and $k = 0.2$ the modified criterion cannot guarantee stability for symmetries of either group. For $k = 0.3$ and $k = 0.4$ the situation is different: the values of δ_{eff} for Π_{fast} are negative and equal to $-2\alpha\beta = -0.128$. This means that the criterion does guarantee stability and the convergence rate is bounded above by $-2\alpha\beta$. This is in line with the observed approximated convergence rates Δ_1. For Π_{slow}, the criterion still does not guarantee stability and so there is also no upper bound on the convergence rate listed. We can draw two conclusions from these observations:

- The proposed criterion giving an upper bound on the convergence rate is conservative: while clearly trajectories converge to the associated linear invariant manifolds for all values of k, the method predicts convergence only for the group associated with faster convergence and this only for $k = 0.3$ and $k = 0.4$.

- The proposed criterion characterizes the qualitative difference between the two groups of symmetries correctly: for $k = 0.3$ and $k = 0.4$ the method predicts convergence for Π_{fast} and not for Π_{slow}.

We move on to apply the same type of analysis to Steur's network.

4.3 Steur's Network

In addition to the discussion of cluster synchronization regions and associated convergence rates, we take a closer look at the modular travelling wave solution in Steur's network. As before, we start by checking conditions 4 and 5 of Theorem 3.2.2.

Conditions 4 and 5

Recall that Steur's network is a directed graph with $N = \bar{M}\bar{N}$ nodes and that the associated Laplacian matrix is given by

$$\mathcal{L}_S(\sigma, \mu) = \sigma \mathcal{L}_{\text{ring}} \otimes B_{\mathcal{L}} + \mu I_{\bar{N}} \otimes \mathcal{L}_{\text{module}}.$$

Condition 4 of Theorem 3.2.2 requires that $\mathcal{L}_S(\sigma, \mu) + \mathcal{L}_S(\sigma, \mu)^\top$ be positive semi-definite but since Steur's network is directed, it is not immediately obvious that this is indeed the case.

Lemma 4.3.1. *The matrix $\mathcal{L}_S(\sigma, \mu) + \mathcal{L}_S(\sigma, \mu)^\top$ is positive semi-definite for $\sigma, \mu \geq 0$.*

The proof of this lemma can be found in Appendix A.

By Lemma 4.3.1 it follows that condition 4 is met for any $\sigma, \mu \geq 0$. As before, we restrict the permutation matrices under consideration to the symmetries of the network. Since all symmetries satisfy $\Pi\mathcal{L} = \mathcal{L}\Pi$, condition 5 is met as well. We continue with an exposition of the symmetries present in Steur's network.

4.3.1 Symmetries

Since the number of nodes in Steur's network is variable, we should think about how many symmetries there are, given the number of nodes per module $\bar{M}$ and the number of modules $\bar{N}$. To arrive at the general expression for the number of symmetries in Steur's network, we consider symmetries stemming from the individual modules and symmetries stemming from the directed ring structure.

Each module contains $\bar{M}$ nodes and only one of these nodes is connected to the previous and the following module. This means that for each module there are $(\bar{M} - 1)!$ symmetries which simply stem from the complete subgraph contained in each module. Since there are $\bar{N}$ modules, this yields $[(\bar{M} - 1)!\,]^{\bar{N}}$ symmetries. Now we need to account for the fact that the modules are connected to one another in a directed ring structure. The *undirected ring* with $\bar{N}$ nodes can be thought of as the regular $\bar{N}$-gon and so the associated group of symmetries is the dihedral group containing $2\bar{N}$ elements. This means that the *directed ring* has half as many, i.e. $\bar{N}$, symmetries. Combining the number of symmetries stemming from the modules together with the ones stemming from the directed ring structure, we conclude that Steur's network has

$$\bar{N}[(\bar{M} - 1)!\,]^{\bar{N}}$$

symmetries in total.

As previously discussed, in general, the number of symmetries in a graph is maximized if all edge/arc weights are identical and parameter configurations with nonidentical values will

generically yield less symmetries. Recall that this is indeed the case of Pogromsky's network. However, for Steur's network, this is not true: any choice of $\sigma > 0$ and $\mu > 0$ will yield the same number of symmetries.

For a number of choices for the network size parameters $\bar{M}$ and $\bar{N}$, the number of symmetries are listed in Table 4.2. Note that for $\bar{M} < 3$, the network loses its modular structure and for $\bar{N} < 3$, the network loses its ring structure. We thus only consider $\bar{M}, \bar{N} \geq 3$.

$\mu, \sigma > 0$	$\bar{N} = 3$	$\bar{N} = 4$	$\bar{N} = 5$
$\bar{M} = 3$	24	64	160
$\bar{M} = 4$	648	5184	38880
$\bar{M} = 5$	41472	1327104	39813120

Table 4.2: Number of symmetries of Steur's network for a selection of values $\bar{M}$ and $\bar{N}$.

Involving the factorial of $\bar{M} - 1$, it is clear that even for relatively small $\bar{M}$ and $\bar{N}$, Steur's network has too many symmetries for us to investigate each one of them. However, it is not necessary to study all symmetries individually: We are interested in the orbit partitions of the permutation matrices associated with the symmetries. If two permutation matrices Π_1 and Π_2 have identical orbit partitions, i.e. $\pi_{\Pi_1} = \pi_{\Pi_2}$, we will consider them equivalent. Having computed all symmetries of a graph, our next step is then to compute a subset such that the associated set of orbit partitions does not contain any duplicates. Using a brute force method, the number of symmetries with mutually distinct orbit partitions has been computed again for the same values of $\bar{M}$ and $\bar{N}$. However, due to excessive run time for larger values of $\bar{M}$ and $\bar{N}$, the number of symmetries with mutually distinct orbit partitions has been computed only for the values as indicated in Table 4.3.

$\mu, \sigma > 0$	$\bar{N} = 3$	$\bar{N} = 4$	$\bar{N} = 5$
$\bar{M} = 3$	13	34	49
$\bar{M} = 4$	189	1179	-
$\bar{M} = 5$	-	-	-

Table 4.3: Number of symmetries of Steur's network with mutually distinct orbit partitions for a selection of values $\bar{M}$ and $\bar{N}$. Run times ranged from 0.23 seconds for $\bar{M} = \bar{N} = 3$ to 256 seconds for $\bar{M} = \bar{N} = 4$.

For simplicity, we will restrict our study of the symmetries to the simplest nontrivial case $\bar{M} = \bar{N} = 3$. Steur's network then has 24 symmetries which can be reduced to a set of 13 with mutually distinct orbit partitions, as can be read from Tables 4.2 and 4.3. We will denote the associated permutation matrices $\Pi_1, \Pi_2, \ldots, \Pi_{13}$. We will now have a closer look at these symmetries.

A comprehensive listing of all 13 symmetries together with visual representations of the associated reduced-order models can be found in Appendix B. We study a selected few of these 13 symmetries in more detail: The first symmetry is simply the trivial symmetry $\Pi_1 = I_9$ and no clustering of nodes takes place. The fourth symmetry and the associated characteristic matrix

are given by

$$\Pi_4 = \begin{bmatrix} 1 & 0 & 0 & & \\ 0 & 0 & 1 & & \\ 0 & 1 & 0 & & \\ & & & I_6 & \end{bmatrix}, \quad P(\pi_{\Pi_4}) = \begin{bmatrix} 1 & 0 & \\ 0 & 1 & \\ 0 & 1 & \\ & & I_6 \end{bmatrix}.$$

The Laplacian matrix of the resulting reduced network is given by

$$\bar{\mathcal{L}}_S = \begin{bmatrix} 3 & -2 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 3 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}.$$

The symmetry Π_4 corresponds to transposing nodes 2 and 3. Clustering according to the orbit partition of this symmetry would then lead to the reduced network as illustrated in Figure 4.5.

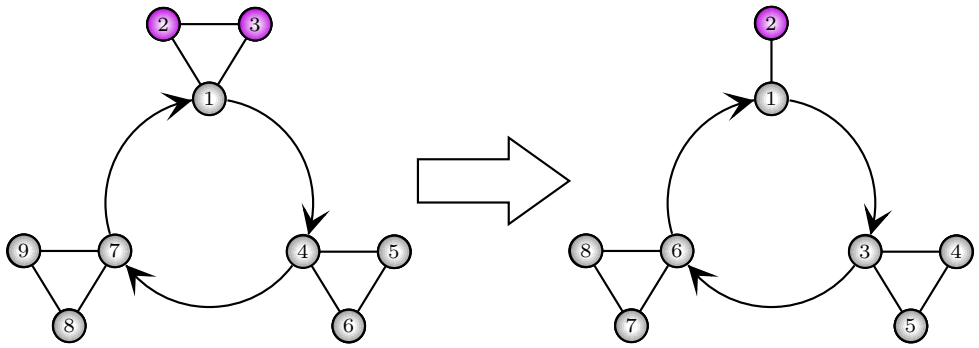

Figure 4.5: Model reduction when clustering according to the orbit partition of Π_4. Nodes of the same colour in the original network are clustered together in the reduced network.

The 9th symmetry and the associated characteristic matrix are given by

$$\Pi_9 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad P(\pi_{\Pi_9}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The Laplacian matrix of the resulting reduced network is given by

$$\bar{\mathcal{L}}_S = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}.$$

Symmetry Π_9 corresponds to moving all nodes 6 positions further, i.e. node 1 goes to 7, node 2 to 8 and so on. This "rotational" symmetry stems from the directed ring character of the network.

Remark. Note that shifting all nodes by 3 positions is also a valid symmetry, however the associated characteristic matrix is the same as for the shift by 6 and so only one of them made it into the reduced set of symmetries.

The model reduction situation associated with Π_9 is illustrated in Figure 4.5.

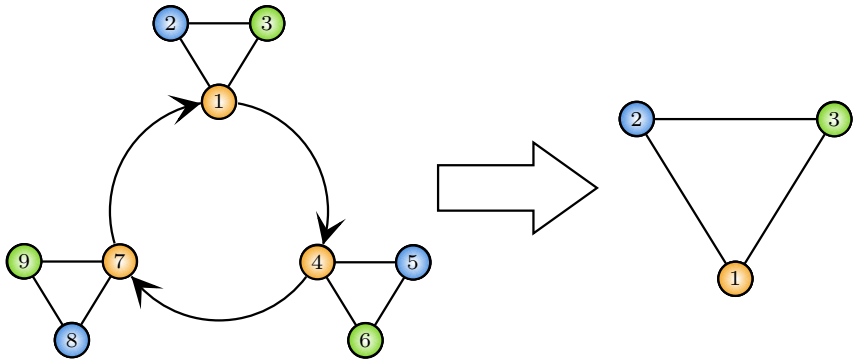

Figure 4.6: Model reduction when clustering according to the orbit partition of Π_9. Nodes of the same colour in the original network are clustered together in the reduced network.

The remaining symmetries can be understood as variations and/or combinations of symmetries Π_4 and Π_9. For example, Π_4 permutes the nodes in module 1 while Π_2 permutes the nodes in module 3. Similarly, Π_{13} corresponds to permuting nodes in the modules while also rotating the whole network. The resulting reduced network only has 2 nodes: one representing the nodes in the directed ring and one representing the nodes in all the modules (see Appendix B). Table 4.4 contains an overview of the symmetries of Steur's network together with a qualitative description of the associated clustering. We move on to study their associated cluster synchronization regions.

Symmetry	**Description of clustering**
$\Pi_1 = I_9$	no clustering
Π_2, Π_3, Π_4	nodes in individual module, single module
Π_5, Π_6, Π_7	nodes in individual module, two modules
Π_8	nodes in individual module, three modules
$\Pi_9, \Pi_{10}, \Pi_{11}, \Pi_{12}$	nodes with same relative position in module
Π_{13}	nodes in directed cycle vs. rest

Table 4.4: Summary of types of symmetries in Steur's network and what kind of clustering they correspond to.

4.3.2 Stability Regions

We perform the same numerical experiment as previously for Pogromsky's network. Note that in order to make use of Theorem 3.2.2, we again need to obtain $\bar{\lambda}$. Similarly as before, the upper bound on C_1, as suggested by Corollary 4.1.1, seems unreasonably conservative and under consideration of numerical simulations, we try $C_1 = 1$ which yields $\bar{\lambda} = \frac{1}{2} + \frac{1}{\beta}$. We will see that also in this case, this can still be regarded as a conservative estimate.

Experimental Setup

We create a grid for a portion of the parameter plane with $\sigma = 0, 0.15, 0.3, ..., 10.05$ and $\mu = 0, 0.15, 0.3, ..., 10.05$ and check whether Theorem 3.2.2 or the modified sufficient condition guarantee cluster synchronization at any pair (σ, μ). In this way, we estimate the region in which cluster synchronization is guaranteed. The results are summarized in Figures 4.7 and 4.8.

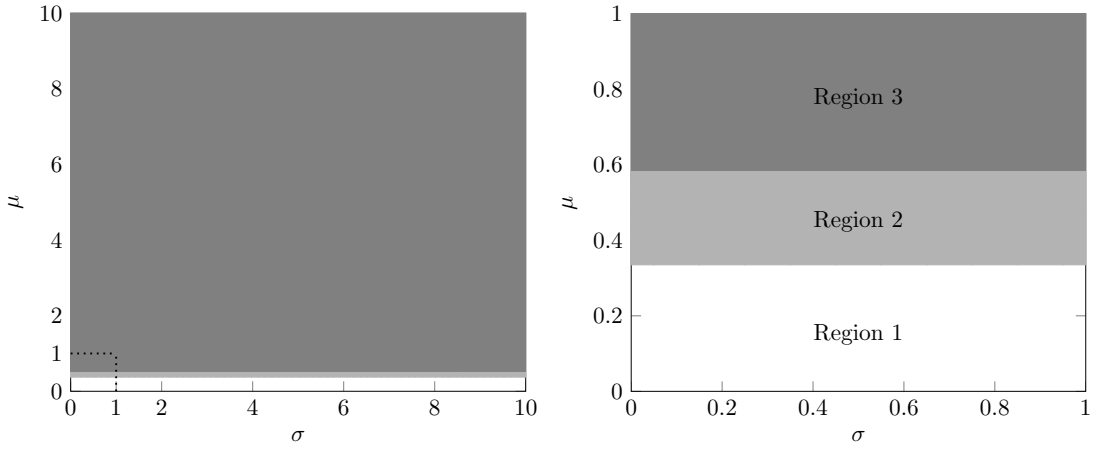

Figure 4.7: Cluster synchronization regions for $\Pi_2, \Pi_3, \ldots, \Pi_8$. Left panel: Synchronization regions for $(\sigma, \mu) \in [0, 10] \times [0, 10]$. At this scale, the regions are hard to differentiate. The dotted lines indicate the zoom window ($[0, 1] \times [0, 1]$) which is depicted in the plot of the right panel. Right panel: Zoomed version of the plot in the left panel. Region 1 corresponds to parameter configurations for which neither Theorem 3.2.2 nor the modified sufficient condition guarantee cluster synchronization. In region 2, the modified sufficient condition guarantees cluster synchronization and in region 3, both Theorem 3.2.2 and the modified sufficient condition guarantee cluster synchronization.

Let us make a few observations. In all cases, the modified sufficient conditions yields less conservative results, just like for Pogromsky's network. So again, we have indication that the modified sufficient condition is less restrictive. Further, there are two groups of symmetries for which the associated cluster synchronization regions are identical namely $\Pi_2, \Pi_3, ..., \Pi_8$ (group 1) and $\Pi_9, \Pi_{10}, ..., \Pi_{13}$ (group 2).

In the case of the first group, the region of guaranteed cluster synchronization is bounded by a line of constant μ. This is plausible since all symmetries in group 1 are associated with synchronization of nodes *within* the individual modules and coupling in modules is governed by μ. In fact, for parameter values in this region, the nodes in the cells

$$\{2, 3\}, \{5, 6\} \text{ and } \{8, 9\}$$

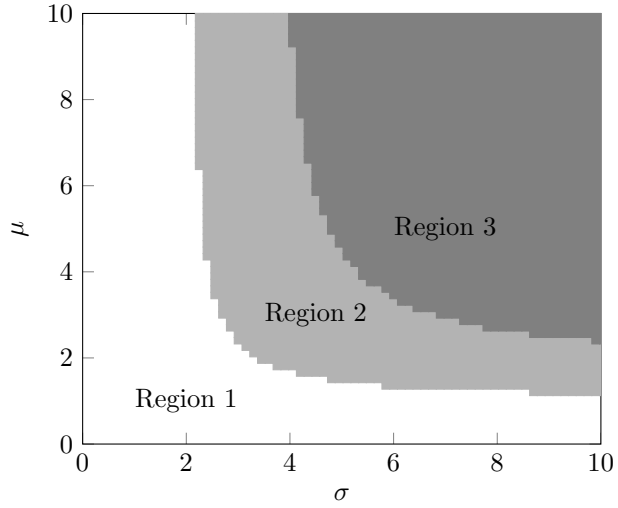

Figure 4.8: Cluster synchronization regions for $\Pi_9, \Pi_{10}, \ldots, \Pi_{13}$. Region 1 corresponds to parameter configurations for which neither Theorem 3.2.2 nor the modified sufficient condition guarantee cluster synchronization. In region 2, the modified sufficient condition guarantees cluster synchronization and in region 3, both Theorem 3.2.2 and the modified sufficient condition guarantee cluster synchronization.

are guaranteed to synchronize. Indeed, the orbit partition of Π_8 is

$$\pi_{\Pi_8} = \{\{2,3\},\{5,6\},\{8,9\}\},$$

which can be thought of as the "coarsest" orbit partition of the first group.

