

# ScholarWorks@GSU

## Synchronization in Dynamical Networks with Mixed Coupling

Authors	Carter, Douglas M
Citation	Carter, Douglas M. "Synchronization in Dynamical Networks with Mixed Coupling". Dissertation. Georgia State University, 2016. <a href="https://doi.org/10.57709/8513838">https://doi.org/10.57709/8513838</a>
DOI	<a href="https://doi.org/10.57709/8513838">https://doi.org/10.57709/8513838</a>
Download date	2026-03-08 21:20:35
Link to Item	<a href="https://hdl.handle.net/20.500.14694/10304">https://hdl.handle.net/20.500.14694/10304</a>

# SYNCHRONIZATION IN DYNAMICAL NETWORKS WITH MIXED COUPLING

by

DOUGLAS M. CARTER JR.

Under the Direction of Igor Belykh, PhD

## ABSTRACT

Synchronization is an important phenomenon which plays a central role in the function or dysfunction of a wide spectrum of biological and technological networks. Despite the vast literature on network synchronization, the majority of research activities have been focused on oscillators connected through one network. However, in many realistic biological and engineering systems the units can be coupled via multiple, independent networks. This thesis contributes toward the rigorous understanding of the emergence of stable synchronization in dynamical networks with mixed coupling. A mixed network is composed of subgraphs connecting a subnetwork of oscillators via one of the individual oscillator's variables. An illustrative example is a network of Lorenz systems with mixed couplings where some of the oscillators are coupled through the  $x$ -variable, some through the  $y$ -variable and some through both. This thesis presents a new general synchronization method called the Mixed Connection Graph method, which removes a long-standing obstacle in studying synchronization in mixed dynamical networks of different nature. This method links the stability theory, including the Lyapunov function approach with graph theoretical quantities. The application of the method to specific networks reveals surprising, counterintuitive effects, not seen in networks with one connection graph.

INDEX WORDS: Dynamical System, Synchronization, Stability, Graph Theory

SYNCHRONIZATION IN DYNAMICAL NETWORKS WITH MIXED COUPLING

by

DOUGLAS M. CARTER JR.

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2016

Copyright by  
Douglas Carter Jr  
2016

SYNCHRONIZATION IN DYNAMICAL NETWORKS WITH MIXED COUPLING

by

DOUGLAS M. CARTER JR.

Committee Chair: Igor Belykh

Committee: Alexandra Smirnova  
Vladimir Bondarenko  
Yaroslav Molkov

Electronic Version Approved:

Office of Graduate Studies  
College of Arts and Sciences  
Georgia State University  
May 2016

## DEDICATION

I would like to thank my family for believing in me and their support as I pursued my Ph.D. Many thanks to my Mom and Dad for teaching me a great deal of humility and discipline along with their unconditional love. Dad, although you're resting peacefully in heaven, I just want you to know you've turned me into a well-polished man and made me more than proud to be your son. If I shall be so lucky to have a son/daughter one day, rest assured Dad, my child will be raised with respect and honor all the other facets you've blessed me. I can truthfully say without a shadow of doubt, your legacy will carry on through many people because everyone who's crossed path with you are a testament to the life lessons gained and I'd be more than honored to be at the helm. Thank you Dad for the knowledge you've empowered upon me. The skills you've taught me will continue to carry on through me. Mom, your accounting background has certainly steered me in the area of mathematics which was a blessing in disguise. In addition, thank you for your continued patience and assistance during those times when I just wasn't able to comprehend/grasp a particular subject from elementary to high school.

To all my Crucian sisters, thanks for offering your guidance as I battle many challenges considering I am the only male in the family and the youngest. Each one of you has brought me comfort in my heart knowing I could always count on you when Dad or Mom assigned someone in charge.

## ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my PhD advisor, Professor Igor Belykh, for his consistent support, guidance, and encouragements for the past few years. His advice always inspired me to think broader and deeper. I feel fortunate to have been given the honor and privilege to work with him.

Moreover, I offer my sincere gratitude to Professors Vladimir Bondarenko, Alexandra Smirnova, and Yaroslav Molkov for serving on my Ph.D committee and taking time out of their busy schedule to review this work. I am also deeply indebted to Dr. Bondarenko for introducing me to Dr. Belykh prior to being accepted into the PhD program because if it wasn't for him, I would've never been given the chance to work with Dr. Belykh.

In addition, I would be remiss if I didn't mention the professor who helped jump start my interest in Applied Mathematics. His name is Sudhakar Pandit. I had the pleasure in taking many undergraduate courses under his tutelage. During my junior year, he would notice my talent through leading class discussions and excellent test grades. This observation would give me the opportunity to work on my first research paper [1] and a second paper [2] within one year later. Thereafter, both papers were published in two popular journals. So I give thanks to Dr. Pandit for playing a tremendous role with my education accomplishments.

I am also grateful to Professor Francesco Sorrentino of the Department of Mechanical Engineering at the University of New Mexico who visited our Mathematics Department and shared his Matlab code for the simultaneous block-diagonalization of two matrices which I used for a comparative study of the method developed in this thesis and the existing approaches.

I'd be remissed to not mention the Mathematics Department staff who assisted me with numerous task while I was an instructor and student at GSU. These staff members are Sandra Ahuama-Jonas, Yvonne Pierce and Earnestine Collier-Jones. I certainly appreciate all the advice and services you've offered me through this journey.

Finally, thank you to all my friends who have offered their assistance in this journey :  
Russell Jeter, Tingli Xing, Kun Zhao, Malcolm DeVoe, Leslie Meadows, Jeffrey McCannon,  
Reimbay Reimbayev, Barrett Brister, Maxime Bouadoumou and Kevin Daley.

### **Support**

This research was partly supported by the National Science Foundation under Grants No.  
DMS-1009744 and DMS-1616345.

## TABLE OF CONTENTS

<b>ACKNOWLEDGEMENTS</b>		<b>v</b>
<b>LIST OF FIGURES</b>		<b>ix</b>
<b>Chapter 1 INTRODUCTION</b>		<b>1</b>
1.1 Motivation and problem statement		1
1.2 Synchronization in one-component networks: A review		5
1.2.1 Network model and type of synchronization considered		5
1.2.2 Existing eigenvalue-based stability methods		7
1.2.3 Connection graph stability method		9
<b>Chapter 2 SYNCHRONIZATION IN MIXED NETWORKS: THE MIXED GRAPH CONNECTION METHOD</b>		<b>13</b>
2.1 Network model and problem statement		13
2.2 Simultaneous Block Diagonalization and Reduction Dimensionality		16
2.3 Mixed Graph Stability Method: The General Proof		17
2.3.1 Stability System for the Difference Variables		18
<b>Chapter 3 APPLICATION OF THE MIXED GRAPH METHOD</b>		<b>32</b>
3.1 Four Oscillator Chain Network		32
3.1.1 Proof of the Stability Condition		33
3.1.2 Computing the Stability Conditions		41
3.2 Other Networks		47
3.2.1 Ten Oscillator Chain Network		48
3.2.2 Six Oscillator Tree Network		50

	viii
3.3 Numerical vs Theoretical Synchronization Thresholds . . . . .	58
3.3.1 Another Twenty Oscillator Network . . . . .	64
<b>Chapter 4 CONCLUSIONS . . . . .</b>	<b>68</b>
<b>REFERENCES . . . . .</b>	<b>70</b>
<b>APPENDICES . . . . .</b>	<b>75</b>
<b>Appendix A SYNCHRONIZATION CONDITION FOR TWO <math>X, Y</math>-COUPLED LORENZ SYSTEMS . . . . .</b>	<b>76</b>
<b>Appendix B COMPUTING THE SHORTEST PATH . . . . .</b>	<b>79</b>
<b>Appendix C MATLAB CODE FOR SYNCHRONIZATION THRESH- OLD . . . . .</b>	<b>82</b>
<b>Appendix D MATLAB CODE FOR MIXED BK . . . . .</b>	<b>86</b>
<b>Appendix E PYTHON CODE FOR UNIFORM BK . . . . .</b>	<b>95</b>

## LIST OF FIGURES

Figure 1.1	Five-oscillator network. . . . .	6
Figure 1.2	Synchronization of two oscillators. . . . .	7
Figure 1.3	Star Graph . . . . .	10
Figure 1.4	Shortest Path counter-example . . . . .	11
Figure 2.1	Hyper-Network example . . . . .	14
Figure 2.2	Lorenz and Chua attractors . . . . .	15
Figure 2.3	Hypernetwork with reduced dimensionality . . . . .	17
Figure 3.1	Lorenz locally mixed-coupled oscillators . . . . .	33
Figure 3.2	Stability Diagram for the Mixed Auxiliary System . . . . .	44
Figure 3.3	Stability diagrams for $X$ and $Y$ auxiliary systems . . . . .	46
Figure 3.4	Example of three oscillators coupled through $x$ and $y$ . . . . .	47
Figure 3.5	Three oscillators uniformly coupled through $x$ . . . . .	48
Figure 3.6	Ten Lorenz locally uniform coupled oscillators . . . . .	49
Figure 3.7	Ten Lorenz locally coupled oscillators . . . . .	49
Figure 3.8	Synchronization thresholds for ten Lorenz locally uniform/mixed coupled oscillators . . . . .	50
Figure 3.9	Six uniform coupled oscillators . . . . .	51
Figure 3.10	Depiction of six coupled oscillators replaced one at a time with $y$ -edges . . . . .	53
Figure 3.11	Depiction of six coupled oscillators replaced two at a time with $y$ -edges . . . . .	54
Figure 3.12	Depiction of six coupled oscillators successively replaced with $y$ -edges . . . . .	56
Figure 3.13	Lorenz and Chua's coupling strength vs traffic load of six coupled oscillators replaced from $x$ to $y$ -edges one at a time . . . . .	57
Figure 3.14	Lorenz and Chua's coupling strength vs traffic load of six coupled oscillators replaced from $x$ to $y$ -edges successively . . . . .	58
Figure 3.15	20 mixed coupled oscillators . . . . .	59

Figure 3.16 Numerics of one $y$ -edge replacement for 20 Lorenz and Chua oscillators . . . . .	60
Figure 3.17 Numerics of successive $y$ -edge replacement for 20 Lorenz and Chua oscillators . . . . .	61
Figure 3.18 Stability region with best fit curve . . . . .	62
Figure 3.19 Lorenz and Chua synchronization thresholds vs theoretical bounds	63
Figure 3.20 A 20-oscillator complex network with cycles . . . . .	65
Figure 3.21 20 Lorenz oscillators one edge and successive replacement . . . . .	66
Figure 4.1 One-hundred mixed coupled oscillators . . . . .	69

## Chapter 1

### INTRODUCTION

#### 1.1 Motivation and problem statement

Networks of dynamical systems are common models for many systems in physics, engineering, chemistry, biology, and the social sciences, including power grids, computer clocks, ecological and neuronal networks [3, 4, 5, 6, 7, 8]. The strongest form of network cooperative dynamics is synchronization, when the cells composing the network, evolve in unison, even though their initial conditions can be different. The phenomenon of synchronization, originally called “sympathy of clocks,” was first discovered by the great Dutch mathematician, astronomer, and horologist Christian Huygens in the 17th century. Huygens noticed that two oscillating pendulum clocks, suspended on a wooden beam, spontaneously become phase-locked and remain in this state. Synchronization was later discovered in many other systems where it plays an important role in the function or dysfunction of a wide spectrum of technological and biological networks [9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27].