For the second group, sufficiently large values of σ and μ guarantee synchronization of the nodes 1, 4 and 7 which are the nodes making up the directed ring. However, the region of guaranteed cluster synchronization is contained in the one for the first group and so all nodes within the modules are also guaranteed to synchronize. This means that the nodes in the cells

$$\{1,4,7\},\{2,3,5,6,8,9\}$$

are guaranteed to synchronize. Indeed, the orbit partition of Π_{13} is

$$\pi_{\Pi_{13}} = \{\{1,4,7\},\{2,3,5,6,8,9\}\},$$

which can be thought of as the "coarsest" orbit partition of the second group. We continue our investigation of these two groups of symmetries in the next section.

4.3.3 Convergence Rates

We repeat the experiment as described in section 4.2.3, this time for Steur's network. For $\sigma = \mu$ taking values 0.4, 0.6, 0.8 and 1, we plot the quantity $d(t;\Pi)$ for the 13 symmetries of Steur's network with $\bar{M} = \bar{N} = 3$.

As can be seen from Figure 4.9, again two groups of symmetries appear. The first group with faster overall convergence consists of the permutation matrices $\Pi_2, \ldots, \Pi_8$ which act on the nodes in the modules only. The second group with slower overall convergence is $\Pi_9, \ldots, \Pi_{13}$. We again see the same split into two groups as in the analysis of the regions of guaranteed

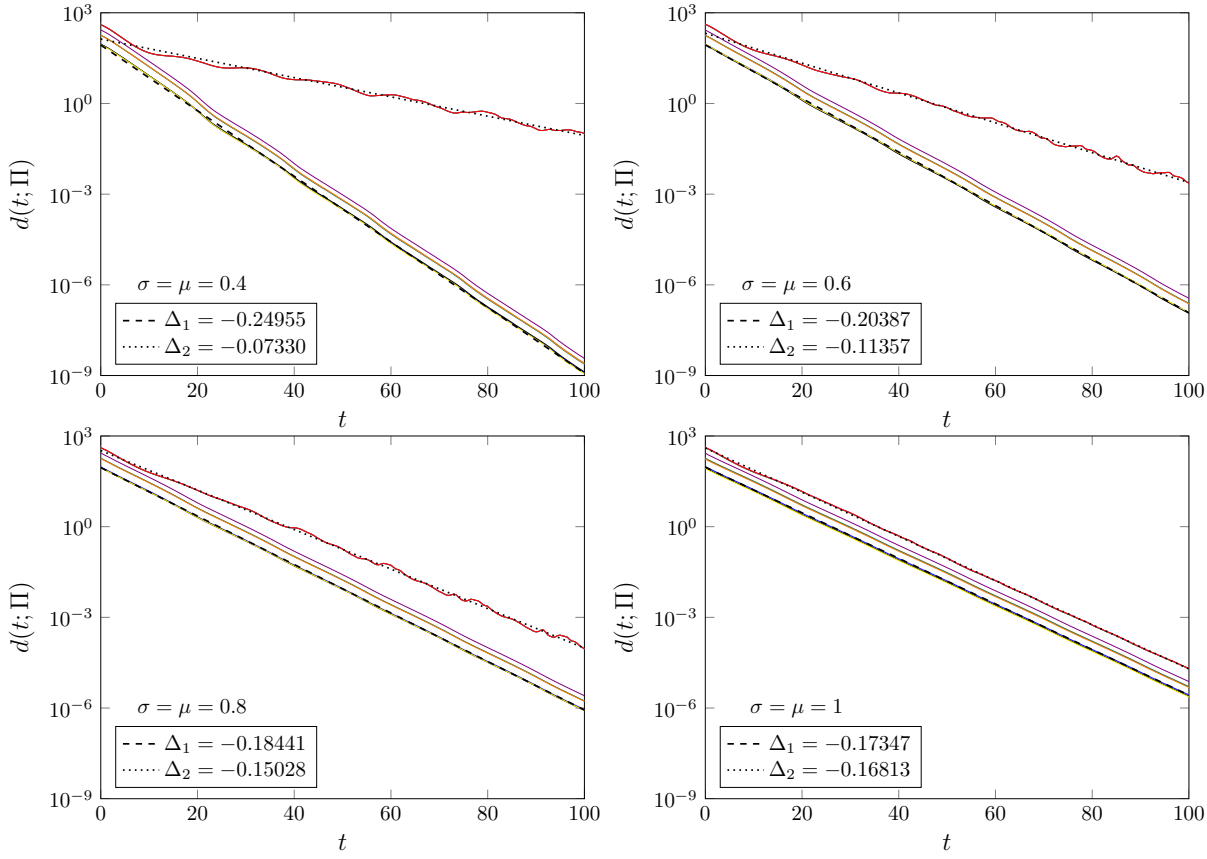

Figure 4.9: Each plot has the time t on the abscissa and the quantity $d(t;\Pi)$ on the ordinate. Initial conditions have been sampled uniformly such that $|z_j(0)| \leq 1$ and $|y_j(0)| \leq 2$ for all $j = 1, ..., 9$. The quantity $d(t;\Pi)$ is then computed with $s = 1000$ independently sampled initial conditions. Plotted is $d(t;\Pi_i)$ for the previously discussed 13 symmetries $\Pi_1, \ldots, \Pi_{13}$ of Steur's network. Each coloured solid line corresponds to $d(t;\Pi_i)$ for a different symmetry. A logarithmic scale for the ordinate is used in each plot. Read from the top left to the bottom right, the parameters are $\sigma = \mu = 0.4, 0.6, 0.8$ and $\sigma = \mu = 1$. In each plot, we use a linear fit to approximate the average slope of the curve with the highest and lowest value $d(100;\Pi_i)$. These linear fits correspond to the black and white dotted/dashed lines. The slopes of these lines are given by Δ_1 and Δ_2.

cluster synchronization. Let us see how the observed approximate convergence rates compare to the upper bounds given by Lemma 3.3.3. We consider $\Pi_{\text{fast}} := \Pi_8$ and $\Pi_{\text{slow}} := \Pi_{13}$ as representatives of the two groups. The values are summarized in Table 4.5.

		$\sigma = \mu = 0.4$	$\sigma = \mu = 0.6$	$\sigma = \mu = 0.8$	$\sigma = \mu = 1$
Π_{fast}	Δ_1	-0.250	-0.204	-0.184	-0.173
	δ_{eff}	-0.128	-0.128	-0.128	-0.128
Π_{slow}	Δ_2	-0.073	-0.113	-0.150	-0.168
	δ_{eff}	-	-	-	-

Table 4.5: Observed approximate convergence rates and upper bounds.

The predicted values of δ_{eff} in the row Π_{fast} are all negative and give valid upper bounds on the convergence rate. Moreover, we see that all values of δ_{eff} are equal to $-2\alpha\beta$ which is the minimal value for the upper bound which can be derived using Lemma 3.3.3.

For the second group, represented by Π_{slow}, the situation is again different: for no value of

the parameters σ and μ the modified sufficient condition guarantees stability, yet the trajectories converge towards the cluster synchronization manifolds. The conclusions we can draw from this case are the same as before: the proposed criterion is conservative, yet it characterizes the qualitative difference between the two groups of symmetries, which is visible in the observed convergence rate approximations Δ_1 and Δ_2, faithfully.

Having established that the modified criterion seems to make meaningful distinctions between groups of symmetries in terms of their observed convergence rates, we move on to investigate possible implications for clustering-based model reduction for Steur's network.

4.3.4 Stability of the Travelling Wave in Steur's Network

In the introduction of Steur's network in section 2.3, we noted that for $\bar{M} = \bar{N} = 5$ and $\sigma = \mu = 1$, the oscillators in the network either fully synchronize or only the oscillators in the modules synchronize and a modular travelling wave emerges. In both cases, the trajectories approach a periodic solution in state space. We refer to these solutions as *limit cycles*. If a limit cycle attracts nearby solutions, we say that it is *locally asymptotically stable*.

From our observations, we conclude that there are parameter values for which both limit cycles are locally asymptotically stable. The chosen initial condition determines to which limit cycle the associated trajectory will converge. In this case we speak of *multi-stability* as there are multiple attractors present.

Using the theory presented in chapter 3 and the observations made in the previous sections, we must conclude that the modular travelling wave is not always locally asymptotically stable: Theorem 3.2.2 implies that for sufficiently strong coupling, i.e. for sufficiently large values of σ and μ, the fully synchronous solution is *globally* asymptotically stable which means that, independent of the choice of initial condition, the oscillators will always synchronize. Conversely, this implies that the modular travelling wave loses its local stability somewhere. We proceed to investigate for which parameter values this is the case.

In the following analysis, we make use of the Poincaré map associated with a limit cycle and apply basic Floquet theory to determine a region of local asymptotic stability for the modular travelling wave. In case the reader is not familiar with these concepts, Appendix C contains a brief review.

In principle, if a limit cycle can be parameterized explicitly, it is possible to study the local stability "by hand". Using the Poincaré map we could compute the Floquet multipliers and their dependencies on the system parameters explicitly and simply check when such a multiplier leaves the unit circle. However, the FHN oscillator does not allow for such a paramterization. It is yet possible to carry out a local stability analysis: Numerical continuation software such as AUTO or MATCONT allow for the computation of Floquet multipliers and track them for varying system parameters when the explicit parametrization of limit cycles is impossible. For reasons of familiarity with MATLAB, we decide to use MATCONT for our analysis.

MATCONT allows for the continuation of limit cycles from a previously computed solution. This means that we can employ a method of trial and error with regard to coupling strengths σ and μ, as well as initial conditions to find a solution that converges to the modular travelling wave and then simply continue the associated limit cycle from there. To avoid unnecessary

complexity, we want to find the smallest values of $\bar{M}$ and $\bar{N}$ such that this trial and error method is feasible. As is demonstrated by Steur et al. [31], for $\bar{N} = 5$ and $\bar{M}$ between 2 and 10, a substantial proportion of solutions converge to a modular travelling wave, as detected in their experiments.

Indeed, for $\bar{N} = 5$, using our trial and error method, we easily find parameter configurations and initial conditions that converge to a modular travelling wave for values of $\bar{M}$ between 2 and 10.

Example 6. For $\bar{N} = 5$ and $\bar{M} = 3$ we find (for some suitable initial conditions) a modular travelling wave solution for $(\sigma, \mu) = (0.5, 1)$. Feeding this solution to MATCONT, we can now continue this limit cycle for varying σ and μ while tracking the Floquet multipliers of the limit cycle. It is to be noted that initially, all multipliers lie in the unit circle and so the limit cycle is locally stable.

Continuing the limit cycle for increasing σ, MATCONT indicates that around $(\sigma, \mu) = (0.7, 1)$, a conjugate pair of multipliers lies on the unit circle, marking a Neimark-Sacker bifurcation. At this point, the modular travelling wave loses its local stability. MATCONT then allows to continue this Neimark-Sacker point throughout the parameter plane which yields a smooth curve. At each point of this curve, a Neimark-Sacker bifurcation occurs and the limit cycle associated with the modular travelling wave loses its local stability. Effectively, this curve is the boundary of the region of local stability for the modular travelling wave.

We obtain the curve of Neimark-Sacker bifurcations, as described in Example 6, for $\bar{N} = 5$ and $\bar{M} = 2, 3, 4, 5$ and plot the curves together with the regions of guaranteed stability, as obtained from the modified sufficient condition. The results are summarized in Figure 4.10.

A few observations about Figure 4.10: In the "U"-shaped region enclosed by the Neimark-Sacker curves, the modular travelling wave is locally stable, while on the outside, only fully synchronous solutions have been observed. It is unknown if, besides the modular travelling wave and fully synchronous solution, there exist any more attractors in the system, however, none have become apparent in the course of this analysis.

In each case, the Neimark-Sacker curve is fully contained in region 2. This is the region for which the modified sufficient condition guarantees cluster synchronization of the nodes in the modules. For fixed $\bar{N} = 5$, as we increase $\bar{M}$ from 3 to 5, we see that the overall "U"-shape of the Neimark-Sacker curve stays the same, yet it appears to become somewhat wider. This widening also makes intuitive sense: as $\bar{M}$ increases, the modules become larger and so the relative influence of the inter-module coupling strength σ becomes less. Consequently, larger values of σ are required to attain full synchronization and the "U"-shaped stability regions become wider.

The Neimark-Sacker curves do not cross into region 3 in any case. This also makes sense because in region 3, the parameter configurations are such that the nodes within the directed ring are guaranteed to synchronize and a travelling wave could not possibly emerge.

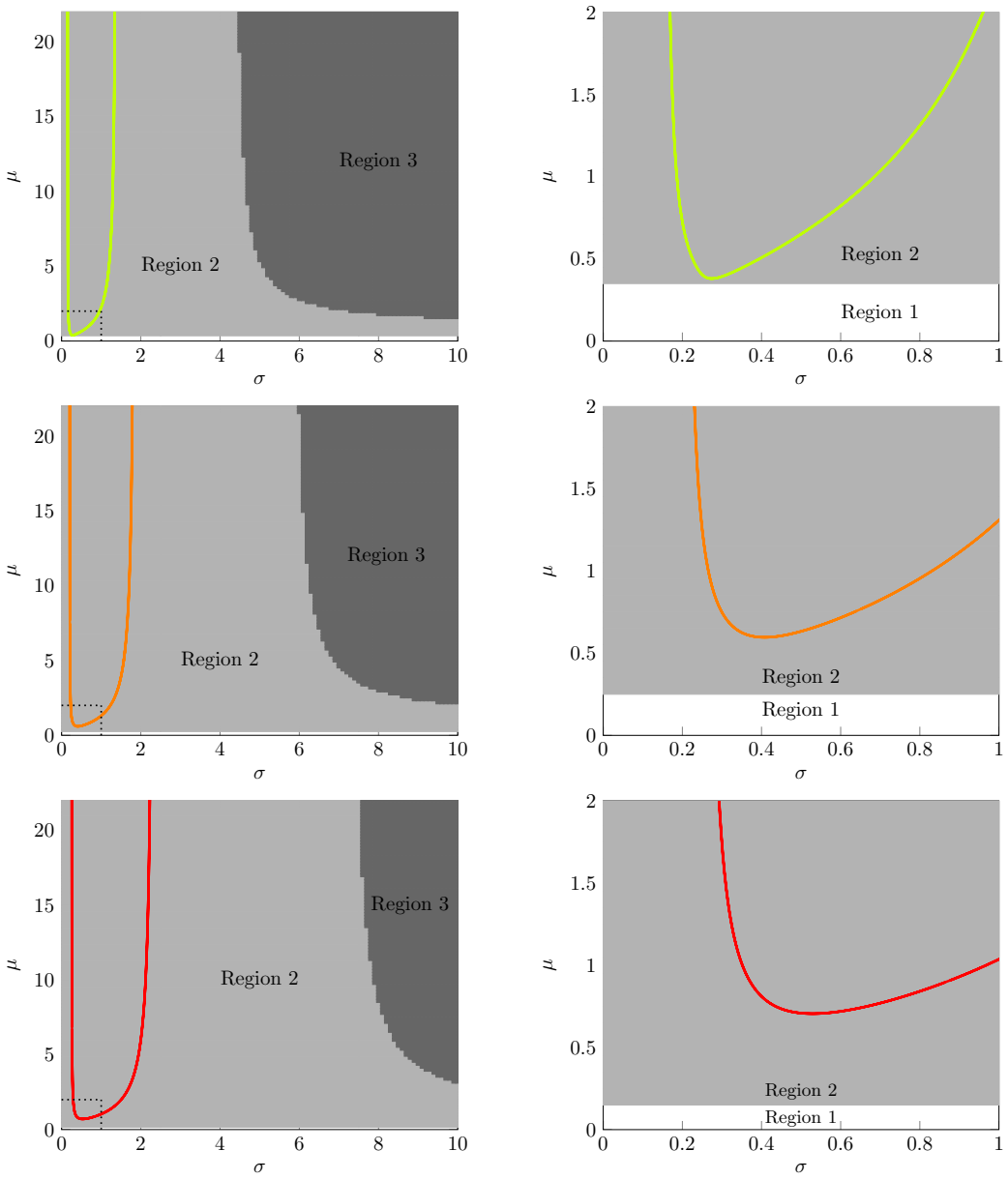

Figure 4.10: Plotted are the curves of Neimark-Sacker bifurcations for $\bar{N} = 5$ and $\bar{M} = 3, 4, 5$ (top to bottom, in green, orange and red respectively). Further, the regions of guaranteed cluster synchronization as derived from the modified sufficient condition are shown. In region 1, cluster synchronization is not guaranteed for any symmetry. In region 2, cluster synchronization of the nodes in the modules is guaranteed and in region 3, the nodes making up the directed ring synchronize and at the same time, the nodes in all across all modules synchronize. In each row, the plot on the right corresponds to the region marked by the dotted lines in the plot on the left.

What does this means in terms of clustering-based model reduction? To evaluate how well a reduced system approximates the original system, we will consider to what degree the region of local stability of the travelling wave translates faithfully from the original to the reduced-order model. Evidently, clustering the network according to any of the symmetries associated with region 3 ($\Pi_9, \ldots, \Pi_{13}$) would result in a reduced-order model with too little complexity in the sense that the directed ring structure would be destroyed and a "modular" travelling wave is impossible to attain. Not so for the symmetries associated with region 2: clustering according to any of the symmetries $\Pi_2, \ldots, \Pi_8$ would leave the directed ring structure intact. For these

symmetries, "modular" travelling waves should exist and there is hope that the region of local stability is similar to that of the original system.

To check if this is indeed the case, we once more consider Steur's network with $\bar{N} = 5$ and $\bar{M} = 2, 3, 4, 5$. Note that this time, we also include $\bar{M} = 2$. This we do since clustering together the nodes in the modules results in reduced networks that have the same topology as Steur's network with $\bar{M} = 2$. For each combination of $\bar{N}$ and $\bar{M}$, we then perform clustering-based model reduction according to Π_8. The stability region of the modular travelling wave is computed for each version of Steur's network and its associated reduced-order model. The results are summarized in Figure 4.11.

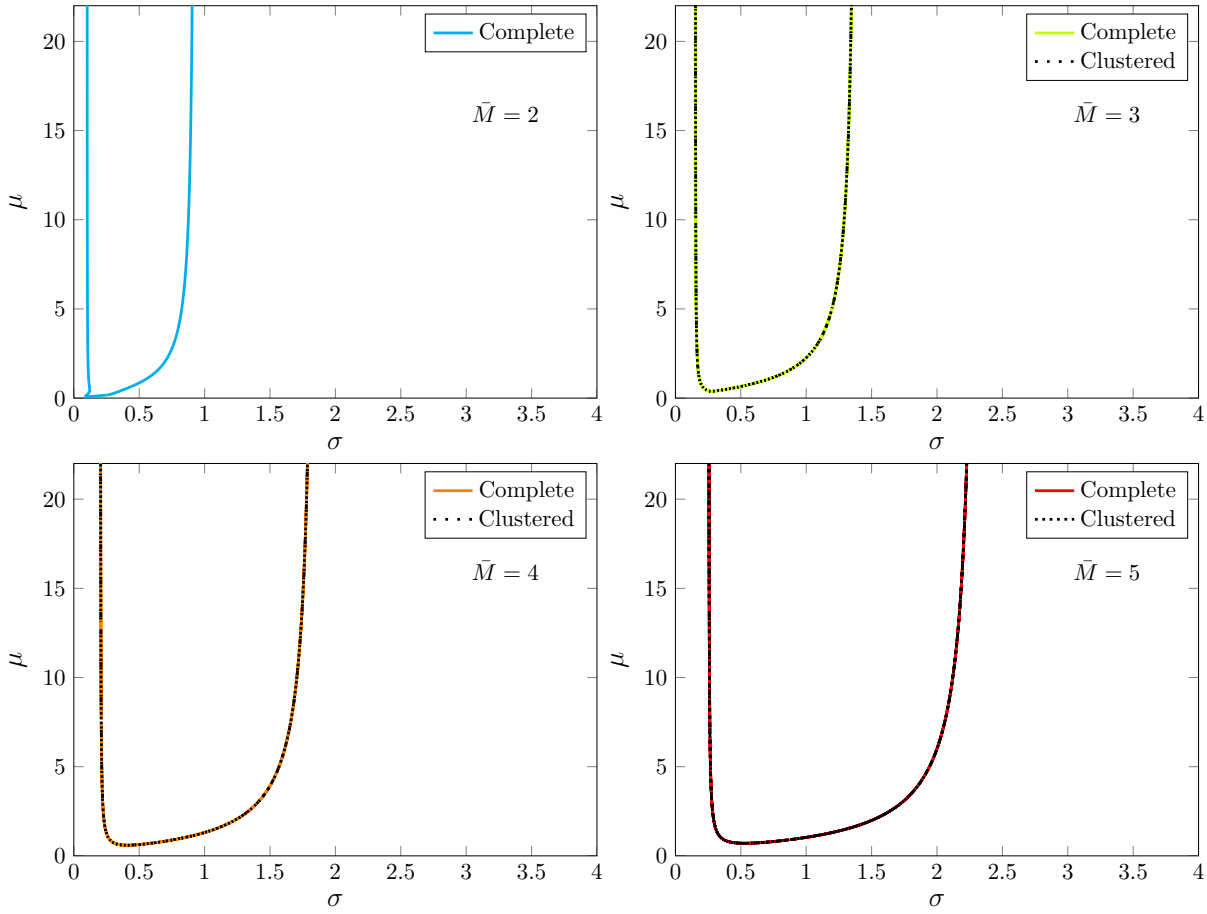

Figure 4.11: The Neimark-Sacker curves associated with the modular travelling wave solution for $\bar{N} = 5$ and $\bar{M} = 2, 3, 4, 5$ (top left to bottom right). In each plot, the Neimark-Sacker curve of the complete network is shown in colour, while the Neimark-Sacker curve for the clustered network is shown as a dotted line. Note that for $\bar{M} = 2$, the complete system and the clustered system are identical, thus only one curve is shown.

As can be seen in Figure 4.11, the local stability regions of the corresponding clustered systems are virtually identical to the ones of the complete systems. This indicates that the interaction of nodes within the modules has little to nothing to do with the emergence of "modular" travelling waves, which can be considered a characteristic emergent dynamical feature of the inter-modular directed ring structure. This poses the question if this is a persistent phenomenon that can be observed for arbitrary inter-modular network structures and associated characteristic emergent dynamical features. This issue will be addressed in more detail in the

discussion (see chapter 5).

What do these results mean in terms of the two groups of symmetries as discerned by Theorem 3.2.2 and Lemma 3.3.3? Theorem 3.2.2 and Lemma 3.3.3 can be used to make a distinction between the symmetries of a network system. This distinction boils down to which eigenvalue of the symmetrized graph Laplacian a group of symmetries is associated with. The larger the eigenvalue, the lower the required overall coupling strength which guarantees cluster synchronization (Theorem 3.2.2). As was demonstrated in Lemma 3.3.3, the association with these eigenvalues also impacts the convergence rates of trajectories approaching the associated cluster synchronization manifolds, albeit only provable in terms of upper bounds. However, as was demonstrated with the numerical experiments involving the approximations of these convergence rates by means of straightforward integration of the system, the group distinction actually impacts convergence rates directly: plotting the time-series for the involved Lyapunov functions, we saw that the group of symmetries which was identified by Theorem 3.2.2 and Lemma 3.3.3 as the "faster" group, actually did converge more quickly even though cluster synchronization could not be guaranteed by means of Theorem 3.2.2 or Lemma 3.3.3. We then saw that clustering according to symmetries of the "faster" group resulted in a well-approximated region of local stability for the modular travelling wave in Steur's network. This implies that the spectral properties of the symmetrized graph Laplacian might be a suitable tool to identify possible node clusterings which result in good reduced-order models.

4.4 Summary

In this chapter, we aimed to apply the insights about cluster synchronization from chapter 3 to the problem as stated in chapter 2. It was demonstrated that the FHN oscillator as well as the example networks satisfy the conditions of Theorem 3.2.2 and the derived modified sufficient condition. In order to make use of these results, we investigated the symmetries for both Pogromsky's network and Steur's network. We saw that the symmetries for both networks could be placed into two groups where each group has distinct features associated with cluster synchronization. This was reflected in the regions of guaranteed stability as well as the observed approximate convergence rates. Qualitative differences between the groups were predicted by Theorem 3.2.2 and Lemma 3.3.3 and are associated with the spectral properties of the symmetrized graph Laplacian. Finally, in the case of Steur's network, we studied the local stability region for the modular travelling wave using the numerical continuation software MATCONT. It was demonstrated that by clustering together nodes in the same modules, the local stability region is nearly perfectly preserved when applying clustering-based model order reduction.

Chapter 5

Conclusion and Discussion

Conclusion

In this thesis, we studied network systems of diffusively coupled oscillators and, in particular, model order reduction by means of node clustering. We addressed the question whether there are any natural choices for nodes that may be clustered together such that the reduced-order model has similar dynamical features as the original one. Intuition suggests to cluster together nodes which tend to synchronize quickly in the original model and we investigated whether this intuition can be backed by more formal argumentation.

To identify groupings of nodes that synchronize with one another in the original model, we studied the symmetries of the graph underlying the network system. Orbit partitions associated with these symmetries function as a starting point to determine groups of nodes that synchronize faster than others. In particular, we reviewed a result by Pogromsky [22, 23] that divides the set of symmetries into groups according to how strong the overall coupling needs to be, such that cluster synchronization with respect to the associated partitions is ensured. For two example networks, it was then shown that in both cases, the symmetries separate into two groups. Further, it was demonstrated that the convergence rate of a certain Lyapunov function, evaluated along state trajectories, is again related to the underlying symmetry and in particular, which group this symmetry belongs to. In this way, Pogromsky's result can be used to separate the symmetries in a network into a number of groups, each of which is associated with an eigenvalue of the symmetrized graph Laplacian. The larger the associated eigenvalue, the quicker the cluster synchronization.

In the case of a modular directed ring network, we saw that clustering according to symmetries from one group produced reduced-order models in which the local stability region of the modular travelling wave is well preserved. Clustering nodes according to any symmetry from the other group resulted in reduced-order models which do not even allow for the emergence of a travelling wave in the first place, not to speak of the preservation of the associated region of local stability. From this, we conclude that in the modular directed ring network, the travelling wave solution is distinctly associated with the directed ring structure: any symmetry-based clustering which destroys this structure leads to a reduced-order model in which this special solution does not exist. In terms of clustering-based model order reduction, this gives a practical approach on how to decide which nodes to cluster: given a network Laplacian, we find the symmetries, we

compute the associated eigenvalues and then cluster according to symmetries associated with the larger eigenvalues.

Discussion

The analysis as laid out in this thesis, with particular emphasis on the discussed result by Pogromsky [22, 23], suggests that the symmetries can be divided into groups where each group has certain cluster synchronization properties associated with it. These properties are:

1. Symmetries within the same group are associated with a certain threshold such that if the overall coupling exceeds this threshold, cluster synchronization sets in.

2. Convergence rates of trajectories, approaching the linear invariant manifolds associated with cluster synchronization, are identical for symmetries within the same group and differ between the groups.

Both these properties are related to the eigenvalues of the symmetrized Laplacian matrix and so it appears that, in network systems with diffusive coupling, the spectral properties of the symmetrized Laplacian matrix can be used to predict the outcome of clustering-based model reduction. However, this proposition comes with a number of caveats:

A central issue for this kind of analysis is the lack of a uniform method by which to compare the behaviour of the full-order model and the reduced-order model. While for input-output systems the performance of model order reduction can be quantified by means of comparing transfer functions, this is of course not possible when treating autonomous systems. For Steur's network, the modular travelling wave is a characteristic dynamical feature. Therefore, comparing how well its local stability region translates under model order reduction can be regarded as an appropriate means to measure performance. For an arbitrary network however, it is not clear what would constitute a "characteristic dynamical feature". In this sense, the analysis reduces to a case-by-case method. However, it would not be surprising if the existence of certain symmetries in a network is associated with the emergence of certain dynamical features in a more general setting. Repeating the kind of analysis as proposed in this thesis for a wider variety of networks might yield additional insight with regard to this issue. Another approach of comparing autonomous behaviour for simple ring networks is sketched out in Appendix E: Here, the relationship between the directed ring network and its undirected counterpart is explored.

Throughout this thesis, symmetries and their relationship to cluster synchronization have played a key role. Pogromsky's result introduces a natural hierarchy for groups of symmetries, however, it is not clear how this hierarchical structure relates to the group structure of the symmetries. An interesting starting point for further research in this direction might be to study this relationship, e.g. by means of finding generators of the symmetry group and then investigating the eigenvalues of the symmetrized Laplacian matrix which are associated with these generators.

A further limitation of this research is the restriction to the treatment of partitions which are associated with symmetries only. As has been discussed in section 3.1, there are more partitions that are associated with invariant synchronization manifolds. However, finding the symmetries

of a given network is a nontrivial problem in itself, not to mention the construction of additional valid cluster synchronization patterns. For smaller networks, finding partitions beyond those relating to symmetries should be feasible and the analysis as proposed in this thesis could yield more insight on the relationship between the different kinds of cluster synchronization patterns.

Finally, the requirement that the individual subsystems are identical and strictly semi-passive is a limitation as well. Another line of research, employing a contraction-theoretic approach to guarantee cluster synchronization, has been developed by Aminzare et al [1]: First described by Lohmiller & Slotine [13] and inspired by fluid-mechanical principles, contraction theory represents a slightly different take on stability analysis. In the classical sense, stability analysis is dealing with the convergence of trajectories to some reference motion or equilibrium point. However, to talk about stability, it is not necessary to know what this reference motion actually is. In this sense, contraction theory is concerned with the convergence of nearby trajectories rather than the convergence to some prescribed reference motion. This shift in perspective allows for a different mathematical approach to the problem of stability: rather than dealing with motion *integrals* (like in Lyapunov stability theory), contraction theory allows for a *differential* description, which simplifies the stability analysis in certain cases considerably.

Appendix A

Proofs

Proof of Lemma 3.1.1

Lemma. *Let $G = (V, E, A)$ be a graph, $\mathcal{L}$ be the associated Laplacian matrix and $\mathcal{G}$ be the symmetry group of G. Then for any $g \in \mathcal{G}$, it holds that*

$$\Pi_g \mathcal{L} = \mathcal{L} \Pi_g. \tag{A.1}$$

Proof. Recall that $\mathcal{L} = D - A$ where $D = \operatorname{diag}(d_1, \dots, d_N)$ with $d_i = \sum_{j=1}^{N} a_{ij}$ and A is the adjacency matrix. Let

$$\Pi_g = \begin{bmatrix} e_{g(1)} & \dots & e_{g(N)} \end{bmatrix},$$

then

$$\begin{aligned}
\Pi_g \mathcal{L} \Pi_g^\top &= \begin{bmatrix} e_{g(1)} & e_{g(2)} & \dots & e_{g(N)} \end{bmatrix} \begin{bmatrix} d_1 & -a_{12} & \dots & -a_{1N} \\ -a_{21} & d_2 & \dots & -a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{N1} & -a_{N2} & \dots & d_N \end{bmatrix} \begin{bmatrix} e_{g(1)}^\top \\ e_{g(2)}^\top \\ \vdots \\ e_{g(N)}^\top \end{bmatrix} \\
&= \begin{bmatrix} e_{g(1)} & e_{g(2)} & \dots & e_{g(N)} \end{bmatrix} \begin{bmatrix} d_1 e_{g(1)}^\top - \sum_{j \neq 1}^{N} a_{1j} e_{g(j)}^\top \\ d_2 e_{g(2)}^\top - \sum_{j \neq 2}^{N} a_{2j} e_{g(j)}^\top \\ \vdots \\ d_N e_{g(N)}^\top - \sum_{j \neq N}^{N} a_{Nj} e_{g(j)}^\top \end{bmatrix} \\
&= \sum_{i=1}^{N} e_{g(i)} \Big(d_i e_{g(i)}^\top - \sum_{j \neq i}^{N} a_{ij} e_{g(j)}^\top \Big) \\
&= \sum_{i=1}^{N} d_i e_{g(i)} e_{g(i)}^\top - \sum_{i=1}^{N} \sum_{j \neq i}^{N} a_{ij} e_{g(i)} e_{g(j)}^\top.
\end{aligned}$$

Now let $k = g(i)$ and $l = g(j)$. Since g is a permutation and thus a bijection, we have $i = g^{-1}(k)$ and $j = g^{-1}(l)$. Since $g \in \mathcal{G}$, we have that $a_{ij} = a_{g(i)g(j)}$ and hence $a_{ij} = a_{g^{-1}(i)g^{-1}(j)}$. It follows

that the row sums of $\mathcal{L}$ are invariant under the permutation g, i.e. $d_i = d_{g(i)}$ and thus

$$\begin{aligned}
\Pi_g \mathcal{L} \Pi_g^\top &= \sum_{i=1}^{N} d_i e_{g(i)} e_{g(i)}^\top - \sum_{i=1}^{N} \sum_{j \neq i}^{N} a_{ij} e_{g(i)} e_{g(j)}^\top \\
&= \sum_{k=1}^{N} d_{g^{-1}(k)} e_k e_k^\top - \sum_{k=1}^{N} \sum_{l \neq k}^{N} a_{g^{-1}(k) g^{-1}(l)} e_k e_l^\top \\
&= \sum_{k=1}^{N} d_k e_k e_k^\top - \sum_{k=1}^{N} \sum_{l \neq k}^{N} a_{kl} e_k e_l^\top \\
&= D - A = \mathcal{L},
\end{aligned}$$

which proves the lemma. ■

Proof of Lemma 3.3.1.