Examples of synchronization in biological systems and networks include spontaneous synchronization of fireflies [9], synchronization of heart pacemaker cells and phase-locked rhythms in periodically stimulated heart cells [12, 13, 14], and synchronized locomotive patterns defined by symmetrical animal gaits [9, 10, 11]. In neuronal networks, synchronization has been broadly observed in pathological brain states, especially during epilepsy and Parkinson’s tremors [15, 16]. In engineering networks, synchronization of computer clocks over the Internet is a representative example. The proper functioning of a power-grid network also requires that its power generators remain synchronized. The disturbance of synchronization has been shown to be a cause of large power outages [22]. Synchronization of life cycles in ecological networks for a variety of species within metapopulations has been also reported

(for an example, see [28, 29, 30, 31, 32] and the references therein). In particular, life-cycle synchronization was observed in populations of moths and butterflies [33, 34], crabs [35], fish [28], birds [36] and lynx [37]. The first example of population synchrony dates back to the prominent study of fur returns of Canadian lynx to the Hudson Bay Company [37]. In most of the ecological networks, spatial synchrony is undesirable as it increases the danger of extinction, where an endangered population of one patch or forest cannot be saved by migration from another patch, which is also near an extinction state due to synchrony.

A great deal of attention has been focused on examining the relation between synchronization of oscillators composing a networks and the network topology [4, 6, 8, 38]. Obviously network topology plays a major role in network synchronization as densely coupled networks are typically easier to synchronize than sparse networks. In this regard, much attention has been paid to algebraic, statistical, and graph theoretical properties (see, for example, [3, 4, 5, 6, 7, 8, 39, 38, 40, 41, 42, 43, 44, 45, 46] and references therein). In particular, various stability methods have been developed to understand what factors are responsible for the onset of synchronization in a given network of oscillators [40, 41, 42, 45, 43, 44, 46, 47, 48, 49, 50, 51, 52].

Despite the vast literature to be found on network dynamics and synchronization, the majority of research activities have been focused on oscillators connected through one network (one type of coupling). However, in many realistic biological and engineering systems the units can be coupled via multiple, independent networks. Neurons are typically connected through mixed couplings such as excitatory, inhibitory, and electrical synapses, each corresponding to a different circuitry. In engineering systems, examples of mixed, independent networks include coupled grids of power stations and communication servers where the failure of nodes in one network may lead to the failure of dependent nodes in another network [53].

Typically, in networks of continuous time oscillators, the synchronous solution becomes stable when the coupling strength between oscillators exceeds a threshold value. This threshold depends on the individual oscillator dynamics and on the network topology. In this con-

text, a central question is to find the bounds on the coupling strengths so that the stability of synchronization is guaranteed. The Master Stability function (developed by Louis Pecora and Thomas Carroll) [40] and Connection Graph method (developed by my Ph.D. advisor, Igor Belykh, jointly with Vladimir Belykh and Martin Hasler) [42, 44, 45] are usually used to answer this question in networks with one connection graph. The Master Stability function relies on the calculation of the maximum Lyapunov exponent for the least stable transversal mode of the synchronous manifold along with the eigenvalues of the connectivity matrix. The Connection Graph method presents an alternate way to analyze synchronization conditions and requires no explicit knowledge of the spectrum of the connection matrix. It associates the Lyapunov function approach with graph theory. In this case, the main step is to calculate a bound on the sum of all paths passing through a given edge within the connection graph. Both methods reduce the dimensionality of the problem such that synchronization in a large complex network can be predicted from the dynamics of the individual node and the graph structure. Unfortunately, the Master Stability Function or its eigenvalue-based analogs [39] cannot be directly applied to network with mixed coupling, defined by two or more distinct graphs as it requires *simultaneous* diagonalization of two (or more) connectivity matrices. This is impossible in general unless the two matrices commute or one of them is a complete graph [26, 27]. A nice approach based on simultaneous block diagonalization of two connectivity matrices was proposed in [27]. This approach allows one to reduce the dimensionality of a large network to a smaller network whose synchronization condition can be used to evaluate the stability of synchronization in the large network. For some network topologies, typically possessing highly symmetrical, this technique yields a substantial reduction of the dimensionality; however, this reduction is less significant in general. The reduced network typically contains weighted positive and negative connections, including self loops such that the role of hyper-network topologies and the location of an edge remains difficult to evaluate.

This thesis addresses the challenging topic of synchronization in dynamical networks with mixed coupling and contributes toward the rigorous understanding of the emergence of stable synchronization. A mixed network is composed of subgraphs, connecting a subnetwork

of oscillators via one of the individual oscillator's variables. An illustrative example is a network of Lorenz systems with mixed couplings where some of the oscillators are coupled through the  $x$ -variable, some through the  $y$ -variable, and some through both. This thesis develops a new general synchronization method, called the Mixed Connection Graph method, which removes a long-standing obstacle to studying synchronization in mixed dynamical networks of a different nature. This method extends the Connection Graph method for one-component networks to mixed networks; however, this extension is not straightforward and requires overcoming a number of technically challenging issues as will be shown in Chapter 2.

The application of the method to specific networks reveals surprising, counterintuitive effects due to the mixed coupling. In particular, it shows that replacing a lightly loaded link with a stronger pairwise converging coupling (a "good" link) via another variable may improve synchronizability, as one would expect. At the same time, such a replacement of a highly loaded link may essentially worsen synchronizability and make the network unsynchronizable, turning the pairwise stabilizing "good" link into a trouble maker.

The layout of this thesis as follows. In the rest of this introductory chapter (Chapter 1), we review the existing method for synchronization in one-component networks. Chapter 2 contains the main theorem, which constitutes the Mixed Connection Graph method, and its derivation. In Chapter 3, we show how this method can be applied to specific networks. This application involves calculations of network traffic loads (these terms will be introduced later in Chapters 1 and 2). For large networks, these calculations require the development of several algebraic algorithms for calculating shortest paths. These algorithms have also been designed in this thesis. We also compare our theoretical and numerical results, which show an excellent match. Appendix A contains a proof of synchronization in two-node network of Lorenz systems with both  $x$  and  $y$  connections. This proof is used in the derivation of the main theorem in Chapter 2. Finally, Appendices B-E contain MATLAB and PYTHON codes for the algorithms used in the study of networks in Chapter 3.

## 1.2 Synchronization in one-component networks: A review

To review the synchronization existing methods, we first introduce the network model and discuss its properties in terms of the eigenvalues of the corresponding connectivity matrix.

### 1.2.1 Network model and type of synchronization considered

We consider the network of  $n$  identical oscillators that are linearly and mutually coupled

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{F}(\mathbf{x}_i) + \sum_{j=1}^n c_{ij} P(\mathbf{x}_j - \mathbf{x}_i), \quad i = 1, \dots, n, \quad (1.1)$$

Here,  $\mathbf{x}_i$  is the  $d$ -vector containing the coordinates of the  $i$ -th oscillator, and  $\mathbf{F}(\mathbf{x}_i)$  is a non-linear vector function which describes the oscillators' individual dynamics. The non-zero elements of the  $d \times d$  matrix  $P$  determine which variables couple the oscillators. In Chapter 2, we will refer to matrix  $P$  as the “inner” matrix. The oscillators are coupled via diffusive coupling which is a standard model, for example, for electrical coupling in engineering circuits and network, net migration between the patches in ecological networks where the migration is assumed to be proportional to the population size differences, and gap junctions in neuronal networks. As a result, the connectivity matrix  $C = (c_{ij})$  is a  $n \times n$  Laplacian matrix with zero row-sums and nonnegative off-diagonal elements. As the connections are mutual, the Laplacian matrix is symmetric such that the coupling strength  $c_{ij}=c_{ji}$  for all  $i$  and  $j$ . The matrix  $C$  defines a connected graph with  $n$  vertices and  $m$  edges. Connectivity matrix  $C$  has one zero eigenvalue, due to the zero row sum constrain,  $\lambda_1 = 0$  and  $(n - 1)$  negative eigenvalues  $\lambda_n < \dots < \lambda_3 < \lambda_2 < 0$ . Fig. 1.1(Top) displays an example of a five-node network (1.1) along with its connectivity matrix  $C$  and the eigenvalues. In general, the second largest eigenvalue  $\lambda_2$  is referred to as the algebraic connectivity and plays a pivotal role in the stability methods which follow.

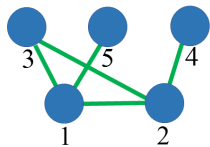


Figure 1.1: (Top). Example of five-node network in a one-component graph. (Bottom). The corresponding Laplacian matrix  $C$  with eigenvalues  $\lambda_1 = 0$ ,  $\lambda_2 = -\frac{5}{2} + \frac{\sqrt{13}}{2}$ ,  $\lambda_3 = -\frac{5}{2} + \frac{\sqrt{5}}{2}$ ,  $\lambda_4 = -\frac{5}{2} - \frac{\sqrt{5}}{2}$ , and  $\lambda_5 = -\frac{5}{2} - \frac{\sqrt{13}}{2}$ .

$$A = \begin{bmatrix} -3 & 1 & 1 & 0 & 1 \\ 1 & -3 & 1 & 1 & 0 \\ 1 & 1 & -2 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Complete synchronization in the network (1.1) is defined by the invariant hyperplane  $D = \{\mathbf{x}_1 = \mathbf{x}_2 = \dots = \mathbf{x}_n\}$ , called the synchronization manifold. The eigenvector associated with  $\lambda_1 = 0$  determines longitudinal direction along the synchronization manifold. All the other eigenvalues  $\lambda_2, \dots, \lambda_n$  are negative and define transversal directions and therefore determine the transversal stability of the synchronization manifold  $D$ . If the synchronization manifold is globally stable, i.e. any initial conditions eventually bring the trajectory to the manifold, then synchronization is globally stable (see the onset of synchronization in Fig.1.2). Based on their synchronizability, there are two main types of dynamical networks (1.1).

**Type I: Unbounded region of synchrony** [49]. In Type I networks, synchronization becomes globally stable when the coupling strength exceeds a threshold value and remains stable with further increased coupling. Therefore, the coupling range in which synchronization is globally stable is unbounded from the right such that this range is  $S_1 = (\alpha_1, \infty)$ , where  $\alpha_1$  is the threshold value of coupling. Once the threshold is reached, synchronization remains globally stable. Most dynamical networks belong to Type I, including Lorenz oscillators, double scroll oscillators and Hodgkin-Huxley neuron models [43].

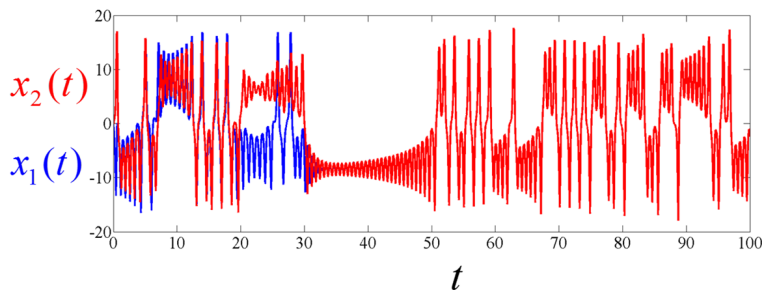


Figure 1.2: Complete synchronization between two Lorenz oscillators. Time-series indicate the onset of complete synchronization, arising from randomly chosen initial conditions.

**Type II: Bounded region of synchrony** ([40]). In Type II networks, synchronization never becomes globally stable. However, it can be stable within a limited bounded range,  $S_2 = (\alpha_2, \alpha_3)$ , where  $\alpha_2$  is the threshold value of coupling similar to  $\alpha_1$ , such that synchronization becomes locally stable when the coupling is sufficiently strong. However, synchronization loses its stability when the coupling is further increased and becomes stronger than  $\alpha_3$ . As a result, rigorous stability methods are only applicable to Type I networks. The method, which we develop in Chapter 2, can also handle only Type I networks; this point will be discussed in more detail in Chapter 2. Type II is a very narrow class of networks which includes  $x$ -coupled Rössler systems [42].

### 1.2.2 Existing eigenvalue-based stability methods

The main goal of a stability method for synchronization in a given network is to separate the role of the individual oscillator's dynamics and the contribution of network topology. This amounts to predicting the synchronization threshold in a large network, from the synchronization threshold in a two-node network, composed of the same oscillators and the properties of the connection matrix/graph. In this regard, the eigenvalue-based methods are based on the use of the eigenvalues of the connectivity matrix.

An important milestone in the research on network synchronization is the 1996 Wu-Chua conjecture [49]. Even though, it is not always true [40, 43] (more specifically, for Type II oscillators), it demonstrates how the synchronization threshold scales with the second largest eigenvalue  $\lambda_2$  of the connectivity matrix  $C$ .

**Wu-Chua conjecture [49].**

If  $c_{n_1}^*$  is the synchronization threshold in a network with  $n_1$  oscillators, then the synchronization threshold,  $c_{n_2}^*$ , in a similarly coupled network of size  $n_2$  satisfies the condition

$$c_{n_1}^* |\lambda_2(n_1)| = c_{n_2}^* |\lambda_2(n_2)|,$$

where  $\lambda_2(n_1)$  and  $\lambda_2(n_2)$  the second largest eigenvalues of the connectivity matrix  $C$  for the  $n_1$  and  $n_2$  networks, respectively. Backed up by a stability argument, this conjecture claims that the synchronization threshold,  $c_{network}^*$ , in a large network with a complex network structure, can be predicted from synchronization threshold,  $c_{12}^*$ , in the similarly coupled network of only two coupled oscillators. That is,

$$c_{network}^* = \frac{2c_{12}^*}{|\lambda_2|} = \frac{a}{|\lambda_2|},$$

as the absolute value of the second largest eigenvalue of the two-oscillator connectivity matrix is  $|\lambda_2(2)| = 2$ . Note that parameter  $a$  is the double synchronization threshold  $c_{12}^*$ . This parameter will reappear and be central for our stability studies, presented in Chapter 2.

This conjecture is based on a natural assumption that if the least stable transversal mode, determined by  $\lambda_2$  becomes stable, then all other transversal modes must also become stable. While this is true for a majority of dynamical networks (Type-I networks), this conjecture is not true for Type II networks, where increasing coupling destabilizes stability via other transversal modes.

To overcome this difficulty, Louis Pecora and Thomas Carroll of the Naval Research Laboratory developed a universal method called the **Master Stability Function** [40], which can handle both Type-I and Type-II networks and takes into account not only the second

eigenvalue  $\lambda_2$ , but the eigenratio,  $\lambda_2/\lambda_n$ , where  $\lambda_n$  is the smallest (most negative) eigenvalue of  $C$ . The main underlying idea is that the least and the most stable modes, associated by  $\lambda_2$  and  $\lambda_n$ , must be taken into account to guarantee the stability of synchronization.

This method is a semi-analytical as it is based on the numerical calculation of the transversal Lyapunov exponents and analytical calculations of  $\lambda_2$  and  $\lambda_n$ . Master Stability function is a state-of-the-art method which is widely used in the studies of synchronization (as of April 2016, the paper introduced the Master Stability function [40] and published in Physical Review Letters in 1998 has 1,634 citations).

### 1.2.3 Connection graph stability method

The eigenvalue-base methods do not explicitly indicate how a change of the network structure and the addition or removal of an edge affects synchronization, as this dependence is somewhat hidden in the change of  $\lambda_2$  and  $\lambda_n$ . They might also be non-applicable to networks with time-varying topologies.

A stability method, called the Connection Graph Stability Method, and developed by Igor Belykh, Vladimir Belykh and Martin Hasler in 2004 [42], proposes an alternative way of identifying the synchronization conditions. This method avoids the calculation the eigenvalues but instead relies on the identifying the corresponding sums of path lengths. Below, we briefly review the method and give two illustrative examples.

**Theorem 1.1 [Connection Graph Stability Method [42]]** (sufficient conditions). *Complete synchronization in the network (1.1) is stable if*

$$c_k > c^* = \frac{a_x}{n} b_k(n, m) \text{ for } k = 1, \dots, m, \quad (1.2)$$

where  $a_x = 2c_{12}^*$  is the double coupling strength sufficient for global stability of synchronization in two coupled oscillators. The quantity  $b_k(n, m) = \sum_{j>i; k \in P_{ij}}^n |P_{ij}|$  is the sum of the lengths of all chosen paths  $P_{ij}$  which pass through a given edge  $k$  that belongs to the connection graph. The uniform synchronization threshold can be achieved by  $c^* = \max_k \frac{a}{n} b_k$ .

To apply the method for a particular network, one should make two steps. First, one has to identify the synchronization threshold in the two-node network and obtain upper bounds on  $a$ . This is done by constructing using a Lyapunov function and applying it to a stability system, written for the difference variables (this point will be discussed in more detail in Chapter 2). Second, one should choose a path  $P_{ij}$  from any oscillator  $i$  to any other oscillator  $j$  and then determine  $b_k$  for all edges  $k$  on the connection graph. Examples below show how to calculate  $b_k$ .

**Example 1.** [Star configuration]. Consider a star network configuration of Fig. 1.3. To

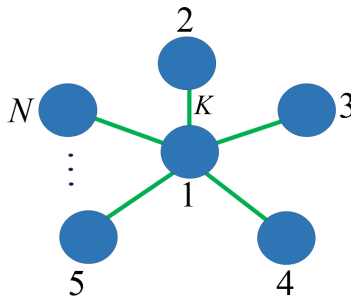


Figure 1.3: Star configuration composed of  $N$  oscillators.

identify  $b_k$  for edge  $k$ , we choose a path from node 2 to nodes 1, 3, ...,  $N$ . As this graph is a tree, the choice of paths is unique. We then determine the lengths of the paths (the number of edges comprising each path) and sum them up:

$$b_K = P_{21} + P_{23} + P_{24} + \cdots + P_{2N} = 1 + 2 + 2 + \cdots + 2 = 1 + \underbrace{2 + 2 + \cdots + 2}_{N-2} = 2N - 3$$

Therefore, the bound on the synchronization threshold on the coupling strength for edge  $k$  is  $c_k > a \cdot \frac{2n-3}{n}$ . As the coupling is homogeneous and edges are topologically identical, this bound becomes the synchronization threshold for the entire network.

Evidently, the bound (1.2) depends on the choice of the paths  $P_{ij}$ . In most cases, one should choose a shortest path from vertex  $i$  to vertex  $j$ . However, in some networks, a different

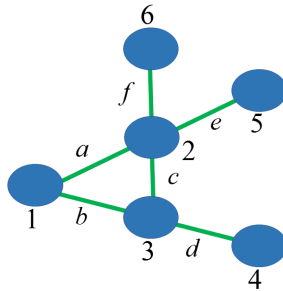


Figure 1.4: Example of a network for which the choice of the shortest paths between the oscillators is not optimal for calculating  $b_k$ .

choice of paths can lead to lower bounds [43]. The following example from [43] supports this claim.

**Example 2.** [Are shortest paths always optimal?]. Consider the network shown in Fig. 1.4 and calculate  $b_k$ . This graph is not a tree, therefore, we can choose paths differently. Let's try the following choice of paths between the nodes:  $P_{12} = a$ ,  $P_{13} = b$ ,  $P_{14} = bd$ ,  $P_{15} = ae$ ,  $P_{16} = af$ ,  $P_{23} = c$ ,  $P_{24} = cd$ ,  $P_{25} = e$ ,  $P_{26} = f$ ,  $P_{34} = d$ ,  $P_{35} = ce$ ,  $P_{36} = cf$ ,  $P_{45} = dce$ ,  $P_{46} = dcf$ ,  $P_{56} = fe$ .

Calculate the sum of path lengths passing through edges

$$a : b_a = |P_{12}| + |P_{15}| + |P_{16}| = 1 + 2 + 2 = 5$$

$$b : b_b = |P_{13}| + |P_{14}| = 1 + 2 = 3$$

$$c : b_c = |P_{23}| + |P_{24}| + |P_{35}| + |P_{36}| + |P_{45}| + |P_{46}| = 13$$

$$d : b_d = |P_{14}| + |P_{24}| + |P_{34}| + |P_{45}| + |P_{46}| = 11$$

$$e : b_e = |P_{15}| + |P_{25}| + |P_{35}| + |P_{45}| + |P_{56}| = 10$$

$$f : b_f = |P_{16}| + |P_{26}| + |P_{36}| + |P_{46}| + |P_{56}| = 10.$$

Observe that  $b_c = 13$  has the maximum “load,” in terms of the length of chosen “roads” that pass through this edge (edge  $c$ ). Therefore, this bound  $b_c = 13$  will yield the upper bound for the synchronization threshold (1.2). Clearly, this edge is a bottle neck for the entire network. In terms of traffic networks, one choice of shortest paths dictates most drivers to

drive through this edge, potentially causing a traffic jam. For those familiar with traffic in the Atlanta area, this edge  $b_c$  may be interpreted as Spaghetti Junction or Jimmy Carter Boulevard.