Lemma. *Let Π be an arbitrary permutation matrix. If R is diagonal and positive definite and Φ is (component-wise) monotonically increasing, then*

$$y^\top (I - \Pi)^\top R (I - \Pi) \Phi(y) \geq 0, \quad \forall y.$$

Proof. Let σ be the permutation associated with Π, i.e.

$$\Pi = [e_{\sigma(1)}, e_{\sigma(2)}, ..., e_{\sigma(N)}],$$

where e_i denotes the ith column of the identity matrix. Since $\Phi(y) = \mathrm{col}(\phi(y_1), \ldots, \phi(y_N))$, we have

$$(I - \Pi) y = \begin{bmatrix} y_1 - y_{\sigma(1)} \\ y_2 - y_{\sigma(2)} \\ \vdots \\ y_N - y_{\sigma(N)} \end{bmatrix}, \quad (I - \Pi)\Phi(y) = \begin{bmatrix} \phi(y_1) - \phi(y_{\sigma(1)}) \\ \phi(y_2) - \phi(y_{\sigma(2)}) \\ \vdots \\ \phi(y_N) - \phi(y_{\sigma(N)}) \end{bmatrix}.$$

So if R is a diagonal matrix with entries $r_i > 0$, we see that

$$y^\top (I - \Pi)^\top R (I - \Pi) \Phi(y) = \sum_{i=1}^{N} r_i (y_i - y_{\sigma(i)})(\phi(y_i) - \phi(y_{\sigma(i)})). \tag{A.2}$$

Moreover since Φ is monotonically increasing in its components, we have

$$\begin{aligned}
y_i - y_{\sigma(i)} > 0 &\implies \phi(y_i) - \phi(y_{\sigma(i)}) > 0, \\
y_i - y_{\sigma(i)} < 0 &\implies \phi(y_i) - \phi(y_{\sigma(i)}) < 0.
\end{aligned}$$

This means that all the terms in the sum from equation (A.2) are non-negative and hence

$$y^\top (I - \Pi)^\top R (I - \Pi) \Phi(y) \geq 0 \quad \forall y.$$

■

Proof of Lemma 3.3.2.

In the proof of Lemma 3.3.2, we make use of *Finsler's Lemma*.

Finsler's Lemma. *(See e.g. [5]) Let $x \in \mathbb{R}^n, Q \in \mathbb{S}^n$ and $B \in \mathbb{R}^{m\times n}$ such that rank$(B) < n$. The following statements are equivalent:*

1. $x^\top Q x < 0, \quad \forall Bx = 0, \quad x \neq 0.$
2. $(B^\perp)^\top Q B^\perp \prec 0.$
3. $\exists \rho \in \mathbb{R}$ *s.t.* $Q - \rho B^\top B \prec 0.$
4. $\exists X \in \mathbb{R}^{n\times m}$ *s.t.* $Q + XB + B^\top X^\top \prec 0.$

Lemma. *Let P be the characteristic matrix associated with the orbit partition of some $\Pi \neq I$ with $\mathcal{L}\Pi = \Pi\mathcal{L}$. Suppose there exists a $\rho \in \mathbb{R}$ such that*

$$2R - R\mathcal{L} - \mathcal{L}^\top R - \rho P P^\top \prec 0.$$

Then the matrix $M(I - \mathcal{L})$, with M as defined in equation (3.22), is negative definite.

Proof. Recall that

$$M = (I - \Pi)^\top R (I - \Pi).$$

Since $\Pi \neq I$, for the characteristic matrix of the associated orbit partition, we have $\text{rank}(P^\top) < N$. Assuming that

$$2R - R\mathcal{L} - \mathcal{L}^\top R - \rho P P^\top \prec 0,$$

and using the equivalence between the first and third statement in Finsler's Lemma, it follows that

$$\xi^\top (2R - R\mathcal{L} - \mathcal{L}^\top R))\xi < 0, \quad \forall \xi \in \ker(P^\top).$$

Noting that $\ker(P^\top) = \text{im}(I - \Pi)$, it follows that

$$\xi^\top (2R - R\mathcal{L} - \mathcal{L}^\top R))\xi < 0, \quad \forall \xi \in \text{im}(I - \Pi),$$

which can be rewritten to

$$y^\top (I - \Pi)^\top (2R - R\mathcal{L} - \mathcal{L}^\top R))(I - \Pi)y < 0, \quad \forall y.$$

Using that for arbitrary square matrices Q, we have $x^\top Q x = \frac{1}{2} x^\top (Q + Q^\top) x$, together with the assumption $\mathcal{L}\Pi = \Pi\mathcal{L}$, it follows that

$$y^\top M (I - \mathcal{L}) y < 0, \quad \forall y.$$

■

Proof of Lemma 3.3.3.

Lemma. *Let P be the characteristic matrix associated with the orbit partition of some $\Pi \neq I$ with $\mathcal{L}\Pi = \Pi\mathcal{L}$. If global asymptotical stability of the set $\mathcal{S}_\Pi$ can be shown by means of Lemma 3.3.2, then an upper bound for the convergence rate of the Lyapunov function, as defined in equation (3.21), along any trajectory is given by*

$$\delta_{\text{eff}} = \begin{cases} -2\alpha\beta\delta_0, & 0 < \delta_0 \leq 1, \\ -2\alpha\beta, & \delta_0 > 1, \end{cases}$$

where δ_0 is determined as

$$\delta_0 = \max\{\delta > 0 \,|\, (1+\alpha\beta\delta)I - \frac{1}{2}(\mathcal{L}+\mathcal{L}^\top) - \rho PP^\top \prec 0, \quad \rho \in \mathbb{R}\}. \tag{A.3}$$

Proof. The proof follows a similar line of reasoning as the proof for Lemma 3.3.2. Given the Lyapunov function candidate $V(x) = \frac{1}{2\alpha}z^\top Mz + \frac{1}{2}y^\top My$ with

$$M = (I_N - \Pi)^\top (I_N - \Pi),$$

it was shown that

$$\begin{aligned} \dot{V}(x) &= -\beta z^\top Mz + y^\top M(I-\mathcal{L})y - \gamma y^\top M\Phi(y) \\ &\leq -\beta z^\top Mz + y^\top M(I-\mathcal{L})y, \\ &= -2\alpha\beta\Big[\frac{1}{2\alpha}z^\top Mz - \frac{1}{2\alpha\beta}y^\top M(I-\mathcal{L})y\Big] \end{aligned}$$

where the inequality follows by Lemma 3.3.1. If we can find a $\delta > 0$ such that

$$y^\top M(I-\mathcal{L})y < -\alpha\beta\delta y^\top My, \quad \forall y, \tag{A.4}$$

then we can write

$$\dot{V}(x) \leq -2\alpha\beta\Big[\frac{1}{2\alpha}z^\top Mz + \frac{\delta}{2}y^\top My\Big].$$

Remark. At this point it becomes clear that for this approach to work, we cannot use M as defined in Lemma 3.3.2: we would end up with a term δR, where both δ and R are variable and the problem could not be formulated in terms of optimization with LMI constraints anymore.

Making a case distinction, we can now recover the Lyapunov function $V(x)$ on the RHS as follows:

Case 1: If $0 < \delta \leq 1$, then can we write

$$\begin{aligned} \dot{V}(x) &\leq -2\alpha\beta\Big[\frac{1}{2\alpha}z^\top Mz - \frac{\delta}{2\alpha}z^\top Mz + \frac{\delta}{2\alpha}z^\top Mz + \frac{\delta}{2}y^\top My\Big] \\ &= -2\alpha\beta\Big[\frac{1}{2\alpha}(1-\delta)z^\top Mz + \delta V(x)\Big] \\ &\leq -2\alpha\beta\delta V(x). \end{aligned}$$

Case 2: On the other hand, if $\delta > 1$, then

$$\begin{aligned}\dot{V}(x) &\leq -2\alpha\beta\Big[\frac{1}{2\alpha}z^\top Mz + \frac{1}{2}y^\top My - \frac{1}{2}y^\top My + \frac{\delta}{2}y^\top My\Big] \\ &= -2\alpha\beta\Big[V(x) + \frac{1}{2}(\delta-1)y^\top My\Big] \\ &\leq -2\alpha\beta V(x).\end{aligned}$$

In conclusion, this means that

$$\dot{V}(x) \leq \begin{cases} -2\alpha\beta\delta V(x), & 0 < \delta \leq 1, \\ -2\alpha\beta V(x), & \delta > 1. \end{cases}$$

Using Finsler's Lemma in the same way as in the proof of Lemma 3.3.2, we find that the maximal $\delta > 0$, such that inequality A.4 holds can be determined by means of maximization problem A.3. This proves the claim. ■

Proof of Lemma 4.1.1

Lemma (see Lemma 1, Steur et al. [31])**.** *The FHN oscillator is strictly semi-passive with respect to input u and output y with a radially unbounded storage function.*

Proof. By the definition of strict semi-passivity, we need to show that there exists a non-negative function $S : \mathbb{R}^2 \to \mathbb{R}$ such that, along the solutions of system (4.1), the function satisfies

$$\dot{S}(x,u) \leq y^\top u - H(x),$$

where $H : \mathbb{R}^2 \to \mathbb{R}$ has the property that there exists $r > 0$ and a continuous and non-negative function ρ, defined for $|x| \geq r$, such that

$$|x| \geq r \implies H(x) \geq \rho(x). \tag{A.5}$$

Let $S(z,y) = \frac{1}{2}(\alpha^{-1}z^2 + y^2)$ then $\dot{S} = yu - H(z,y)$ where

$$H(z,y) = \beta z^2 + \gamma y^4 - y^2.$$

Note that $S(z,y)$ is evidently radially unbounded. Writing

$$\begin{aligned}H(z,y) &= \beta z^2 + \gamma y^4 - y^2 + \beta y^2 - \beta y^2 \\ &= \beta(z^2+y^2) + \Big(\sqrt{\gamma}y^2 - \frac{\beta+1}{2\sqrt{\gamma}}\Big)^2 - \frac{(\beta+1)^2}{4\gamma},\end{aligned}$$

we see that $H(z,y) > 0$ for values of z and y outside the ball $z^2 + y^2 \leq \frac{(\beta+1)^2}{4\beta\gamma}$ which clearly shows that there exist a number $r > 0$ and some positive function ρ such that statement (A.5) holds. ■

Proof of Lemma 4.1.2

Lemma. *Assume the Laplacian matrix $\mathcal{L}$ associated with a diffusively coupled network of N subsystems is such that $\mathcal{L}+\mathcal{L}^\top$ is positive semi-definite. If the subsystems are FHN oscillators, then the region of absorption Ω for trajectories of the network system is contained in the ball of radius $R = \sqrt{2N}\frac{(\alpha\beta+1)}{\sqrt{4\alpha^2\beta\gamma}}$.*

Proof. As was shown in the proof of Lemma 4.1.1, the function

$$S(z_j, y_j) = \frac{1}{2}(\alpha^{-1}z_j^2 + y_j^2)$$

is a valid storage function for the FHN oscillator and $\dot{S} = y_j u_j - H(z_j, y_j)$ where

$$H(z_j, y_j) = \beta z_j^2 + \gamma y_j^4 - y_j^2.$$

Let $x_j = \mathrm{col}(z_j, y_j)$ and $x = \mathrm{col}(x_1, \ldots, x_N)$ as well as $y = \mathrm{col}(y_1, \ldots, y_N)$ and $u = \mathrm{col}(u_1, \ldots, u_N)$, and define

$$V(x) = \sum_{i=1}^{N} S(z_j, y_j).$$

Observe that

$$\frac{1}{2}||x||^2 \leq V(x) \leq \frac{1}{2\alpha}||x||^2,$$

since $\alpha < 1$. A direct computation shows that

$$\dot{V} := \frac{\partial V}{\partial x}(x)\dot{x} = -\sum_{i=1}^{N} H(z_j, y_j) - y^\top \mathcal{L} y,$$

where we use that $y^\top u = -y^\top \mathcal{L} y$. By assumption we have

$$y^\top \mathcal{L} y = \frac{1}{2} y^\top (\mathcal{L} + \mathcal{L}^\top) y \geq 0$$

and hence

$$\dot{V} \leq -\sum_{i=1}^{N} H(z_j, y_j).$$

Using a completion of the square, similar to how it was done in the proof of Lemma 4.1.1, we find

$$\begin{aligned}\dot{V} &\leq -\alpha\beta V + N\frac{(\alpha\beta+1)^2}{4\gamma} \\ &\leq -\alpha\beta\frac{1}{2}||x||^2 + N\frac{(\alpha\beta+1)^2}{4\gamma}.\end{aligned}$$

Requiring this RHS to be negative, we arrive at the following sufficient condition

$$||x|| > \sqrt{2N}\frac{(\alpha\beta+1)}{\sqrt{4\alpha\beta\gamma}} \implies \dot{V}(x) < 0.$$

By [11, Theorem 4.18], it follows that Ω must be contained in the ball with radius

$$R = \sqrt{2N}\frac{(\alpha\beta+1)}{\sqrt{4\alpha^2\beta\gamma}}.$$

■

Proof of Corollary 4.1.1

Corollary. *Assume all conditions of Theorem 3.2.2 are met. For the FHN oscillator, we have*

$$C_1 \leq \begin{cases} 1, & R \leq \sqrt{\frac{2}{\gamma}}, \\ \gamma R^2 - 1, & R > \sqrt{\frac{2}{\gamma}}. \end{cases}$$

Proof. Lemma 4.1.2 guarantees that trajectories of the network system enter the ball $\|x\| \leq R$ in finite time and then stay there. An upper bound for C_1 is thus given by

$$C_1 \leq \max_{(y_i,y_j)\in B_R} \left|1 - \gamma(y_i^2 + y_i y_j + y_j^2)\right|,$$

where $B_R = \{(y_i, y_j) \in \mathbb{R}^2 \,|\, y_i^2 + y_j^2 \leq R^2\}$. Let $r = \sqrt{y_i^2 + y_j^2}$ then we see that

$$\left|1 - \gamma(y_i^2 + y_i y_j + y_j^2)\right| \leq \begin{cases} 1, & r \leq \sqrt{\frac{2}{\gamma}}, \\ \gamma r^2 - 1, & r > \sqrt{\frac{2}{\gamma}}. \end{cases}$$

■

Proof of Lemma 4.3.1.