What if some drivers decide to bypass this “street”? It will decrease the traffic load through the bottle neck connection, at the expense of increased traffic elsewhere. We adopt this idea to lower the upper synchronization bound and change  $P_{23}$  from  $c$  to  $ab$  and recalculate the sum of path lengths (this change only affects edges  $a$ ,  $b$ , and  $c$ ):

$$a : b_a = |P_{12}| + |P_{15}| + |P_{16}| + |P_{23}| = 7$$

$$b : b_b = |P_{13}| + |P_{14}| + |P_{23}| = 5$$

$$c : b_c = |P_{24}| + |P_{35}| + |P_{36}| + |P_{45}| + |P_{46}| = 12.$$

Notice that load on edge  $c$  has reduced to  $b_c = 12$  which gives a lower synchronization bound. Notice that  $b_a = 7$  and  $b_b = 5$  have increased from 5 and 3, respectively; however, this increase does not affect the upper bound. Changing  $P_{36}$  from  $cf$  to  $abf$ , yields even lower bounds:  $b_a = 10$ ;  $b_b = 8$ ,  $b_c = 10$ ,  $b_f = 11$ , making the synchronization threshold (1.2):  $c^* = a \cdot 11/6$ . Finding the best distribution of the chosen paths amount to solving an optimization problem.

The Connection Stability Graph method is typically used for analytical studies of global synchronization in dynamical networks (the paper introduced the Connection Graph method [42] was the top cited paper of *Physica D* (published in the five year period, 2004-2009). As of April 2016, it has a total of 377 citations). Originally developed for undirected networks, it was later extended to directed networks where one has to symmetrize the connections graph and assign additional weights to connections and path lengths to derive the synchronization condition [45].

In the following, we will show that this method can be extended to handle synchronization in networks with mixed couplings which is out of reach of the eigenvalue methods. This extension is highly non-trivial and requires finding a way around a few serious obstacles.

## Chapter 2

### SYNCHRONIZATION IN MIXED NETWORKS: THE MIXED GRAPH CONNECTION METHOD

This chapter contains the main theoretical result of this thesis.

#### 2.1 Network model and problem statement

We consider a general mixed (hyper-) network of  $n$  oscillators with two different connection graphs via two different variables:

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{F}(\mathbf{x}_i) + \sum_{j=1}^n c_{ij} P(\mathbf{x}_j - \mathbf{x}_i) + \sum_{j=1}^n d_{ij} L(\mathbf{x}_j - \mathbf{x}_i), \quad i = 1, \dots, n, \quad (2.1)$$

where  $\mathbf{x}_i = (x_i^1, \dots, x_i^d)$  is the  $d$ -vector containing the coordinates of the  $i$ -th oscillator,  $\mathbf{F}_i : \mathbb{R}^d \rightarrow \mathbb{R}^d$  describes the oscillators' individual dynamics,  $c_{ij}$  and  $d_{ij}$  are the coupling strength.  $C = (c_{ij})$  and  $D = (d_{ij})$  are the  $n \times n$  connectivity matrices, defining two different connection graphs (also denoted by  $C$  and  $D$ , with  $m$  and  $l$  edges, respectively) (see Fig. 2.1 for an example of a combined connection graph). The notations are similar to those of (1.1). The inner matrices  $P$  and  $L$  determine which variables couple the oscillators within the  $C$  and  $D$  graphs, respectively. Without loss of generality, in the following we will consider the oscillators with dimension  $d = 3$  and  $\mathbf{x}_i = (x_i, y_i, z_i)$ . Therefore, the  $C$  graph with the inner matrix  $P = \text{diag}(1, 0, 0)$  corresponds to connections via  $x$ , while the  $D$  graph with the inner matrix  $L = \text{diag}(0, 1, 0)$  indicates connections via  $y$ . Overall, the oscillators of the network are connected through a combination of the two graphs. The graphs are assumed to be undirected [42]. The extension of our results to directed graphs can be performed, by modifying the Generalized Connection Graph method [45, 44] with only moderate effort. This extension is not presented in this thesis and represents a subject of future study.

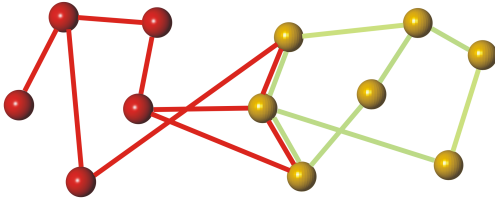


Figure 2.1: Example of a hyper-network with two connection graphs. The red (dark) and yellow (light) edges represent the coupling through  $x$  and  $y$ -coupling, respectively.

Oscillators, comprising the network (2.1), can be periodic or chaotic. As chaotic oscillators are difficult to synchronize, they are usually used as test bed examples for probing the effectiveness of a given stability approach. The oscillators used in the numerical verification of our analytical results in Chapter 3 are Lorenz and Chua oscillators (see 2.2 and 2.3). For the Lorenz and Chua equations, the vector equation (2.1) can be written in a more reader-friendly scalar form such the network (2.1) with the Lorenz oscillator as the individual unit reads

$$\begin{aligned}
 \dot{x}_i &= \sigma(y_i - x_i) + \sum_{j=1}^n c_{ij}x_j, \\
 \dot{y}_i &= rx_i - y_i - x_iz_i + \sum_{j=1}^n d_{ij}y_j, \\
 \dot{z}_i &= -bz_i + x_iy_i, \quad i = 1, \dots, n.
 \end{aligned} \tag{2.2}$$

Similarly, the network model (2.1), comprised by the Chua oscillators [49], takes the following form

$$\begin{aligned}
 \dot{x}_i &= \alpha(y_i - x_i - h(x)) + \sum_{j=1}^n c_{ij}x_j, \\
 \dot{y}_i &= x_i - y_i + z_i + \sum_{j=1}^n d_{ij}y_j, \\
 \dot{z}_i &= -\beta y_i - \gamma z_i, \quad i = 1, \dots, n,
 \end{aligned} \tag{2.3}$$

with

$$h(x) = \begin{cases} m_1(x + 1) - m_0 & x < -1 \\ m_0x & -1 \leq x \leq 1 \\ m_1(x - 1) + m_0 & x > 1 \end{cases}$$

The Lorenz oscillator was proposed by a pioneer of chaos theory, a MIT Professor of Mathematics and Meteorology, Edward N. Lorenz. This was a first example of a deterministic system which was capable of exhibiting aperiodic (chaotic) behavior. For the standard

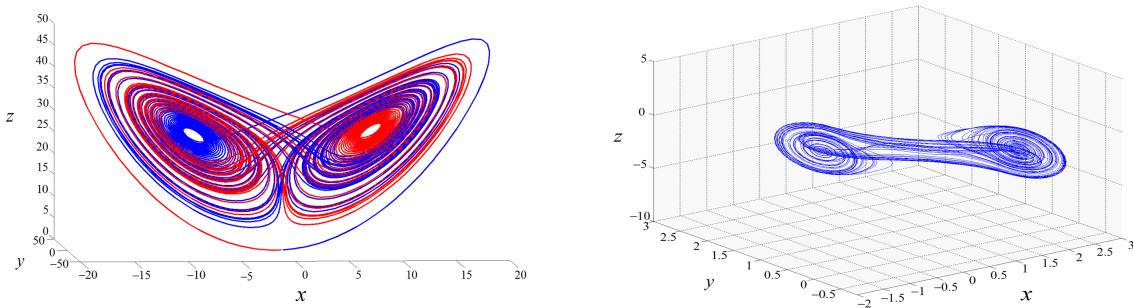


Figure 2.2: (Left). Lorenz attractor exhibiting a chaotic behavior. Parameters are  $r = 28$ ,  $b = \frac{8}{3}$ ,  $\alpha = 10$ . (Right). Double-scroll attractor in the Chua system. Parameters are  $\alpha = 10$ ,  $\beta = 15$ ,  $\gamma = 0.0385$ ,  $m_0 = -1.27$  and  $m_1 = -0.68$ .

parameters  $r = 28$ ,  $b = \frac{8}{3}$ ,  $\alpha = 10$ , the individual Lorenz systems exhibits chaotic behavior, identified as the Lorenz attractor (see Fig. 2.2).

The Chua system was developed by Leon O. Chua, Professor of Electrical Engineering at the University of California, Berkeley. The Chua system is one of the most known electrical circuits that exhibit chaotic behavior. As the attractor of the Chua system with the standard parameters  $\alpha = 10$ ,  $\beta = 15$ ,  $\gamma = 0.0385$ ,  $m_0 = -1.27$  and  $m_1 = -0.68$  has a double-scroll shape, this system is often called the Double-Scroll oscillator. We will also be using this code name for the Chua system throughout this thesis.

Returning to the general network model (2.1), we seek to develop a general, rigorous method for determining the conditions on the network architecture of the two connection graphs, coupling strength  $c_i, d_i$ , and intrinsic parameters of the individual oscillator, comprising the network, that guarantee stable synchronization. As discussed in Chapter 1, this is a highly nontrivial problem as the available eigenvalue methods, including the Master Stability function [40], can not be directly applied to handle the stability of synchronization in the mixed network (2.1). This is due to the fact that the eigenvalue methods allow for the reduction of the stability description by means of the eigenvalues of the connectivity matrix; however, in the case of mixed networks, defined by two matrices, this requires simultaneous diagonalization of two connectivity matrices. Obviously, this is impossible in general

unless the two matrices commute or one of them is a complete graph [26, 27]. The following subsection gives more details on previous attempts towards a rigorous understanding of synchronization in the mixed network (2.1).

## 2.2 Simultaneous Block Diagonalization and Reduction Dimensionality

A nice approach based on simultaneous block diagonalization of two connectivity matrices was proposed in [27]. This approach allows one to reduce the dimensionality of a large network to a smaller network whose synchronization condition can be used to evaluate the stability of synchronization in the large network. For some network topologies, this technique yields a substantial reduction of the dimensionality; however, this reduction is less significant in general. The reduced network typically contains weighted positive and negative connections, including self loops such that the role of hypernetwork topologies and the location of an edge remains difficult to evaluate.

Figure 2.3 indicates the results of the application of the simultaneous block diagonalization method to three mixed networks. The reduction was performed using a Matlab code and algorithm, provided by Professor Francesco Sorrentino, the senior author of [27]. In Fig. 2.3, the blue oscillator represents coupling through  $x$ , while the yellow oscillator represents coupling through  $y$  and the empty oscillator represent coupling through both  $x$  and  $y$ . The green and red edges denote  $x$  and  $y$  coupling, respectively. This dimensionality reduction implies that the stability of synchronization in the original networks (Fig. 2.3), the networks on the left) can be assessed through the stability of the reduced networks (Fig. 2.3), the networks on the right). Unfortunately, this reduction in these three networks does not the stability argument any simpler. For example, the original three-oscillator network with uniform coupling strength of the  $x$ - and  $y$ - coupling (Fig. 2.3(top) ) reduces to the two-oscillator network, which is a desirable reduction, but with weighted negative coefficients and additional self-loops. As a result, this reduction seems to be ineffective for identifying the stability conditions in these specific examples.

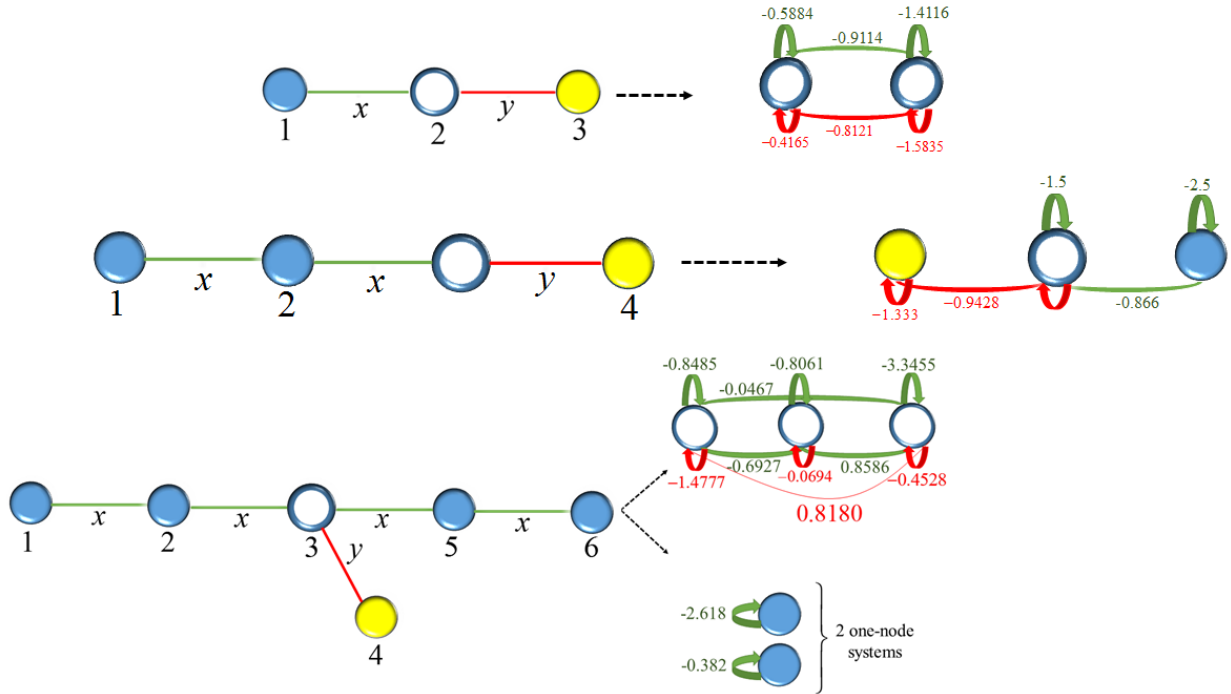


Figure 2.3: Simultaneous Block Diagonalization. (Top) Three-oscillator network with uniform coupling reduces to a two-oscillator network with weighted couplings and self-loops. The colors of the  $x$ - and  $y$ - connections in the original network identify the corresponding type of connection ( $x$  vs.  $y$ ) in the reduced network. (Middle) Four-oscillator network reduces to three coupled oscillators with self loops. (Bottom) Six-oscillator network reduces to a combination of a three- and two-oscillator network such that the stability of synchronization in the six-oscillator network must be assessed via the combined stability conditions for the two reduced networks.

Although, there are examples of network topologies where these reductions are more significant, the stability problem of synchronization in mixed network (2.1) calls for more effective stability methods. One such method, which we have called the Mixed Graph Stability Method, is developed in the next section.

### 2.3 Mixed Graph Stability Method: The General Proof

Our main goal is to extend the Connection Graph Stability method to prove global stability of synchronization in the general mixed network (2.1). This extension is not straightforward as will be seen in what follows.

As in the case of dynamical network with one component graphs, complete synchronization in the mixed network (2.1) is determined by an invariant hyperplane, called the synchronization manifold,  $M = \{\mathbf{x}_1 = \mathbf{x}_2 = \dots = \mathbf{x}_n\}$ . This invariant manifold has dimension  $d$ , which in the case of the 3-D Lorenz and Chua systems equals 3.

Our aim is to derive conditions of global asymptotic stability of the synchronization manifold in the system (2.1). We seek to identify threshold coupling values which are sufficient for global stability of synchronization, and to reveal their dependence on the structure of the mixed connection graph and the intrinsic oscillators' properties. To achieve this goal and develop the stability method, we shall follow the steps of the proof of the Connection Graph Method [42]. The concept will be similar, up to a certain step where a new stability argument will have to be implemented.

The formulation of our main result (Theorem 2.1) is technical and contains notions which are hard to grasp without a proper introduction. Therefore, we prefer to walk the reader through the derivation of the proof first, finally arriving the main conclusion, formulated as Theorem 2.1.

### 2.3.1 Stability System for the Difference Variables

In the network model (2.1) we introduce the difference variables,

$$X_{ij} = x_j - x_i, \quad i, j = 1, \dots, n, \quad (2.4)$$

whose convergence to zero will imply the transversal stability of the synchronization manifold  $M(3)$ .

Subtracting the  $i$ -th equation from the  $(i + 1)$ -th equation in system (2.1), we obtain the equations for the transversal stability of  $M(3)$  :

$$\dot{X}_{ij} = F(x_j) - F(x_i) + \sum_{k=1}^n \{c_{jk}PX_{jk} - c_{ik}PX_{ik} + d_{jk}LX_{jk} - d_{ik}LX_{ik}\}, \quad i, j = 1, \dots, n. \quad (2.5)$$

To obtain the explicit presence of  $X_{ij}$  in  $F(x_j) - F(x_i)$ , we introduce the following vector notation

$$F(x_j) - F(x_i) = \left[ \int_0^1 DF(\beta x_j + (1 - \beta)x_i) d\beta \right] X_{ij},$$

where  $DF$  is a  $d \times d$  Jacobian matrix of  $F$ . This notation is simply a compact form of the Mean Value Theorem,  $f(A) - f(B) = f'(C)(B - A)$ , to the vector functions  $F(x_j)$  and  $F(x_i)$ , where the Jacobian  $DF$  is evaluated at some point  $C \in [X_i, X_j]$ .

Therefore, the difference system (2.5) can be rewritten in the form

$$\dot{X}_{ij} = \left[ \int_0^1 DF(\beta x_j + (1 - \beta)x_i) d\beta \right] X_{ij} + \sum_{k=1}^n \{c_{jk}PX_{jk} - c_{ik}PX_{ik} + d_{jk}LX_{jk} - d_{ik}LX_{ik}\},$$

$$i, j = 1, \dots, n. \quad (2.6)$$

The first term with the brackets yields instability via the divergence of trajectories within the individual chaotic oscillators. The second (summation) term, which represents the contribution of the network connections, may overcome the unstable term, provided that the coupling is strong enough.

Notice that the stability (2.5) is redundant as it contains all possible  $(n-1)n/2$  non-zero differences  $X_{ij}$  along with  $n$  zero differences  $X_{ii} = 0$  which can be disregarded. As the same time, there are only  $(n-1)$  linearly independent differences, required to show the convergence between  $n$  variables  $X_{ij}$ . However, this redundancy property and the consideration of all non-zero  $X_{ij}$  are a key ingredient of our approach which allows for separating the difference variables later in the stability description, without diagonalizing the connectivity matrices.

We strive to find conditions under which the trivial fixed point  $\{X_{ij} = 0, i, j = 1, \dots, n\}$  of system (2.5) is globally stable. This amounts to finding conditions for global stability of the synchronization manifold  $M(3)$  in the network model (2.1).

We introduce the following terms  $A_{ij}X_{ij}$ , where  $A$  is a  $3 \times 3$  matrix, such that

$$A_{ij} = \begin{cases} a_x P & = \text{diag}(a_x, 0, 0) \text{ if oscillators } i \text{ and } j \text{ belong to } x\text{-graph } C \\ a_y L & = \text{diag}(0, a_y, 0) \text{ if oscillators } i \text{ and } j \text{ belong to } y\text{-graph } D \\ K & = \text{diag}(\omega_x, \omega_y, 0) \text{ if oscillators } i \text{ and } j \text{ belong to different graphs, } C \text{ and } D, \end{cases} \quad (2.7)$$

where constants  $a_x, a_y, \omega_x$ , and  $\omega_y$  are to be determined.

We add and subtract additional terms  $A_{ij}X_{ij}$  with matrix  $A_{ij}$  defined in (2.7) from the stability system (2.6) and obtain

$$\dot{X}_{ij} = \left[ \int_0^1 DF(\beta x_j + (1 - \beta)x_i) d\beta - A_{ij} \right] X_{ij} + A_{ij} X_{ij} + \sum_{k=1}^n \{c_{jk} P X_{jk} - c_{ik} P X_{ik} + d_{jk} L X_{jk} - d_{ik} L X_{ik}\}, \quad (2.8)$$

The introduction of new terms  $A_{ij}X_{ij}$  allows for obtaining stability conditions of the trivial fixed point  $X_{ij} = 0$ ,  $i, j = 1, \dots, n$  in two steps. Note that the matrix  $-A$  contributes into the stability of the fixed point and can compensate for instabilities induced by eigenvalues with nonnegative real parts of the Jacobian  $DF$ . This can be achieved by increasing parameters  $a_x, a_y, \omega_x$ , and  $\omega_y$ . At the same time, the instability originated from its positively definite counterpart, matrix  $+A$ , can be damped by the coupling terms with  $c_{ij}$  and  $d_{ij}$ .

*Step I.* We make the first step by introducing the following auxiliary systems

$$\dot{X}_{ij} = \left[ \int_0^1 DF(\beta x_j + (1 - \beta)x_i) d\beta - A_{ij} \right] X_{ij}, \quad i, j = 1, \dots, n. \quad (2.9)$$

This system is identical to system (2.8) where the coupling terms are removed.