Lemma. *The matrix $\mathcal{L}_S(\sigma,\mu) + \mathcal{L}_S(\sigma,\mu)^\top$ is positive semi-definite for $\sigma, \mu \geq 0$.*

Proof. As shown by Steur et al. [31], denoting

$$a = \mu\bar{M}, \quad b_j = \frac{\sigma(1-\omega_{\bar{N}}^{j-1})}{\bar{M}}, \quad \omega_{\bar{N}} = \exp\left(\frac{2\pi i}{\bar{N}}\right),$$
$$d_j = \sqrt{a^2 + \bar{M}^2(\mathrm{Re}(b_j))^2 + (2\bar{M}-4)a\mathrm{Re}(b_j)},$$

the eigenvalues of

$$\mathcal{L}_S(\sigma,\mu) + \mathcal{L}_S(\sigma,\mu)^\top$$

are given by

$$\begin{aligned}
\lambda_1 &= 0, \\
\lambda_j &= a + \bar{M}\mathrm{Re}(b_j) - d_j, \\
\lambda_{N-1+j} &= a + \bar{M}\mathrm{Re}(b_j) + d_j, \\
\lambda_l &= 2a,
\end{aligned}$$

where $j = 2, ..., \bar{N}$ and $l = 2\bar{N}, 2\bar{N}+1, ..., \bar{N}\bar{M}$. For $\bar{M} \geq 2$, all involved terms are nonnegative and we see that the expressions for $\lambda_1, \lambda_{\bar{N}-1+j}$ and λ_l are nonnegative. Further note that

$$\begin{aligned}
\lambda_j \geq 0 &\iff \frac{a + \bar{M}\mathrm{Re}(b_j)}{2} \geq d_j, \\
&\iff \Big(a + \bar{M}\mathrm{Re}(b_j)\Big)^2 \geq a^2 + \bar{M}^2(\mathrm{Re}(b_j))^2 + (2\bar{M} - 4)a\mathrm{Re}(b_j), \\
&\iff 4a\mathrm{Re}(b_j) \geq 0.
\end{aligned}$$

Since both $a \geq 0$ and $\mathrm{Re}(b_j) \geq 0$ we thus also have $\lambda_j \geq 0$. It follows that $\mathcal{L}_S(\sigma, \mu) + \mathcal{L}_S(\sigma, \mu)^\top$ is positive semi-definite for $\sigma, \mu \geq 0$. ■

Appendix B

Symmetries of the Example Network

Symmetries of Pogromsky's Network

This section provides some additional material on the symmetries of Pogromsky's network for $k_2 = k_4$. For $k_2 = k_4$ there are at least 8 symmetries, call them $\Pi_1, \Pi_2, \ldots, \Pi_8$, which are independent of the values of k_1 and k_3. The associated permutation matrices are given below:

$$\Pi_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Pi_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\Pi_3 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Pi_4 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\Pi_5 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \Pi_6 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$\Pi_7 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \Pi_8 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Symmetries of Steur's Network

This section provides some additional material on the symmetries of Steur's network for $\bar{M} = \bar{N} = 3$. As has been previously established, Steur's network has

$$\bar{N}[(\bar{M}-1)!]^{\bar{N}}$$

symmetries (24 for $\bar{M} = \bar{N} = 3$). With each of these symmetries is associated a permutation matrix Π. Since we are interested in the orbit partitions of these symmetries and some of these 24 matrices might have identical orbit partitions, we reduce the set of symmetries to a set such that the associated orbit partitions are mutually distinct and for each symmetry in the original set, the associated orbit partitions is indeed represented in the reduced set. Note that such a reduced set is not unique. In the case of $\bar{M} = \bar{N} = 3$ it is viable compute such a set in a brute force manner while for networks with larger numbers of symmetries, more elegant ways of doing this need to be considered. It turns out, that a reduced set as previously described contains 13 matrices. One possible choice for such a set is listed below: The permutation matrix associated with a certain symmetry is listed on the right, while the Laplacian matrix of the associated reduced network is listed on the right. For a number of representative cases, Figures B.1 through B.5 give a visual representation of the symmetry and the associated reduced network.

$$\Pi_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 3 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix},$$

$$\Pi_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 3 & -1 & -1 & 0 & 0 & 0 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 3 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix},$$

$$\Pi_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 3 & -1 & -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix},$$

$$\Pi_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 3 & -2 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 3 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix},$$

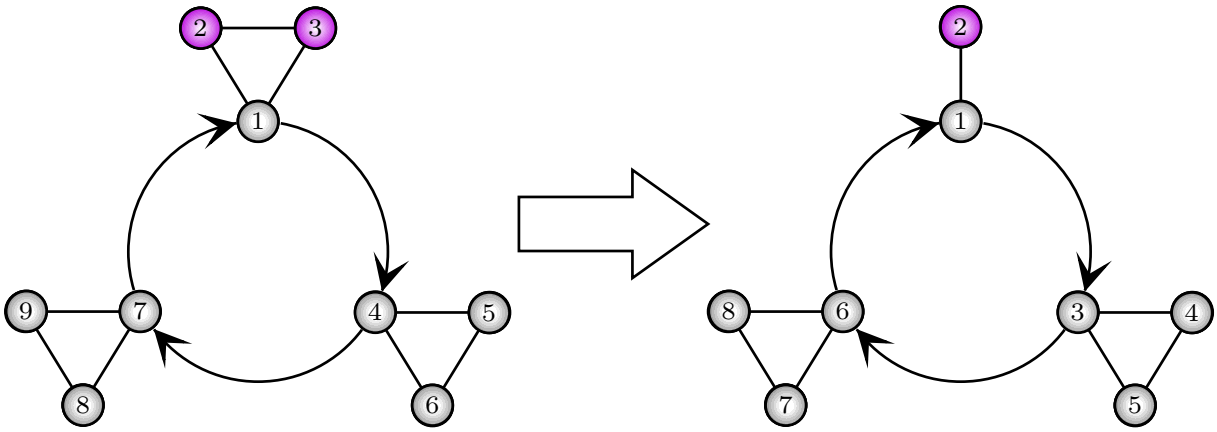

Figure B.1: Model reduction when clustering according to the orbit partition of Π_4. Nodes of the same colour are clustered together in the reduced network. For this type of symmetry, two nodes in a single module get clustered together. Other symmetries that yield a similar clustering of nodes in a single module are Π_2 and Π_3.

$$\Pi_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 3 & -1 & -1 & 0 & 0 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -2 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 3 & -2 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix},$$

$$\Pi_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 3 & -2 & 0 & 0 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 3 & -1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 2 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 3 & -2 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix},$$

$$\Pi_7 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 3 & -2 & 0 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 3 & -2 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix},$$

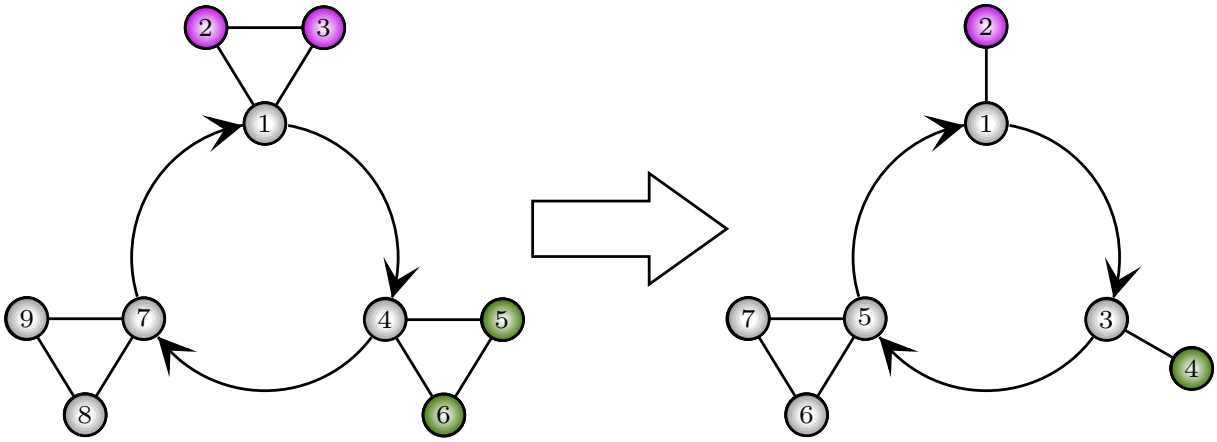

Figure B.2: Model reduction when clustering according to the orbit partition of Π_7. Nodes of the same colour are clustered together in the reduced network. For this type of symmetry, two nodes in the same module get clustered together and this happens for a total of 2 modules. Other symmetries that yield a similar clustering of nodes are Π_5 and Π_6.

$$\Pi_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 3 & -2 & 0 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 3 & -2 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 3 & -2 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix},$$

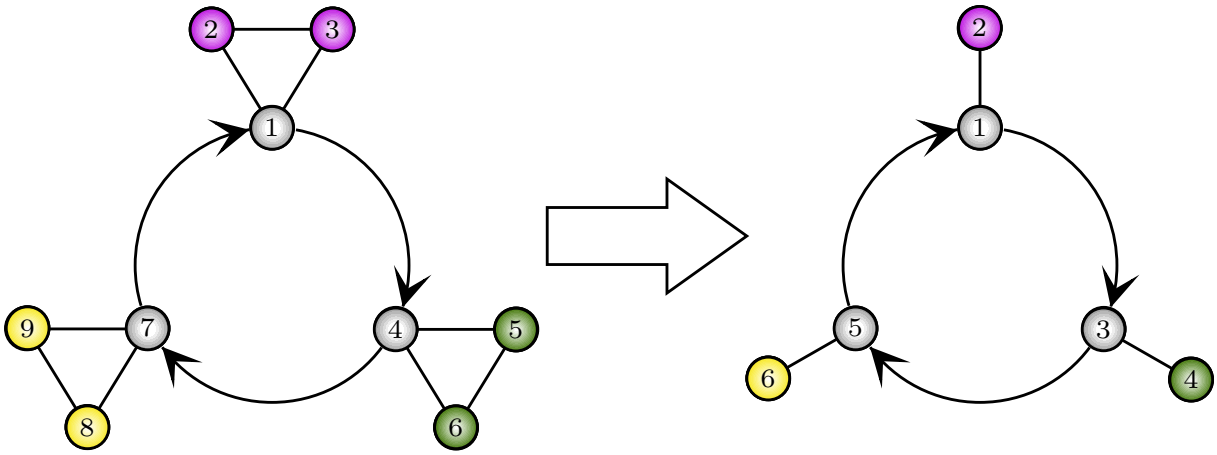

Figure B.3: Model reduction when clustering according to the orbit partition of Π_8. Nodes of the same colour are clustered together in the reduced network. For this type of symmetry, the nodes in the modules get clustered together.

$$\Pi_9 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix},$$

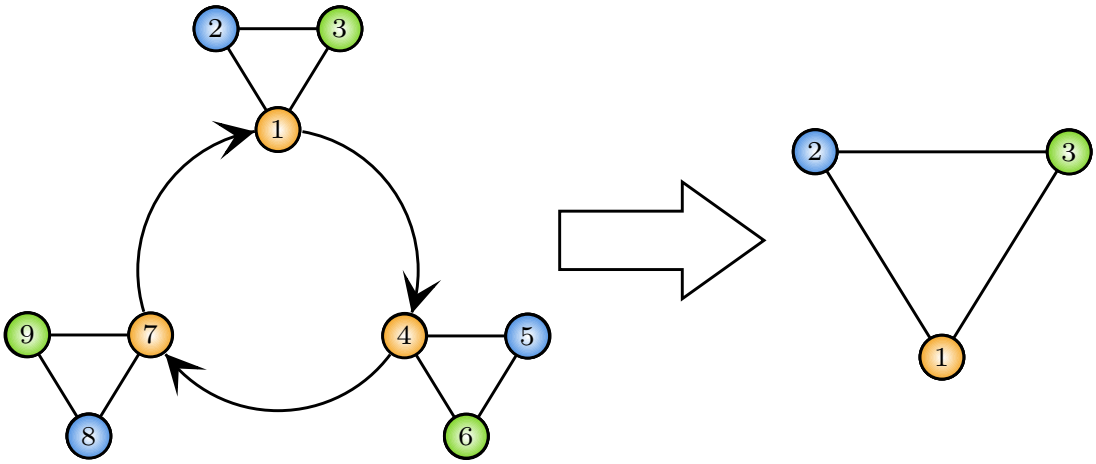

Figure B.4: Model reduction when clustering according to the orbit partition of Π_9. Nodes of the same colour are clustered together in the reduced network. For this type of symmetry, nodes belonging to the directed cycle are clustered together. Furthermore, nodes having similar relative locations inside the modules are clustered together. Other symmetries that yield a similar clustering of nodes are Π_{10}, Π_{11} and Π_{12}.

$$\Pi_{10} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix},$$

$$\Pi_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix},$$

$$\Pi_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix},$$

$$\Pi_{13} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{\mathcal{L}}_S = \begin{bmatrix} 2 & -2 \\ -1 & 1 \end{bmatrix}.$$

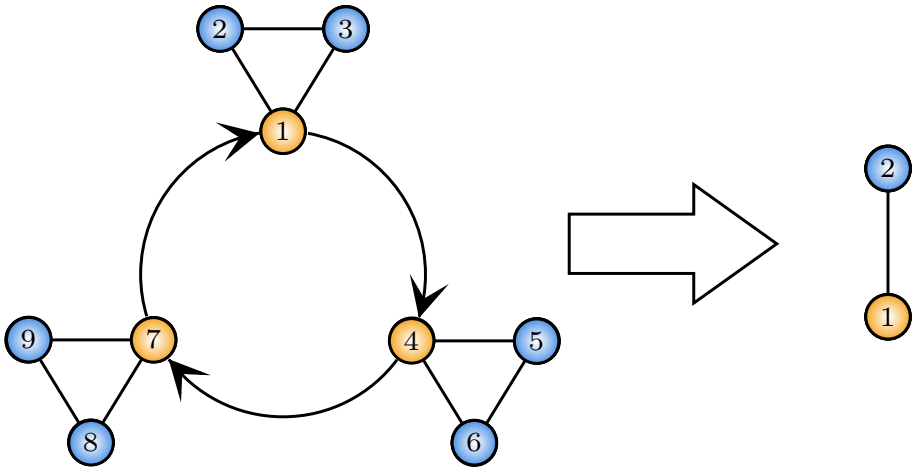

Figure B.5: Model reduction when clustering according to the orbit partition of Π_{13}. Nodes of the same colour are clustered together in the reduced network. For this symmetry, nodes are classified into two groups: those belonging to the inner structure of the directed cycle and those which do not. Nodes in the same group are then clustered together, yielding a network consisting of merely two nodes.

Appendix C

Bifurcations of Limit Cycles

In the following, we will briefly review the concept of the Poincaré map associated with a limit cycle and recall a few basic facts from Floquet theory that allow us to track the stability properties of a limit cycle as parameters in the underlying dynamical system are varied continuously.

The Poincaré Map

The following exposition of the Poincaré map closely follows the corresponding chapter in Kuznetsov's "Elements of Applied Bifurcation Theory" [12]. The Poincaré Map is a powerful tool in the analysis of limit cycle bifurcations. We use it to determine the stability of limit cycles. Let us consider some continuous-time system

$$\dot{x} = f(x), \quad x \in \mathbb{R}^n, \tag{C.1}$$

with a smooth function f. We assume the system (C.1) has a limit cycle L_0. Let x_0 be a point on L_0 and introduce a cross section to the limit cycle at x_0. This cross section, let us call it Σ, is a smooth $n-1$-dimensional hyper-surface intersecting L_0 at a nonzero angle. We assume that, near x_0, the cross section Σ is defined as the zero-level set of some smooth function $g : \mathbb{R} \to \mathbb{R}$ with $g(x_0) = 0$, i.e.

$$\Sigma = \{x \in \mathbb{R}^n \mid g(x) = 0\}.$$

In terms of this function g, we can express the nonzero intersection angle requirement as

$$\langle \nabla g(x_0), f(x_0) \rangle \neq 0,$$

with $\langle \cdot, \cdot \rangle$ being the usual scalar product in $\mathbb{R}^n$. Of course, a possible and very simple choice for Σ would then be to take it as the hyper-plane orthogonal to L_0 passing through x_0. The corresponding function g is then given as

$$g(x) = \langle f(x_0), x - x_0 \rangle.$$

We will now take a closer look at orbits of system (C.1) which pass through Σ close to x_0.

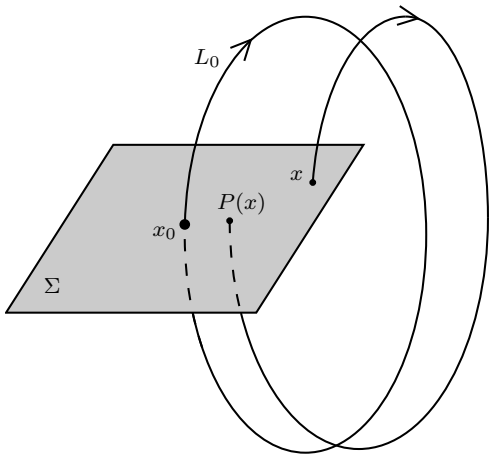

Figure C.1: Cross section at limit cycle and Poincaré map.

The limit cycle L_0 is such an orbit which starts on Σ at x_0 and of course also return to x_0. Since solutions of system (C.1) depend smoothly on their initial conditions we can make the following observations:

1. Take a point $x \in \Sigma$ sufficiently close to x_0, then the flow of system (C.1) will take the point x to another point $\tilde{x} \in \Sigma$ near x_0.