The values  $A_{ij}$  take three different values, depending on whether oscillators  $i$  and  $j$  both belong to  $x$ - or  $y$ -graphs, or belong to the different graphs (see (2.7)). Therefore, we have

three types of the auxiliary systems

$$\dot{X}_{ij} = \left[ \int_0^1 DF(\beta x_j + (1 - \beta)x_i) d\beta - a_x P \right] X_{ij} \quad \text{if } i \text{ and } j \text{ belong to } x\text{-graph}, \quad (2.10)$$

$$\dot{X}_{ij} = \left[ \int_0^1 DF(\beta x_j + (1 - \beta)x_i) d\beta - a_y L \right] X_{ij} \quad \text{if } i \text{ and } j \text{ belong to } y\text{-graph}, \quad (2.11)$$

$$\dot{X}_{ij} = \left[ \int_0^1 DF(\beta x_j + (1 - \beta)x_i) d\beta - K \right] X_{ij} \quad \text{if } i \text{ and } j \text{ belong to different graphs.} \quad (2.12)$$

Remarkably, the auxiliary system (2.10) coincides with the difference system for the global stability of synchronization in the two-oscillator network (2.8) with only  $x$ - coupling, where  $a_x$  plays the role of the double coupling strength that guarantees the stability (see [42] for a detailed discussion on this relation).

Similarly, the stability of auxiliary system (2.11) implies global stability of synchronization in the two-oscillator network (2.1) with only  $y$ - coupling, where  $a_y$  is the double coupling strength of the  $y$  connection. Lastly, the stability of auxiliary system (2.1) guarantees globally stable synchronization in the two-oscillator network with both  $x$ - and  $y$ - coupling, where a combination constant of  $\omega_x$ , and  $\omega_y$ , present in  $K$ , is a combination of the double coupling strengths of  $x$  and  $y$  connection that is sufficient to induce stable synchronization.

Therefore, our immediate goal is to find upper bounds on the values of  $a_x$ ,  $a_y$ , and  $\omega_x$  and  $\omega_y$  (the latter two are present in  $K$ ) that make the origin of the auxiliary systems (2.10) - (2.12). This amounts to proving global synchronization in the three ( $x,y$ , and  $(x,y)$ ) coupled networks that are composed of two oscillators. It is important to emphasize that only Type-I oscillators (see Chapter 1 for the details) are capable of synchronizing globally and retaining synchronization for couplings exceeding the synchronization threshold. Most known oscillators, including the Lorenz and Chua oscillators, which will be used further in Chapter 3 to discuss the implications of our general method, belong to Type-I systems. A much narrow Type-II class of systems contains  $x$ -coupled Rössler systems [42] in which

synchronization is only locally stable, and the systems destabilize with an increase of coupling such that the bounds  $a_x$ ,  $a_y$ , and  $\omega_x$  and  $\omega_y$  do not exist. Hence, we limit our consideration to Type I oscillators.

The proof of global stability in (2.10) - (2.12) and derivation of bounds  $a_x$ ,  $a_y$ , and  $\omega_x$  and  $\omega_y$  involves the construction of a Lyapunov function along with the assumption of the eventual dissipativeness of the coupled system. Therefore, before advancing with the study of larger networks (2.1), one has to prove that globally stable synchronization in the simplest ( $x, y$ , and  $(x, y)$ ) coupled, two-oscillator networks is achievable. The bound  $a_x$  for the  $x$ -coupled Lorenz oscillators was given in [42], while the detailed derivation of bounds  $a_y$ , and  $\omega_x$  and  $\omega_y$  is presented in Appendix A.

Having obtained the bounds  $a_x$ ,  $a_y$ , and  $\omega_x$  and  $\omega_y$ , and therefore proving the stability of the auxiliary systems (2.10) - (2.12), we can now make the second step in analyzing the full stability system (2.8).

*Step II.* The bounds  $a_x$ ,  $a_y$ , and  $\omega_x$  and  $\omega_y$  that stabilize the auxiliary systems (2.10) - (2.12) reduce the stability analysis of system (2.8) to the following equations, by excluding the term in the brackets:

$$\dot{X}_{ij} = A_{ij}X_{ij} + \sum_{k=1}^n \{c_{jk}PX_{jk} - c_{ik}PX_{ik} + d_{jk}LX_{jk} - d_{ik}LX_{ik}\}, \quad i, j = 1, \dots, n. \quad (2.13)$$

Note that the positive term  $A_{ij}X_{ij}$ , which contains the upper bounds  $a_x$ ,  $a_y$ , and  $\omega_x$  and  $\omega_y$ , is destabilizing and must be compensated by the coupling terms. To study the stability of (2.13), we introduce Lyapunov functions of the form

$$V = \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n X_{ij}^T \cdot I \cdot X_{ij}, \quad (2.14)$$

where  $I$  is a  $3 \times 3$  identity matrix (recall that the dimension of the oscillators, comprising the network, is assumed to be 3).

Its time derivative with respect to system (2.13) becomes

$$\begin{aligned} \dot{V} = & \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n X_{ij}^T A_{ij} X_{ij} - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \{c_{jk} X_{jk}^T IPX_{jk} - c_{ik} X_{ik}^T IPX_{ik}\} - \\ & - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \{d_{jk} X_{jk}^T ILX_{jk} - d_{ik} X_{ik}^T ILX_{ik}\}. \end{aligned} \quad (2.15)$$

We need to demonstrate the negative definiteness of the quadratic form  $\dot{V}$ . As ( $X_{ii}^2 = 0$ ,  $X_{ij}^2 = X_{ji}^2$ ), the first sum simplifies to

$$S_1 = \sum_{i=1}^{n-1} \sum_{j>i}^n A_{ij} X_{ij}^2. \quad (2.16)$$

This sum is always positive definite and whose contribution must be compensated for by the second and third sums

$$\begin{aligned} S_2 = & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \{c_{jk} X_{jk}^T IPX_{jk} - c_{ik} X_{ik}^T IPX_{ik}\}, \\ S_3 = & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \{d_{jk} X_{jk}^T ILX_{jk} - d_{ik} X_{ik}^T ILX_{ik}\}. \end{aligned} \quad (2.17)$$

The two terms in both  $S_2$  and  $S_3$  can be made identical due to the coupling symmetry, by exchanging the indices  $i$  by  $j$  in the second terms such that

$$\begin{aligned} S_2 = & -\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n c_{jk} X_{jk}^T IPX_{jk}, \\ S_3 = & \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n d_{jk} X_{jk}^T ILX_{jk}. \end{aligned} \quad (2.18)$$

Since  $X_{jj} = 0$ , we obtain

$$\begin{aligned} S_2 = & -\sum_{i=1}^n \sum_{k=1}^{n-1} \sum_{j>k}^n c_{jk} X_{ji}^T IPX_{jk} - \sum_{i=1}^n \sum_{k=1}^{n-1} \sum_{j<k}^n c_{jk} X_{ji}^T IPX_{jk}, \\ S_3 = & -\sum_{i=1}^n \sum_{k=1}^{n-1} \sum_{j>k}^n d_{jk} X_{ji}^T IPX_{jk} - \sum_{i=1}^n \sum_{k=1}^{n-1} \sum_{j<k}^n d_{jk} X_{ji}^T IPX_{jk}. \end{aligned} \quad (2.19)$$

Again, exchanging  $j$  and  $k$  in the second terms of  $S_2$  and  $S_3$  and implying the symmetries of coupling  $c_{jk} = c_{kj}$  and  $d_{jk} = d_{kj}$ , we obtain

$$\begin{aligned}
S_2 &= - \sum_{i=1}^n \sum_{k=1}^{n-1} \sum_{j>k}^n c_{jk} X_{ji}^T IPX_{jk} - \sum_{i=1}^n \sum_{j=1}^{n-1} \sum_{k<j}^n c_{jk} X_{ki}^T IPX_{kj} = \\
&= - \sum_{i=1}^n \sum_{k=1}^{n-1} \sum_{j>k}^n c_{jk} (X_{ji}^T + X_{ik}^T) IPX_{jk}, \\
S_3 &= - \sum_{i=1}^n \sum_{k=1}^{n-1} \sum_{j>k}^n d_{jk} X_{ji}^T ILX_{jk} - \sum_{i=1}^n \sum_{j=1}^{n-1} \sum_{k<j}^n d_{jk} X_{ki}^T ILX_{kj} = \\
&= - \sum_{i=1}^n \sum_{k=1}^{n-1} \sum_{j>k}^n d_{jk} (X_{ji}^T + X_{ik}^T) ILX_{jk}.
\end{aligned} \tag{2.20}$$

$X_{ji}^T + X_{ik}^T = [x_i^T - x_j^T + x_k^T - x_i^T] = X_{jk}^T$  such that

$$\begin{aligned}
S_2 &= - \sum_{i=1}^n \sum_{k=1}^{n-1} \sum_{j>k}^n c_{j,k} X_{jk}^T IPX_{jk} = - \sum_{k=1}^{n-1} \sum_{j>k}^n n c_{jk} X_{jk}^T IPX_{jk}. \\
S_3 &= - \sum_{i=1}^n \sum_{k=1}^{n-1} \sum_{j>k}^n d_{jk} X_{j,k}^T ILX_{jk} = - \sum_{k=1}^{n-1} \sum_{j>k}^n n d_{jk} X_{jk}^T ILX_{jk}.
\end{aligned} \tag{2.21}$$

Returning to the derivation of the Lyapunov function (2.15) and combining the sums  $S_1$ ,  $S_2$  and  $S_3$  yields the condition which guarantees that  $\dot{V} < 0$ :

$$S_1 + S_2 + S_3 = \sum_{i=1}^{n-1} \sum_{j>i}^n X_{ij}^T I [A_{ij} - n c_{ij} P - n d_{ij} L] X_{ij} < 0 \tag{2.22}$$

The most remarkable property of this condition is that we were able to get rid of cross terms and formulate the condition in terms of  $X_{ij}$ . This is due to the fact that we chose to consider the redundant system with all possible differences  $X_{ij}$ , including linearly dependent ones.

The condition (2.21) finally transforms into

$$n \sum_{i=1}^{n-1} \sum_{j>i}^n [c_{ij} X_{ij}^T IPX_{ij} + d_{ij} X_{ij}^T ILX_{ij}] > \sum_{i=1}^{n-1} \sum_{j>i}^n X_{ij}^T I A_{ij} X_{ij}. \tag{2.23}$$

Notice that the left-hand side (LHS) of this inequality contains only the differences  $X_{ij}$  between the oscillators that belong to the edges on the connection graphs: the first term on the LHS corresponds to the  $x$ -graphs and the second terms are defined by the edges of the

$y$ -graph. At the same time, the differences on the right-hand side of (2.23) correspond to all possible differences between pairs of oscillators that might or might not be defined by edges of the connection graphs. Therefore, to get rid of the presence of the differences  $X_{ij}$  and therefore find the conditions explicit in the parameters of the network model (2.1), we need to express the differences on the RHS via the differences on the LHS such that we will be able to cancel them.