2. Orbits close to L_0 will also intersect Σ transversally (with nonzero angle).

These two facts taken together allow us to define a map $P : \Sigma \to \Sigma$ by

$$x \mapsto \tilde{x} = P(x).$$

We call this map the *Poincaré Map* associated to the cycle L_0.
The Poincaré map P is locally well defined, it is as smooth as the function f and it is invertible. The point x_0 clearly is a fixed point of P, i.e.

$$P(x_0) = x_0.$$

Let us now see what the Poincaré map has to do with the stability of the limit cycle. We introduce local coordinates $\xi = (\xi_1, ..., \xi_{n-1})$ on Σ such that $\xi = 0$ corresponds to x_0. Then locally, we can write the map P in terms of ξ, i.e. $P : \mathbb{R}^{n-1} \to \mathbb{R}^{n-1}$ is the map that takes a ξ which corresponds to x to a $\tilde{\xi}$ which corresponds to $\tilde{x}$,

$$P(\xi) = \tilde{\xi}.$$

By construction, the origin $\xi_0 = 0$ is then a fixed point and the stability of the cycle L_0 is equivalent to the stability of this fixed point. We denote the eigenvalues (multipliers) of the Jacobian matrix of P

$$A = \frac{dP}{d\xi}\Big|_{\xi=0},$$

as $\mu_1, \mu_2, ..., \mu_{n-1}$.
It is shown in "Elements of Applied Bifurcation Theory" that the multipliers of the Jacobian

matrix of the map P are independent of the choice of x_0 on L_0 as well as the choice of cross section and local coordinates. This of course is very convenient. Let us now investigate what this means for solutions of system (C.1).
Suppose $x^0(t)$ is a periodic solution ($x^0(t+T_0) = x^0(t)$) of system (C.1) which corresponds to the limit cycle L_0. Consider a solution of the form

$$x(t) = x^0(t) + u(t),$$

with $u(t)$ a deviation from the periodic solution. The deviation then evolves according to

$$\dot{u}(t) = \dot{x}(t) - \dot{x}^0(t) = f(x^0(t) + u(t)) - f(x^0(t)) = A(t)u(t) + \mathcal{O}(\|u(t)\|^2).$$

Neglecting all higher order terms, i.e. nonlinear terms, we arrive at the T_0-periodic, linear system

$$\dot{u} = A(t)u(t) \tag{C.2}$$

where $A(t) = f_x(x^0(t))$ and $A(t+T_0) = A(t)$. We call system (C.2) the *variational equation* about the cycle L_0.
Sufficiently close to the cycle L_0, the variational equation characterizes the dominant term describing the time evolution of an arbitrary perturbation. This means that the variational equation is closely related to the stability properties of L_0. The time-varying matrix $M(t)$ is called the fundamental solution matrix of system (C.1) if it satisfies

$$\dot{M} = A(t)M,$$

with $M(0) = I$, the identity matrix. It can be shown that any solution $u(t)$ to (C.2) satisfies

$$u(T_0) = M(T_0)u(0).$$

The matrix $M(T_0)$ is called a *monodromy matrix* of the cycle L_0. Provided that we know $M(T_0)$, we can now explicitly describe what happens to a perturbation after one period. In particular, we have the following result

Theorem C.0.1 ([12], p. 29)**.** *The monodromy matrix $M(T_0)$ has eigenvalues*

$$1, \mu_1, ..., \mu_{n-1},$$

where μ_i are the multipliers of the Poincaré map associated with the cycle L_0.

This theorem is the fundamental tool which will allow us to study the stability of limit cycles. The eigenvalues of the monodromy matrix are also referred to as *Floquet multipliers*. Given some limit cycle, we can simply compute the multipliers of the associated Poincaré map, which together with the eigenvalue 1 make up the Floquet multipliers. The monodromy matrix governs what happens to perturbations of solutions close to the limit cycle, so the Floquet multipliers can be used to study the stability of the cycle. We arrive at the following theorem.

Theorem C.0.2 ([12], p. 27)**.** *The cycle L_0 is stable if all eigenvalues (multipliers) $\mu_1, \mu_2, ..., \mu_{n-1}$ of the Jacobian matrix of P*

$$A = \frac{dP}{d\xi}\Big|_{\xi=0},$$

are located in the unit circle $|\mu| = 1$.

In the following, we will consider bifurcations of limit cycles in continuous-time systems which can be detected by monitoring the associated Floquet multipliers.

Floquet Multipliers and Bifurcations

When looking at a limit cycle of some continuous-time dynamical system which depends on a number of parameters, we can compute the associated Floquet multipliers. As we vary the parameter continuously, the Floquet multipliers also vary continuously. Invoking theorem C.0.2, we recall that for a stable limit cycle all Floquet multipliers are located inside the unit circle. It follows that generically there are three ways in which a limit cycle can lose its stability:

1. A Floquet multiplier leaves the unit circle at $\mu = 1$. This is called a *fold/tangent bifurcation.*
2. A Floquet multiplier leaves the unit circle at $\mu = -1$. This is called a *flip/period doubling bifurcation.*
3. A pair of Floquet multipliers leave the unit circle with $\mu_{1,2} = e^{\pm i\theta_0}$ with $0 < \theta_0 < \pi$. This is called a *Neimark-Sacker bifurcation.*

The corresponding situations are depicted in figure C.2. While both fold and flip bifurcations can be observed in one-dimensional systems, the Neimark-Sacker bifurcation occurs only in system with a least two dimensions.
Note that these bifurcations technically occur for the discrete-time system defined by the Poincaré map, however, if such a bifurcation occurs in the Poincaré map of an associated limit cycle, we will also say that the limit cycle itself undergoes the corresponding bifurcation.

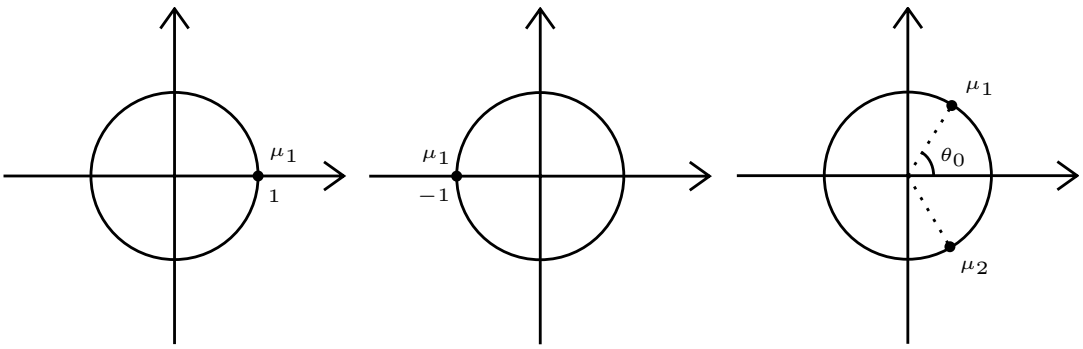

Figure C.2: Bifurcation scenarios for fold, flip and Neimark-Sacker bifurcation.

Appendix D

MATCONT

This appendix aims to familiarize the reader with the MATCONT continuation toolbox for MATLAB. In particular, this appendix contains a step-by-step explanation of the MATLAB script which was used to analyze Steur's network with MATCONT. An example from the results as discussed in section 4.3, will be used to demonstrate how MATCONT has been utilized. The versions in use are MATLAB R2020a and MATCONT version 7p2. MATCONT can be downloaded for free from SourceForge.net.

What is MATCONT?

MATCONT is a numerical bifurcation analysis tool for MATLAB which was first developed in 2000. There exists a command line version called CL_MATCONT and a GUI version simply called MATCONT. For most intends and purposes, the two versions have equivalent functionality, so in the remainder of this section we will simply refer to both as MATCONT.

Features of MATCONT are the continuation of equilibria and limit cycles of systems of Ordinary Differential Equations (ODEs), as well their bifurcations. MATCONT can compute Poincaré maps, as well as phase response curves for limit cycles. The toolbox supports computation of critical normal form coefficients for all codimension 1 and 2 bifurcations for equilibria. For limit cycles, computation of critical coefficients of periodic normal forms for codimension 1 bifurcations are supported. In fact, MATCONT has many more functionalities and a detailed documentation can be found in the official MATCONT manual [10].

A MATCONT tutorial

In this section, it will be outlined how MATCONT has been used in this thesis. In particular, this section aims to explain how the results as presented in section 4.3 were arrived at. In general, computations were carried out using CL_MATCONT, however, certain features of the GUI version were also employed.

This tutorial aims to explain the basics in using MATCONT in general terms, while at the same time working out an illustrative example. In this running example, a bifurcation analysis on Steur's network of FHN oscillators for one set of values for $\bar{M}$ and $\bar{N}$ will be carried out. The occurrence of modular travelling wave solutions will be in the focus.

The actual MATLAB files that have been used are available upon request. At the end of this tutorial, the reader will be able to reproduce all results as presented in section 4.3 using the provided MATLAB scripts and by adjusting parameters where necessary. While trying to give a complete description of the process involved in using MATCONT, only the most important commands and variables are being explained. The description of a number of commands, option settings and variables are not explicitly explained for the sake of brevity. However, in detail explanations of these can be found in the MATCONT manual [10].

Step 1: System Definition in MATCONT

Our first task will be defining our system in MATCONT. In the main MATCONT folder there exists a sub-folder called "Systems" which contains all the information about the individual systems handled by MATCONT. Each system has associated to it a specific folder, a MATLAB ".m" file and a MATLAB ".mat" file. Given a system of ODEs, we can in principle use the template file "standard.m" and fill in the right-hand side of our system of ODEs together with a number of extra information about the system, like Jacobian and Hessian matrices with respect to variables and parameters, if available. When defining a system manually like this, there are a number of things that one needs to pay close attention to, like the ordering of variables and parameters in the definition of the system of ODEs among other things. For smaller systems this is a feasible approach, however, for larger systems another way to define a system might be preferable: The GUI version has a build-in function, which allows to type in the equations of our system of ODEs in a rather straightforward way and then automatically create the aforementioned ".m" and ".mat" files, as well as a corresponding folder. There are two big advantages in using the automatic system file generation of the GUI:

1. The formatting of the system files is automatically taken care of.

2. When typing in the equations in the GUI, one can select that derivatives are computed symbolically, rather than numerically, up to the 5th order.

However, the symbolic computation of derivatives is only possible if the "Symbolic Math Toolbox" for MATLAB is installed. Let us now have a look at how this works in detail.

Example. We consider Steur's network with $\bar{M} = 2$ nodes per module and $\bar{N} = 5$ modules in total. This will give us a system of ODEs with 20 equations and two free parameters σ and μ. With this many equations, we can already anticipate that continuation run-times will become very long.

To define this system in MATCONT we open MATLAB, change the directory to the MATCONT folder and type `matcont`. This starts the GUI. We now define a new system by clicking `Select` and then `System` → `New`. A new window should pop up in which we can now specify our system. We name our system "FHN_M2N5" to represent the fact that we consider Steur's network of FHN oscillators with $\bar{M} = 2$ and $\bar{N} = 5$. As coordinates we type in `y1,z1,y2,z2,...,y10,z10`, note that we actually have to type out *all* coordinates, so no "..." notation. Further note that for all numerical computations we list the y component before the z component of each oscillator while in the main body of the thesis, we list z first and then

y which is more in line with the existing literature.
For parameters we type `sigma,mu` and as time we simply take `t`. Given the size of our system we only select symbolic derivatives of first order while the rest is computed numerically. Finally, we type in the ODEs individually. Here are the first two equations as an example:

```
y1'=(1-sigma-mu)*y1+(-1)*z1+(mu)*y2+(sigma)*y3-(1/3)*y1*y1*y1
z1'=(2/25)*y1+(-8/125)*z
```

These are the equations for the first FHN oscillator in Steur's network. Proper bracketing is particularly important at this point. Further note that the differential equations have to be specific in accordance with the order in which the coordinates were specified. It is also advised to type `x*x*x` rather than `x^3` as this sometimes leads to problems.

Of course typing out coordinates and the corresponding ODEs individually would be a tiresome process so we could use a MATLAB script to generate the text which is then to be pasted into the GUI. Once the equations have been entered, we press `OK` and after a short pre-computation, we find the folder "FHN_M5N5" together with the files "FHN_M5N5.m" and "FHN_M5N5.mat" in the "Systems" folder of MATCONT. We are now ready to analyze the system using either CL_MATCONT or with the GUI.

Step 2: Plotting Solutions as Time Series

Having defined our system, we will now start to analyze it using CL_MATCONT. We open MATLAB and choose as directory the MATCONT folder. To start we want to integrate the system of ODEs which is defined in our system file for a fixed configuration of parameters. To do this we first need to initialize MATCONT, then we can simply use the build-in "ode45" solver, which is simply MATLAB's "ode45" solver adapted for MATCONT. Let us again consider our running example.

Example. The analysis of Steurs network of FHN oscillators with $\bar{M} = 2$ and $\bar{N} = 5$ will be carried out by running the sections of the MATLAB script "FHN_M2N5_CL.m" which is available upon request. However, the most important commands as used in this script will be pointed out and explained so it should be possible to follow along and carry out the described analysis without the mentioned script.

- Set the variable `cds` to be global. The variable `cds` contains information about the most recent continuation and we will make use of it at a later point.
- Include the command `init`. This will add to path all necessary directories so that MATCONT can run in command-line mode.
- Include the commands `Func=eval('FHN_M2N5')` and `FHN_M2N5_ode=Func2`. The first command selects the earlier defined system with the name "FHN_M2N5" and the second command creates a function handle which evaluates the second function defined in the system file. This second function is simply the right-hand side of the system of ODEs.
- After having specified an initial condition `x0` and parameters `P0=[0.25;0.5]` we call

```
[t,y]=ode45(FHN_M2N5_ode,[0,tend],x0,[],P0(1),P0(2));
```

to compute the solution of the system of ODEs defined in "FHN_M2N5.m" for the time interval `0` to `tend` with the default set of options (`[]` as fourth argument). The result is a vector `t` containing the time steps and a solution matrix `y`. Solutions can now be plotted.

We plot only the solution curves $y_j(t)$ for the oscillators in this network.

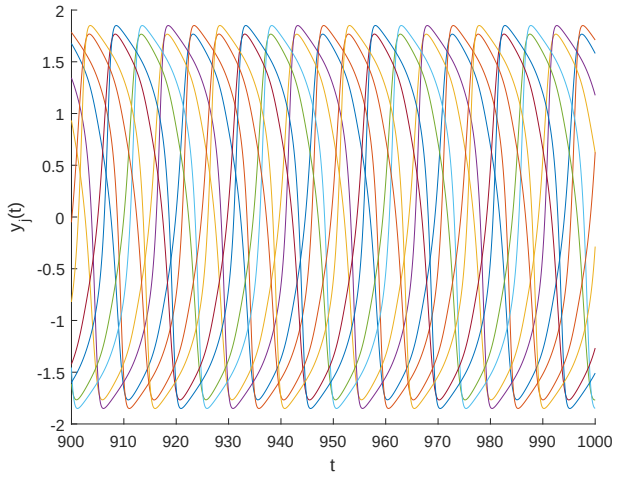

Figure D.1: Time series for Steur's network after initial response of the system.

As can be seen from figure D.1, the chosen parameter setting and initial condition are such that they correspond to a travelling wave solution. If not known beforehand, some trial and error would be necessary to find such a combination of parameter values and initial condition.

The travelling wave solution is periodic in time which means that we expect the orbit corresponding to this solution to converge to a limit cycle (LC) in state space. Inspection of figure D.1 indicates, that this travelling wave solution has a period of $t \approx 25$. This will now become important as we set out to continue the corresponding LC for one of the parameters.

Step 3: Limit Cycle Initialization

To continue a limit cycle (LC) in MATCONT for a certain parameter, we first need to *initialize* the LC continuation. Among other things, a LC continuation can be initialized from a Hopf-point, from a previous LC continuation or from a previously computed orbit. We will use initialization from a previously computed orbit. Let us see what this looks like for our system.

Example. Previously, we found a parameter setting $(\sigma, \mu) = (0.25, 0.5)$ and initial condition that yield a travelling wave solution. This solution corresponds to a LC in state space with a period of $t \approx 25$. We will now continue this LC for decreasing μ. To initialize a LC continuation from an orbit, we first need to know the period of the LC approximately. This we can find by inspection of figure D.1. Once we know the period, we need to make sure that the computed orbit is sufficiently close to the LC. If the LC is attracting, which it is in our case, an arbitrary orbit starting from a point in its basin of attraction will approach the LC with time. To make sure that the orbit is sufficiently close to the LC we can simply choose a very large time interval

for the initial time integration. We increase the time interval until we are certain that the orbit is no longer in transient but is close to the LC. To initialize the LC continuation, MATCONT now needs to know the solution points corresponding to roughly one period from the end time of our time integration.

Remark. In principle, MATCONT would take *any* points corresponding to approximately one period of the solution, however, the further back in the time simulation we take these points, the closer we are to the actual LC.

To find the index in the solution array corresponding to one period from the last time point we use

```
ind_period=find(t>t(end)-FHN_period,1,'first');
```

We set the current parameter values to `p=P0` and select μ as the active parameter (`ap`), i.e. we set the *second* parameter to be active, `ap=2`. The LC continuation is then initialized with

```
t_init=t(ind_period-overlap:end);
y_init=y(ind_period-overlap:end,:);
ntst=40;
ncol=4;
tol=1e-2;
[x0_LC,v0_LC]=initOrbLC(@FHN_M2N5,t_init,y_init,p,ap,ntst,ncol,tol);
```

The two arguments `ntst` and `ncol` of the `initOrbLC` command are parameters for orthogonal collocation which is used to discretize LCs in MATCONT. In our case, this means that the normalized interval $[0, 1]$, on which we consider the LC, is divided into `ntst=40` test-intervals of variable length and each of these intervals is then again divided into `ncol=4` intervals of equal length. In particular, the LC is then approximated by a continuous piece-wise polynomial which is of degree `ncol` in each test-interval. Increasing these two parameters is one of the things we can do, should we later run into numerical issues regarding the accuracy of results. Lastly, the parameter `tol` indicates the tolerance with which the LC is detected from the computed orbit.