So far, we have been closely following the steps in the derivation of the Connection Graph method [42] for networks with only one connection graph, connecting all  $n$  oscillators. The inequality (2.23) is similar to that of the Connection Graph method, except for the presence of the second term on the RHS. A new non-trivial observation, however, is that the total number of oscillators  $n$  in the network (2.1), composed of two connection graphs appears as a factor in both sums on the RHS, corresponding to the  $x$ - and  $y$ -graphs, even though each graph itself connects fewer oscillators. The stability argument which follows drastically differs from that of the Connection Graph method.

Denote on the LHS (2.23), the differences  $X_{ij}$  corresponding edges of the  $x$ -graph by  $\tilde{X}_k$ ,  $k = 1, \dots, m$  and the differences  $X_{ij}$  corresponding edges of the  $y$ -graph by  $\tilde{Y}_k$ ,  $k = 1, \dots, l$ . Recall that  $m$  and  $l$  are the number of edges on the  $x$ - and  $y$ -graphs, respectively. In addition, let  $X_k$  be a scalar from the vector  $\tilde{X}_k$  which indicates the scalar difference between  $x_i$  and  $x_j$ , corresponding to an edge on the  $x$ -graph. Similarly, let  $Y_k$  be a scalar from the vector  $\tilde{Y}_k$ , defined by the corresponding  $y_i$  and  $y_j$ . Following these notations, the differences  $X_{ij}$  on the RHS will now define the scalars  $X_{ij} = x_j - x_i$  and  $Y_{ij} = y_j - y_i$ , where we abuse the notation  $X_{ij}$ , originally used for the vector difference. If the inequality (2.23) is satisfied in terms of  $x_i$  and  $y_i$  (via the scalar differences  $X_k, Y_k$ ), then it will also be satisfied for the remaining scalar  $z_i$ . Recall that  $(x_i, y_i, z_i)$  are the scalar coordinates of the individual oscillator, composing the network (2.1).

Using these notations, we can rewrite (2.23) as follows

$$\begin{aligned}
n \sum_{k=1}^m c_k X_k^2 + n \sum_{k=1}^l d_k Y_k^2 > a_x \sum_{i=1}^{n-1} \sum_{j>i, (i,j) \in C} X_{ij}^2 + a_y \sum_{i=1}^{n-1} \sum_{j>i, (i,j) \in D} Y_{ij}^2 + \\
\sum_{i=1}^{n-1} \sum_{j>i, i \in (C,D), j \in (D,C)} [\omega_x X_{ij}^2 + \omega_y Y_{ij}^2],
\end{aligned} \tag{2.24}$$

where  $c_k = c_{i_k j_k}$  and  $d_k = d_{i_k j_k}$ . Here, the RHS of (2.24) has three terms, obtained by splitting the difference variables into three groups, according to the coefficients of  $A_{ij}$  (cf. (2.23) and (2.10)-(2.12)). The first sum on the LHS is composed of the differences that belong to the  $x$ -graph  $C$ , the second sum corresponds to the  $y$ -graph  $D$ , whereas the third sum identifies the differences between the oscillators, belonging to different graphs such that  $i \in C$  and  $j \in D$  or vice versa.

To recalculate the difference variables of the RHS via the the variables  $X_k$  and  $Y_k$ , we should first choose a path from oscillator  $i$  to oscillator  $j$  for any pair of oscillators  $(i, j)$ . We denote it by  $P_{ij}$ . Its path length  $|P_{ij}|$  is the number of edges, comprising the path. The important property of the path  $P_{ij}$  is that if, for example, it passes the oscillators with indices 1, 2, 3, and 4, then the corresponding difference  $X_{14} = x_4 - x_1 = (x_4 - x_3) + (x_3 - x_2) + (x_2 - x_1) = X_{12} + X_{23} + X_{34}$ , where the differences  $X_{12}$ ,  $X_{23}$ , and  $X_{34}$  correspond to the edges and the path length  $|P_{1,4}| = 3$ .

The choice of paths is not unique. We typically choose a shortest path between any pair of  $i$  and  $j$ ; however, a different choice of paths can yield closer estimates, as discussed in one of the examples in Chapter 1. Once the choice of paths is made, we stick with it and start recalculating the difference variables on the RHS of (2.24) via  $X_k$  and  $Y_k$ . A potential problem is that we have to deal not with the variables  $X_{ij}$ , but with their squares  $X_{ij}^2$ , coming from the calculations of the derivative of the Lyapunov function (2.15). Therefore, we have to apply the Cauchy-Schwartz inequality:  $X_{14}^2 = (X_{12} + X_{23} + X_{34})^2 \leq 3(X_{12}^2 + X_{23}^2 + X_{34}^2)$ . Notice the appearance of the factor 3, indicating the number of edges, comprising the path.

Similarly, for any difference  $X_{ij}$  and  $Y_{ij}$  we have

$$\begin{aligned} X_{ij}^2 &= \left( \sum_{k \in P_{ij}} X_k \right)^2 \leq |P_{ij}| \sum_{k \in P_{ij}} X_k^2, \\ Y_{ij}^2 &= \left( \sum_{k \in P_{ij}} Y_k \right)^2 \leq |P_{ij}| \sum_{k \in P_{ij}} Y_k^2, \end{aligned} \quad (2.25)$$

where once again  $|P_{ij}|$  indicates the length of the chosen path from oscillator  $i$  to oscillator  $j$  along the connection graph, combined of the  $x$ - and  $y$ - graphs. At this point, we do not differentiate between paths containing only  $x$  or  $y$  edges, but we have to consider mixed paths when necessary.

Applying this idea to each difference variable on the RHS of (2.24), we obtain the following condition

$$\begin{aligned} n \sum_{k=1}^m c_k X_k^2 + n \sum_{k=1}^l d_k Y_k^2 &> \sum_{k=1}^m [a_x b_k^{x-graph} + \omega_x b_k^{mixed}] X_k^2 + \sum_{k=1}^l [a_y b_k^{y-graph} + \omega_y b_k^{mixed}] Y_k^2 + \\ &\quad \sum_{k \in D} [\omega_x b_k^{mixed}] X_k^2 + \sum_{k \in C} [\omega_y b_k^{mixed}] Y_k^2, \end{aligned} \quad (2.26)$$

where  $b_k^{x-graph} = \sum_{j>i; k \in P_{ij} \in C}^n |P_{ij}|$  is the sum of the lengths of all chosen paths which belong to the  $x$ -graph  $C$  and go through a given  $x$ -edge  $k$ . Similarly,  $b_k^{y-graph} = \sum_{j>i; k \in P_{ij} \in D}^n |P_{ij}|$  is the sum of the lengths of all chosen paths which belong to the  $y$ -graph  $D$  and go through a given  $y$ -edge  $k$ . Finally,  $b_k^{mixed} = \sum_{j>i; k \in P_{ij} \in (C,D)}^n |P_{ij}|$  is the sum of the lengths of all chosen paths which contain  $x$  and  $y$  edges and belong to the mixed  $xy$ -graph and go through a given edge  $k$  which may be an  $x$  or  $y$  edge.

In terms of traffic networks, these three graph theoretical quantities  $b_k^{x-graph}$ ,  $b_k^{y-graph}$  and  $b_k^{mixed}$  represent the total lengths of the chosen roads that go through a given edge  $k$  which can be loosely defined as a busy street. Therefore, we will be referring to them as “traffic” loads.

Note that the two sums on the LHS of (2.26) correspond to the first sums on the RHS. Therefore, if the third and fourth sums on the RHS were absent, we would immediately obtain

the stability conditions as we would drop the summation signs and the difference variables and obtain for  $x$  edges:  $nc_k > a_x b_k^{x-graph} + \omega_x b_k^{mixed}$  and  $y$  edges:  $nd_k > a_y b_k^{y-graph} + \omega_y b_k^{mixed}$ . However, the presence of the third and fourth sums on the RHS makes the argument much more complicated but yields a number of surprising implications of our results to specific networks, discussed in Chapter 3.

A major stability problem, associated with the third and fourth terms, is rooted in the fact that, for example, the third sum  $\sum_{k \in D} [\omega_x b_k^{mixed}] X_k^2$  contains the difference variables  $X_k$  that correspond to the edges of the  $y$  graph. As a result, the first sum  $n \sum_{k=1}^m c_k X_k^2$  which contains the variables  $X_k$  that correspond to  $x$  edges, cannot compensate the third sum on RHS as they belong to different graphs and therefore cannot be compared. At the same time, the second sum  $n \sum_{k=1}^l d_k Y_k^2$  on the LHS does belong to the  $y$ -graph but contains the variables  $Y_k$  and not  $X_k$  needed to handle the third sum. The same problem relates to the fourth sum  $\sum_{k \in C} [\omega_y b_k^{mixed}] Y_k^2$  which contains  $Y_k$  variables, corresponding to the “wrong” graph ( $x$ -graph).

How can we get around this problem as we simply do not have means on the LHS to compensate for the troublesome sums on the RHS? A solution comes from economics: if you do not have means, borrow them! [But act responsibly]. This remark is added to entertain the reader that might be tired of following the proof up to this point.

In fact, the only place to “borrow” these terms from is the auxiliary stability systems (2.10) and (2.11) as they do contain the desired variables  $X_k$  and  $Y_k$ , corresponding to the “right” graphs (the  $x$  and  $y$  graphs, respectively). Therefore, we need to go back and modify the auxiliary systems (2.10) and (2.11) as follows

$$\dot{X}_{ij} = \left[ \int_0^1 DF(\beta x_j + (1 - \beta)x_i) d\beta - [a_x + \alpha_k^x]P + \omega_x b_k^{mixed} L \right] X_{ij} \text{ if } i, j \in x\text{-edge } k, \quad (2.27)$$

$$\dot{X}_{ij} = \left[ \int_0^1 DF(\beta x_j + (1 - \beta)x_i) d\beta - (a_y + \alpha_k^y)L + \omega_x b_k^{mixed} P \right] X_{ij} \text{ if } i, j \in y\text{-edge } k \quad (2.28)$$

Here, for consistency we use the old vector notation  $X_{ij} = (x_j - x_i, y_j - y_i, z_j - z_i)$ . The addition a positive term  $\omega_x b_k^{mixed} L$  to the auxiliary system (2.27) worsens its stability, therefore we have to introduce an additional term  $\alpha_k^x$  and make sure that it is sufficiently large to stabilize the new auxiliary system. A very important property is that, in (2.27), we have to add the positive, destabilizing term  $(\omega_x b_k^{mixed}) L X_{ij}$  to the second equation for the  $(y_j - y_i)$  difference but try to stabilize the system via increasing the additional parameter  $\alpha_k^x$  in the first equation for the  $(x_j - x_i)$  equation (note different inner matrices:  $P$  vs  $L$  in (2.27)). Depending on the individual oscillator, chosen as the individual unit, this might not be possible, especially when traffic load  $b_k^{mixed}$  on the edge  $k$  is high. This property will be discussed in detail for the Lorenz and Chua oscillator examples, given further in Chapter 3. A similar argument carries over to the auxiliary system (2.28) where we add the destabilizing term  $\omega_x b_k^{mixed}$  to first oscillator equation  $(x_j - x_i)$ , but seek to stabilize the system via the additional parameter  $\alpha_k^y$ .