Having run this section, we now have produced an initial point `x0_LC` corresponding to the LC with the current parameter setting and a tangent vector `v0_LC` which is tangent to the curve whose continuation corresponds to the continuation of the LC.

Step 4: Limit Cycle Continuation

Having initialized the limit cycle (LC), we can now continue it. To do so we first adjust the `opt` structure such that all features, we want MATCONT to employ, are activated. In particular, we set

```
opt=contset;
opt=contset(opt,'InitStepsize',0.01);
opt=contset(opt,'MinStepsize',0.01);
opt=contset(opt,'MaxStepsize',0.1);
opt=contset(opt,'MaxNumPoints',300);
```

```
opt=contset(opt,'Singularities',1);
opt=contset(opt,'IgnoreSingularity',[1;2;3]);
opt=contset(opt,'Backward',0);
opt=contset(opt,'Multipliers',1);
```

The first command deletes any preexisting configurations for the `opt` structure. The following three parameters control the initial, minimal and maximal step size during the continuation process respectively. The parameter `MaxNumPoints` indicates the maximum number of points we allow MATCONT to compute. Setting `Singularities` to 1 means that during the continuation process we want to check for zeros of test-functions, which indicate bifurcations. We can use the `IgnoreSingularity` parameter vector to indicate which bifurcations we *do not* want to check for. In our case these are all bifurcations except for the Neimark-Sacker (NS) bifurcation with index 4. Having selected all the wanted features, we can now continue the limit cycle with the command

```
[xLC,vLC,sLC,hLC,fLC]=cont(@limitcycle,x0_LC,v0_LC,opt);
```

The created variables contain the following information: Matrices `xLC` and `vLC` contain points on the continued curve that correspond to the continued LC. The structure `sLC` contains information about the found singularities. The matrices `hLC` and `fLC` contain more technical information about the continuation process but here, we will not use them a lot.

Since we are looking for a point in parameter space at which a NS bifurcation occurs, we could in principle run the continuation process from the initial LC given by `x0` and `v0` with ever increasing `MaxNumPoints` until we find such a NS point. Suppose we have the initial continuation run with `MaxNumPoints=300` but we have not found an NS point yet, even though we are sure we are moving in the right direction. Then it would be handy to simply extent this continuation rather than starting a new, longer continuation run from scratch. To do this we can use something like:

```
cds.options.MaxNumPoints=50;
[xLC,vLC,sLC,hLC,fLC]=cont(xLC,vLC,sLC,hLC,fLC,cds);
```

The first command lets us access and change the options settings in the global `cds` variable. This is done since we are no longer passing `@limitcycle` as the first argument but rather continue a previous continuation without reference to what kind of continuation that might have been. The second command then continues the LC with the specific options as found in `cds`. If we run the corresponding commands, MATLAB will display the following in the command window:

```
start computing extended curve
Neimark-Sacker (period = 2.349473e+01, parameter = 1.679790e-01)
Normal form coefficient = 4.196710e-07
elapsed time  = 66.9 secs
```

We found a Neimark-Sacker point for the parameter configuration $(\sigma, \mu) = (0.25, 0.1679790)$.

Step 5: Continuation of a Neimark-Sacker Bifurcation

The continuation of a Neimark-Sacker (NS) point is carried out very much analogously to the limit cycle (LC) continuation as discussed previously. Before we can continue such a bifurcation point, we first need to initialize such a continuation. To initialize, we use the following commands:

```
ap=[1 2];
ntst=40;
ncol=4;
[x0_NS,v0_NS]=init_NS_NS(@FHN_M2N5,xLC,sLC(2),ap,ntst,ncol);
```

The first command sets both σ and μ to be active parameters as we continue this bifurcation in two parameters. The second command initializes the NS continuation from a NS point. Recall, that during the LC continuation we discovered a NS bifurcation at $(\sigma, \mu) = (0.25, 0.1679790)$. Together with the current LC represented by `xLC`, the information about the location of the NS point contained in `sLC` is passed to the `init_NS_NS` command. The rest of the parameters that need to be specified were discussed earlier.
Having initialized the NS continuation, we can again specify the options `opt` with which we want to carry out the continuation. We use the following command to continue the NS point.

```
[xNS,vNS,sNS,hNS,fNS]=cont(@neimarksacker,x0_NS,v0_NS,opt);
```

We can then use the following command to plot the resulting curve in the parameter plane with the following command:

```
plot(xNS(end-2,:),xNS(end-1,:),'*b')
```

We simply select the third to last and second to last components of `xNS"`which contain the parameter values for the NS continuation. Using something along the lines of:

```
cds.options.MaxNumPoints=5;
cds.options.MinStepsize=0.1;
cds.options.InitStepsize=0.11;
[xNS,vNS,sNS,hNS,fNS]=cont(xNS,vNS,sNS,hNS,fNS,cds);
```

We can continue this NS curve in the parameter plane as far as we like. A number of calls `cont(xNS,vNS,sNS,hNS,fNS,cds)`, as well as adjusting the step sizes and the maximum number of points, might be necessary to reach a satisfactory result. After a number of extensions of the continuation, we arrive at the NS curve for $\bar{M} = 2$ and $\bar{N} = 5$ as displayed in Figure D.2 (compare to Figures 4.10 and 4.11 in section 4.3).

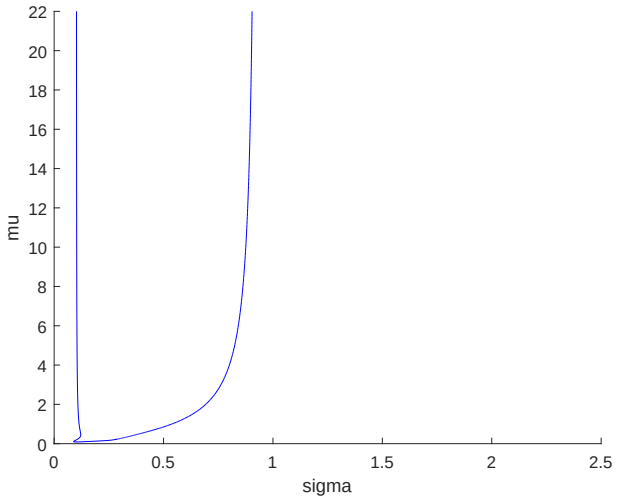

Figure D.2: Neimark-Sacker curve for Steur's network with $\bar{M} = 2$ and $\bar{N} = 5$.

Step 6: Locating the Chenciner Bifurcation

We saw earlier, that every time MATCONT detects a Neimark-Sacker (NS) bifurcation during the continuation of a limit cycle (LC), it also computes the normal form coefficient. Initializing the LC continuation from orbits corresponding to different parameter configurations, one soon notices that these normal form coefficients are positive in certain regions on the NS curve and negative in others. This indicates, that there occurs a Chenciner (CH) bifurcation, i.e. there is a point on the NS curve where the normal form coefficient is zero. In this case CL_MATCONT allows for checking for such a point by means of monitoring the test-functions. To continue the NS curve while checking for a CH bifurcation we can largely reuse the commands for the continuation of the NS curve. We merely change the initialization to

```
[x0_NS,v0_NS]=init_CH_NS(@FHN_M2N5,xLC,sLC(2),ap,ntst,ncol);
```

Remark. It does not matter if one uses `init_CH_NS` or `init_NS_NS` as calling `init_CH_NS` boils down to calling `init_NS_NS` with the same arguments.

To check for the CH bifurcation we have to switch on the computation of multipliers, monitor for singularities and ignore all except for the singularity with index 6 (CH bifurcation). Further, we set our options such that the test functions are being computed.

```
opt=contset(opt,'Multipliers',1);
opt=contset(opt,'Singularities',1);
opt=contset(opt,'IgnoreSingularity',[1;2;3;4;5;7;8]);
opt=contset(opt,'Testfunctions',1);
[xNS,vNS,sNS,hNS,fNS]=cont(@neimarksacker,x0_NS,v0_NS,opt);
```

Lastly, having computed `xNS` and `vNS`, we can also compute the NS normal form coefficient at point i on the curve using the following command:

```
feval(cds.curve_testf,6,xNS(:,i),vNS(:,i));
```

Extending the curve a number of times while checking for a CH bifurcation, we note that at a certain point, we will get the warning "the real part of the cubic coefficient of the normal form xxxxxxxxx is not purely imaginary." At this point calculations stop and we cannot interact with MATCONT anymore. What exactly happens here is still unclear. However, if we plot the NS normal form coefficients in a neighborhood of this point, we obtain figure D.3. It seems that the normal form coefficient vanishes somewhere between $\sigma = 0.225$ and $\sigma = 0.23$. This is strong indication that a Chenciner bifurcation indeed occurs.

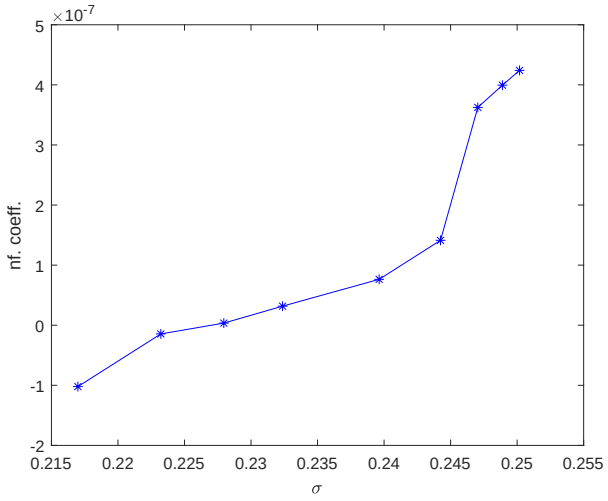

Figure D.3: Normal form coefficients of the NS bifurcation in the neighborhood where we suspect the Chenciner bifurcation.

Appendix E

Travelling Waves in Ring Networks

The effect of network topology on the dynamics of an associated network system is a widely studied problem. Gorban et al. [8] study network systems of nonlinear oscillators in which the network structure is a directed chain or directed ring. Denoting N the number of nodes in a ring network and σ the overall coupling strength, the authors demonstrate that for certain pairs (σ, N), oscillators can be expected to fully synchronize while for others, travelling waves might coexist with the fully synchronous solution. Inspired by this approach, we study the existence of travelling waves in directed and undirected ring networks and extend the analysis by also consider orientation of travelling waves.

The Laplacian matrix of the directed ring is given by

$$\mathcal{L}_{\mathrm{d}} = \begin{bmatrix} 1 & 0 & 0 & \dots & -1 \\ -1 & 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & -1 & 1 & 0 \\ 0 & 0 & \dots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{N \times N},$$

and the Laplacian of the undirected ring is given by

$$\mathcal{L}_{\mathrm{u}} = \frac{1}{2}(\mathcal{L}_{\mathrm{d}} + \mathcal{L}_{\mathrm{d}}^{\top}).$$

For fixed N, we consider the Laplacian parametrized by an overall coupling strength σ, i.e.

$$\mathcal{L}(\sigma) = \sigma \mathcal{L}_0.$$

Here $\mathcal{L}_0 = \mathcal{L}_d$ or $\mathcal{L}_0 = \mathcal{L}_u$. We consider a grid for the coupling strengths $\sigma = \{0.1, 0.2, \dots, 1\}$. For each σ, we sample uniformly randomly from $|z_i(0)| \leq \frac{15}{8}\sqrt{3}$ and $|y_i(0)| \leq \frac{3}{2}\sqrt{3}$ a total of 1000 initial conditions. Using MATLAB's *ode45* solver with relative and absolute error tolerance of 10^{-5} the system is integrated for intervals of $\Delta t = 100$ units of time. The integration runs for a maximum time of $t_{\mathrm{end}} = 500$. After each time interval Δt it is checked whether the solution as computed on the previous interval is fully synchronous or a travelling wave. If at $t_{\mathrm{end}} = 500$ the solution could not be identified as fully synchronous or a travelling wave, it will be registered as "unidentified". Furthermore, every time a travelling wave is registered, we also check for the

direction of the wave, i.e. does it wander clockwise or anti-clockwise. For $N = 8$, the results of this experiment are summarized in Figures E.1 and E.2.

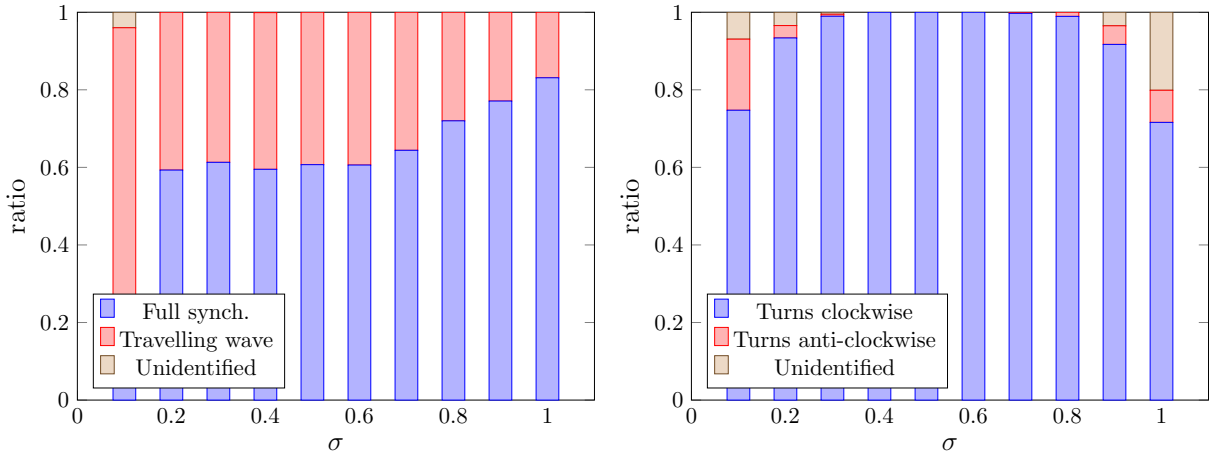

Figure E.1: Directed ring network with $N = 8$ nodes. Left panel: Relative numbers of solutions identified as fully synchronous or travelling wave, as a function of the coupling strength σ. Right panel: Relative numbers of travelling wave solutions turning clockwise or anti-clockwise, as a function of the coupling strength σ.

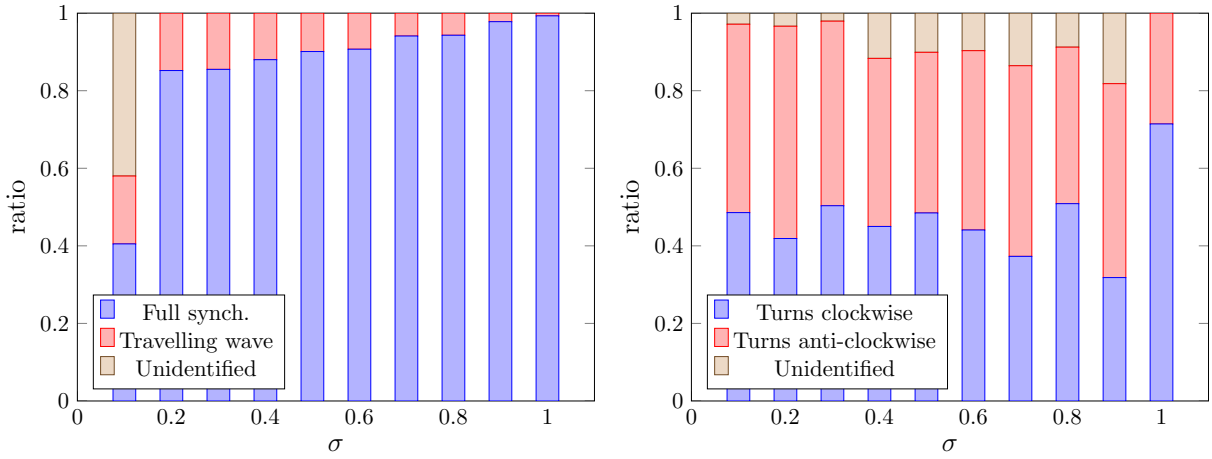

Figure E.2: Undirected ring network with $N = 8$ nodes. Left panel: Relative numbers of solutions identified as fully synchronous or travelling wave, as a function of the coupling strength σ. Right panel: Relative numbers of travelling wave solutions turning clockwise or anti-clockwise, as a function of the coupling strength σ.

Comparing Figures E.1 and E.2, it appears that directed coupling promotes the occurrence of travelling waves in general and that these travelling waves then wander exclusively in the direction as prescribed by the directed coupling. In the case of undirected coupling, the occurrence of full synchronization is promoted in general. However, travelling waves wandering in either direction can occur. As can be seen from Figure E.2, travelling waves of either orientation occur with almost equal frequency across values of σ. This is not very surprising because the undirected ring is essentially half clockwise oriented directed ring and half anti-clockwise directed ring. Let us explore what the situation is for a ring network that is a convex combination of clockwise and anti-clockwise oriented directed ring.

We consider a modified ring network with Laplacian matrix

$$\mathcal{L}_{\mathrm{m}}(p) = \begin{bmatrix} 1 & -p & 0 & \dots & -(1-p) \\ -(1-p) & 1 & -p & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & -(1-p) & 1 & -p \\ -p & 0 & \dots & -(1-p) & 1 \end{bmatrix}, \quad p \in [0,1].$$

Note that $\mathcal{L}_{\mathrm{m}}(0) = \mathcal{L}_{\mathrm{d}}$ is simply the Laplacian of the clockwise directed ring while $\mathcal{L}_{\mathrm{m}}(1)$ is the Laplacian of the anti-clockwise directed ring. Further, $\mathcal{L}_{\mathrm{m}}(\frac{1}{2}) = \mathcal{L}_{\mathrm{u}}$ and so the modified ring network can be thought of as a superposition of directed networks of either orientation where p modulates between the two orientations. We consider the Laplacian with a fixed overall coupling strength σ which is independent of p, i.e. $\mathcal{L}(p) = \sigma\mathcal{L}_m(p)$. The choice of σ is important: if it is too high the system will simply synchronize all the time and the effect of p will not be visible, if it is too low, convergence to full synchronization or travelling waves will take too long and our setup will fail to identify them. From Figures E.1 and E.2, we can see that for $\sigma = 0.5$ in the directed case roughly 40% of solutions are identified as travelling waves and for the undirected ring it is roughly 10%. With a total of 1000 independent initial conditions this leaves us with around 400 and 100 solutions which will serve as a basis to calculate the ratio of clockwise versus anti-clockwise wandering solutions. We again run the experiment as previously discussed. The results are summarized in Figure E.3.

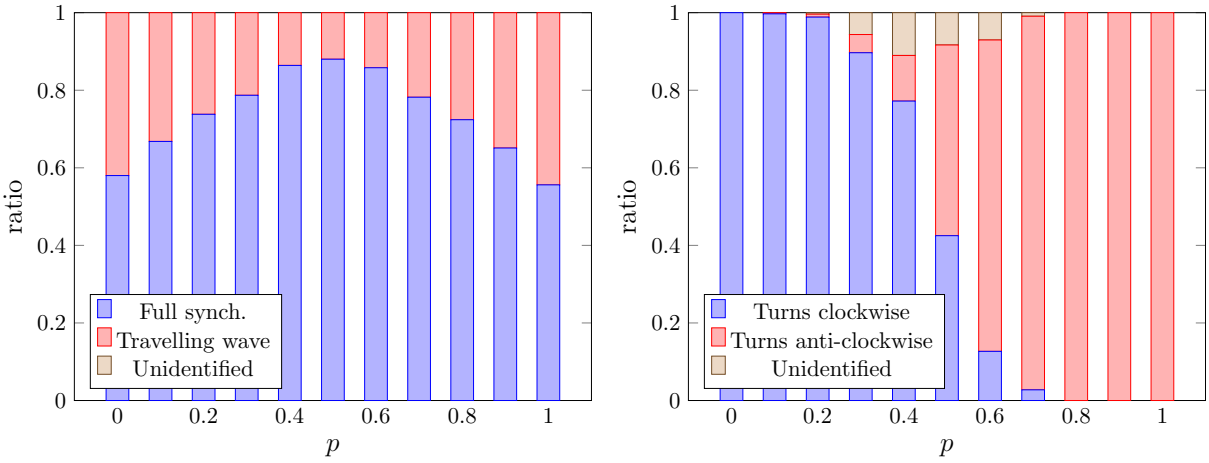

Figure E.3: Modified ring network with $N = 8$ nodes and overall coupling strength $\sigma = 0.5$. Left panel: Relative numbers of solutions identified as fully synchronous or travelling wave, as a function of the modulation parameter p. Right panel: Relative numbers of travelling wave solutions turning clockwise or anti-clockwise, as a function of p.

As expected, the number of travelling waves solutions ranges from roughly 40% in the case of a directed ring ($p = 0$ or $p = 1$) of either orientation to about 10% in the case of the undirected ring ($p = 0.5$). For $p = 0$ we see that all travelling waves turn clockwise while for $p = 1$ all solutions turn anti-clockwise. For $p = 0.5$ roughly 50% of travelling waves wander clockwise while 50% wander anti-clockwise. Interestingly, the relationship between the ratios and the parameter p does not seem to be linear. Further, the dependency on N might also be

interesting.

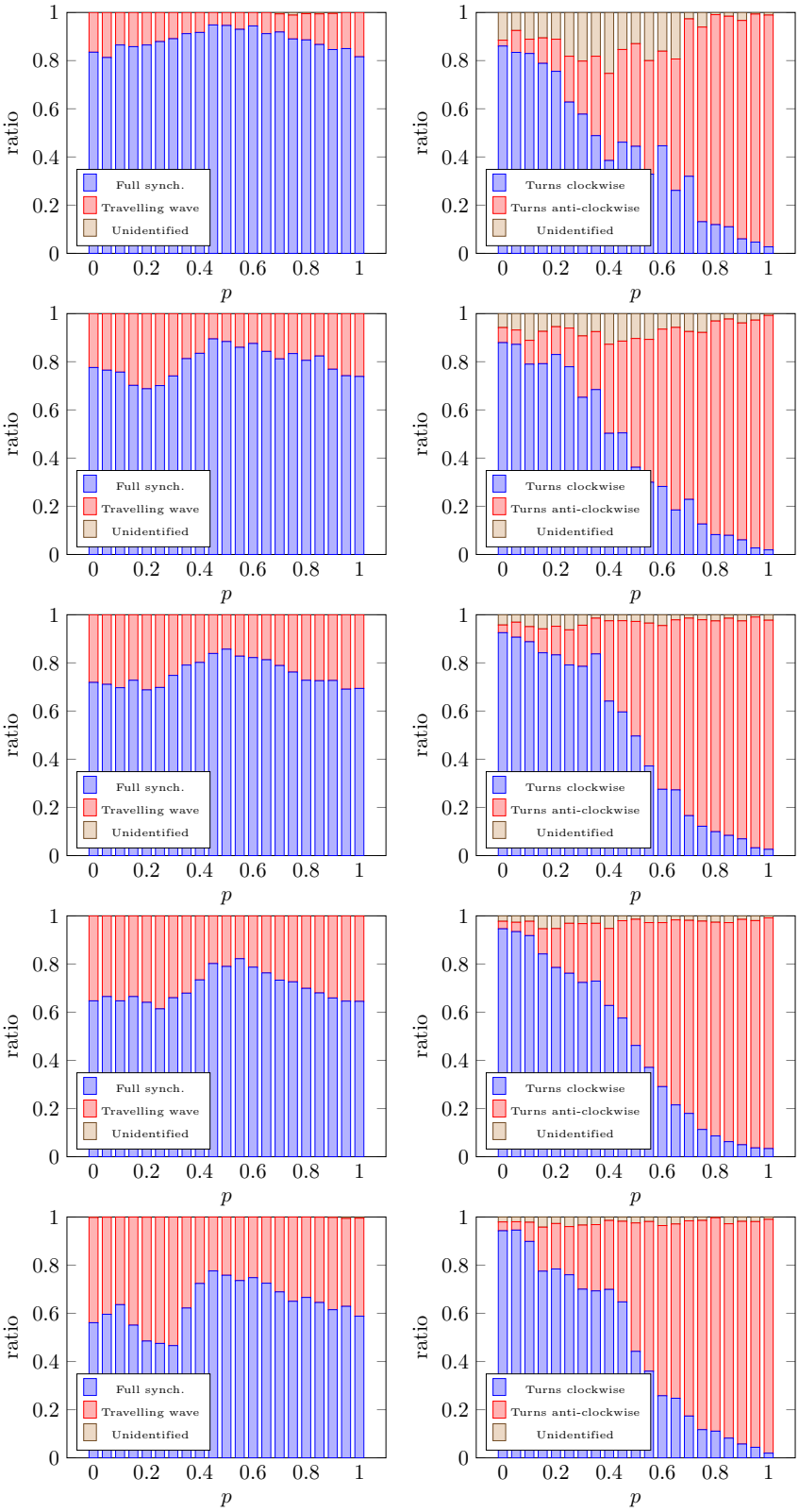

Figure E.4: Modified ring network with $N = 6, 7, \ldots, 10$ nodes and overall coupling strength $\sigma = 0.25$. The plots in row 1 correspond to $N = 6$, the plots in row 2 to $N = 7$ etc.. Left panels: Relative numbers of solutions identified as fully synchronous or travelling wave, as a function of the modulation parameter p. Right panels: Relative numbers of travelling wave solutions turning clockwise or anti-clockwise, as a function of p.

To explore the dependency on the number of nodes N in the modified ring network, we repeat the previously discussed experiment for $N = 6, 7, \ldots, 10$ with $p = \{0, 0.05, 0.1, \ldots, 1\}$ and overall coupling strength $\sigma = 0.25$. This overall coupling strength has been chosen to ensure that the set of studied travelling waves is large enough. With this parameter setting, the relative number of solutions identified as travelling wave ranges from roughly 5% in the case of $N = 6$ and $p = 0.5$ to over 50% for $N = 10$ and $p = 0.3$. The results are summarized in Figure E.4. For $N = 6, 7, \ldots, 10$, Figure E.5 shows the ratio of travelling waves which turn clockwise as a function of p.

The plots support the hypothesis that the relationship between the ratios and the parameter p is nonlinear, however it is not entirely clear whether the number of nodes has an influence on the shape of these curves. An extended experimental setup with increased accuracy for the detection of travelling waves and their direction together with an increased number of independently sampled initial conditions might yield more precise answers to this question.

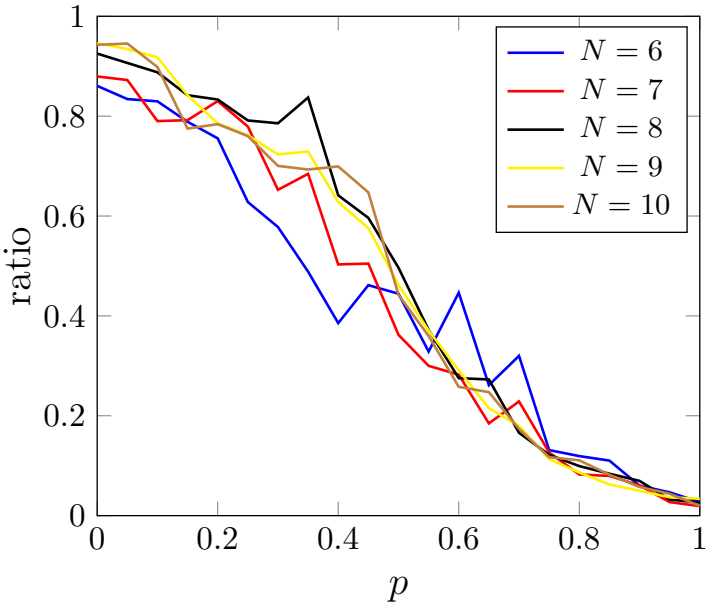

Figure E.5: Relative number of travelling waves that turn clockwise in the modified ring network as a function of the parameter p.

Bibliography

[1] Z. Aminzare, B. Dey, E. N. Davison, and N. E. Leonard. Cluster synchronization of diffusively coupled nonlinear systems: A contraction-based approach. *Journal of Nonlinear Science*, pages 1–23, 2018.

[2] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*, volume 6. Siam, 2005.

[3] F. A. C. Azevedo, L. R. B. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. L. Ferretti, R. E. P. Leite, W. J. Filho, R. Lent, and S. Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5):532–541, 2009.

[4] B. Bollobás. *Modern Graph Theory*. Springer Science & Business Media, 2013.

[5] M. C. de Oliveira and R. E. Skelton. On stability tests for linear systems. *IFAC Proceedings Volumes*, 35(1):125–130, 2002.

[6] B. P. Demidovich. On dissipativity of some system of nonlinear differential equations: I. *Vestnik Moscow State University*, 6:19–27, 1961.

[7] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1(6):445, 1961.

[8] A. N. Gorban, N. Jarman, E. Steur, H. Nijmeijer, C. Van Leeuwen, and I. Y. Tyukin. Directed cycles and multi-stability of coherent dynamics in systems of coupled nonlinear oscillators. *IFAC-PapersOnLine*, 48(18):19–24, 2015.

[9] A. N. Gorban, N. Jarman, E. Steur, C. van Leeuwen, and I. Y. Tyukin. Leaders do not look back, or do they? *Mathematical Modelling of Natural Phenomena*, 10(3):212–231, 2015.

[10] W. Govaerts, Y. A. Kuznetsov, H. G. E. Meijer, B. Al-Hdaibat, V. De Witte, A. Dhooge, W. Mestrom, N. Neirynck, A. M. Riet, and B. Sautois. Matcont: Continuation toolbox for odes in matlab. 2018.

[11] H. K. Khalil and J. W. Grizzle. *Nonlinear Systems*, volume 3. Prentice Hall Upper Saddle River, NJ, 2002.

[12] Y. A. Kuznetsov. *Elements of Applied Bifurcation Theory*, volume 112. Springer Science & Business Media, 2013.

[13] W. Lohmiller and J. J. E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

[14] B. D. MacArthur, R. J. Sánchez-García, and J. W. Anderson. Symmetry in complex networks. *Discrete Applied Mathematics*, 156(18):3525–3531, 2008.

[15] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*, volume 33. Princeton University Press, 2010.

[16] R. E. Mirollo and S. H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, 1990.

[17] N. Monshizadeh, H. L. Trentelman, and M. K. Camlibel. Projection-Based Model Reduction of Multi-Agent Systems Using Graph Partitions. *IEEE Transactions on Control of Network Systems*, 1(2):145–154, 2014.

[18] A. E. Motter, S. A. Myers, M. Anghel, and T. Nishikawa. Spontaneous synchrony in power-grid networks. *Nature Physics*, 9(3):191–197, 2013.

[19] J. Nagumo, S. Arimoto, and S. Yoshizawa. An Active Pulse Transmission Line Simulating Nerve Axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

[20] S. Nair and N. E. Leonard. Stable Synchronization of Mechanical System Networks. *SIAM Journal on Control and Optimization*, 47(2):661–683, 2008.

[21] L. M. Pecora, F. Sorrentino, A. M. Hagerstrom, T. E. Murphy, and R. Roy. Cluster synchronization and isolated desynchronization in complex networks with symmetries. *Nature Communications*, 5(1):1–8, 2014.

[22] A. Y. Pogromsky. A partial synchronization theorem. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 18(3):037107, 2008.

[23] A. Y. Pogromsky. Erratum:"A partial synchronization theorem"[Chaos 18, 037107 (2008)]. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(4):049901, 2009.

[24] A. Y. Pogromsky, T. Glad, and H. Nijmeijer. On diffusion driven oscillations in coupled dynamical systems. *International Journal of Bifurcation and Chaos*, 9(04):629–644, 1999.

[25] A. Y. Pogromsky and H. Nijmeijer. Cooperative oscillatory behavior of mutually coupled dynamical systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(2):152–162, 2001.

[26] A. Y. Pogromsky, G. Santoboni, and H. Nijmeijer. Partial synchronization: from symmetry towards stability. *Physica D: Nonlinear Phenomena*, 172(1-4):65–87, 2002.

[27] A. J. Raven. Clustering-based model reduction of a network of nonlinear oscillators. BSc. thesis, University of Groningen, 2019.

[28] C. Rocsoreanu, A. Georgescu, and N. Giurgiteanu. *The FitzHugh-Nagumo Model: Bifurcation and Dynamics*, volume 10. Springer Science & Business Media, 2012.

[29] S. Smale. A mathematical model of two cells via Turing's equation. In *The Hopf Bifurcation and Its Applications*, pages 354–367. Springer, 1976.

[30] F. Sorrentino, L. M. Pecora, A. M. Hagerstrom, T. E. Murphy, and R. Roy. Complete characterization of the stability of cluster synchronization in complex dynamical networks. *Science Advances*, 2(4), 2016.

[31] E. Steur, I. Tyukin, A. N. Gorban, N. Jarman, H. Nijmeijer, and C. van Leeuwen. Coupling-modulated multi-stability and coherent dynamics in directed networks of heterogeneous nonlinear oscillators with modular topology. *IFAC-PapersOnLine*, 49(14):62–67, 2016.