Notice that we only modify the auxiliary systems for  $x$  and  $y$  edges. All the other auxiliary systems for  $X_{ij}$ , that do not correspond to edges of either the  $x$  or  $y$  graphs, remain intact and defined via the original systems (2.10) - (2.12).

Thus, the modifications of (2.27) and (2.28) make the troublesome sums  $\sum_{k \in D} [\omega_x b_k^{mixed}] X_k^2$  and  $\sum_{k \in C} [\omega_y b_k^{mixed}] Y_k^2$  in 2.26) disappear at the expense of worsened stability conditions of the corresponding auxiliary systems, which is reflected by the appearance of additional terms with  $\alpha_k^x$  and  $\alpha_k^y$ . Therefore, 2.26) turns into

$$n \sum_{k=1}^m c_k X_k^2 + n \sum_{k=1}^l d_k Y_k^2 > \sum_{k=1}^m [a_x b_k^{x-graph} + \omega_x b_k^{mixed} + \alpha_k^x] X_k^2 + \sum_{k=1}^l [a_y b_k^{y-graph} + \omega_y b_k^{mixed} + \alpha_k^y] Y_k^2. \quad (2.29)$$

Here, note the new stabilizing constants  $\alpha_k^x$  and  $\alpha_k^y$ . Depending on the individual oscillator dynamics and traffic load on edge  $k$ , these constants might have to be very large or even infinite.

Comparing the terms containing  $X_k$  and  $Y_k$  on the LHS and RHS of (2.29) and omitting the summation signs, we obtain the following conditions

$$\begin{aligned} nc_k X_k^2 &> [a_x b_k^{x-graph} + \omega_x b_k^{mixed} + \alpha_k^x] X_k^2, \quad k = 1, \dots, m \\ nd_k Y_k^2 &> [a_y b_k^{y-graph} + \omega_y b_k^{mixed} + \alpha_k^y] Y_k^2, \quad k = 1, \dots, l. \end{aligned} \quad (2.30)$$

Finally, we omit the difference variables to obtain the bounds on coupling strengths,  $c_k$  for  $x$  edges and  $d_k$  for  $y$  edges, sufficient to make the derivative of the Lyapunov function (2.15) negative definite, and therefore, ensure global stability of synchronization in the network (2.1). It follows from (2.30) that these upper bounds are

$$\begin{aligned} c_k &> \frac{1}{n} [a_x b_k^{x-graph} + \omega_x b_k^{mixed} + \alpha_k^x], \quad k = 1, \dots, m \\ d_k &> \frac{1}{n} [a_y b_k^{y-graph} + \omega_y b_k^{mixed} + \alpha_k^y], \quad k = 1, \dots, l. \end{aligned} \quad (2.31)$$

Thus, we conclude this derivation and formulate the main theoretical result of this thesis.

**Theorem 2.1 [Mixed Connection Graph Method] (Belykh, Carter, and Jeter [56, 57])**

*Synchronization in the network (2.1) is globally stable if for each edge  $k = 1, \dots, m$  ( $l$ )*

$$\begin{aligned} c_{ij} \equiv c_k &> \frac{1}{n} \left\{ a_x \cdot b_k^{x-graph} + \omega_x \cdot b_k^{mixed} + \alpha_k^x \right\} \\ d_{ij} \equiv d_k &> \frac{1}{n} \left\{ a_y \cdot b_k^{y-graph} + \omega_y \cdot b_k^{mixed} + \alpha_k^y \right\}, \end{aligned} \quad (2.32)$$

where  $b^{x-graph}$  ( $b^{y-graph}$ ) is the sum of the lengths of all chosen paths  $P_{ij}$  which pass through a given edge  $k$  that belongs to the  $x$ - ( $y$ -)connection graph. Constant  $a_x$  ( $a_y$ ) is the double coupling strength sufficient for synchronization in the network of two  $x$ -coupled ( $y$ -coupled) oscillators, composing the network. The constants  $\omega_x$  and  $\omega_y$  represent a combination of the double coupling strengths for  $c_{12}$  and  $d_{12}$  sufficient for synchronization in the two-oscillator network with both  $x$  and  $y$  connections. Finally, the constants  $\alpha_x^k$  and  $\alpha_y^k$  are chosen large enough such that they can stabilize the following stability systems written for the difference

variables that correspond to an edge  $k : X_k = X_{ij} = \mathbf{x}_j - \mathbf{x}_i :$

$$\text{For } \alpha_x^k : \dot{X}_k = \left[ \int_0^1 DF(\beta X_i + (1 - \beta) X_j) d\beta \right] X_k + \omega_x b_k^{\text{mixed}} L X_k - (a_x + \alpha_k^x) P X_k, \quad (2.33)$$

$$\text{For } \alpha_y^k : \dot{X}_k = \left[ \int_0^1 DF(\beta X_i + (1 - \beta) X_j) d\beta \right] X_k + \omega_y b_k^{\text{mixed}} P X_k - (a_y + \alpha_k^y) L X_k, \quad (2.34)$$

where the Jacobian  $DF$  can be calculated explicitly via the parameters of the individual oscillator. For clarity, the constants  $a_x$  and  $a_y$  can be chosen equal to represent the double coupling strength sufficient for synchronization in the two-oscillator network with both  $x$  and  $y$  connections. The quantity  $b_k^{\text{mixed}}$  is the sum of the lengths of all chosen paths  $P_{ij}$  between pairs of oscillators  $i$  and  $j$  from different ( $x$  and  $y$ ) connection graphs.

This Mixed Connection Graph method contains an explicit recipe for calculating the upper bounds on the coupling strengths  $c_k$  and  $d_k$  as the constants  $a_x$ ,  $a_y$ ,  $\omega_x$ ,  $\omega_y$ , and  $\alpha_x^k$  and  $\alpha_y^k$  can be explicitly calculated for a network with a given network topology of the two connection graphs. The important quantities such as traffic loads  $b^{x\text{-graph}}$ ,  $b^{y\text{-graph}}$ , and  $b^{\text{mixed}}$  and their use in the stability conditions of Theorem 2.1 clearly reveal the role of network topology in synchronization of a given network. In small networks, traffic loads  $b^{x\text{-graph}}$ ,  $b^{y\text{-graph}}$ , and  $b^{\text{mixed}}$  can be easily calculated by hand, whereas their calculation in large networks require the use of graph combinatorial algorithms. The algorithms will be developed and reported in Chapter 3, when applying the Mixed Connection Graph method to specific networks of Lorenz and Chua oscillators. These applications will reveal highly counterintuitive results which are due to the mixed coupling structure and never observed in networks with one component (not-mixed) graph, discussed in Introduction.

## Chapter 3

### APPLICATION OF THE MIXED GRAPH METHOD

#### 3.1 Four Oscillator Chain Network

To make the proof of the general theorem more clear, especially in regard to where specific components of the stability criteria come from, we will walk the reader through the derivation of the bounds on the coupling threshold for synchronization given in (2.32) for a simple network. Consider four oscillators with the network topology shown in Fig. 3.1. The blue oscillators (oscillators 1 and 2) only belong to the  $x$  connection graph, the yellow oscillator (oscillator 4) only belongs to the  $y$  connection graph, and the white oscillator (oscillator 3) belongs to both the  $x$  and  $y$  connection graphs. We say that oscillator 3 belongs to the “mixed component” of the graph.

Similar to the general equation (2.1), this network represents the following system of differential equations:

$$\begin{aligned}
 \dot{x}_1 &= F(x_1, y_1, z_1) + c_{12}(x_2 - x_1), & \dot{y}_1 &= G(x_1, y_1, z_1), & \dot{z}_1 &= H(\dots) \\
 \dot{x}_2 &= F(x_2, y_2, z_2) + c_{21}(x_1 - x_2) + c_{23}(x_3 - x_2), & \dot{y}_2 &= G(x_2, y_2, z_2), & \dot{z}_2 &= H(\dots) \\
 \dot{x}_3 &= F(x_3, y_3, z_3) + c_{32}(x_2 - x_3), & \dot{y}_3 &= G(x_3, y_3, z_3) + d_{34}(y_4 - y_3), & \dot{z}_3 &= H(\dots) \\
 \dot{x}_4 &= F(x_4, y_4, z_4), & \dot{y}_4 &= G(x_4, y_4, z_4) + d_{43}(y_3 - y_4), & \dot{z}_4 &= H(\dots),
 \end{aligned} \tag{3.1}$$

where functions  $F((x_i, y_i, z_i))$ ,  $G(x_i, y_i, z_i)$ , and  $H(x_i, y_i, z_i)$  define the individual dynamics of the  $i$ -th oscillator.  $c$  and  $d$  are the coupling strengths between  $x$ -coupled and  $y$ -coupled oscillators, respectively.

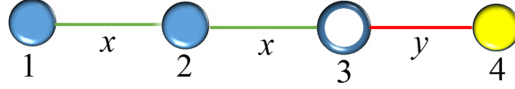


Figure 3.1: Network of four oscillators with a mixed connection graph.

### 3.1.1 Proof of the Stability Condition

Our main objective is to obtain conditions for the global asymptotic stability of synchronization in system (3.1). We seek to determine threshold values for the coupling strength required to synchronize the network. Furthermore, we aim to reveal the dependence of the coupling threshold on the number of oscillators, the coupling configuration, and the individual oscillator dynamics.

We will limit our example to oscillators whose dynamics are governed by the Lorenz equations. Replacing functions  $F$ ,  $G$ , and  $H$  with the corresponding equations for the Lorenz system, we obtain:

$$\begin{aligned}
 \dot{x}_1 &= \sigma(y_1 - x_1) + c_{12}(x_2 - x_1), & \dot{y}_1 &= x_1(r - z_1) - y_1, \\
 \dot{z}_1 &= x_1 y_1 - b z_1, & \dot{x}_2 &= \sigma(y_2 - x_2) + c_{21}(x_1 - x_2) + c_{23}(x_3 - x_2), \\
 \dot{y}_2 &= x_2(r - z_2) - y_2, & \dot{z}_2 &= x_2 y_2 - b z_2, \\
 \dot{x}_3 &= \sigma(y_3 - x_3) + c_{32}(x_2 - x_3), & \dot{y}_3 &= x_3(r - z_3) - y_3 + d_{34}(y_4 - y_3), \\
 \dot{z}_3 &= x_3 y_3 - b z_3, & \dot{x}_4 &= \sigma(y_4 - x_4), \\
 \dot{y}_4 &= x_4(r - z_4) - y_4 + d_{43}(y_3 - y_4), & \dot{z}_4 &= x_4 y_4 - b z_4.
 \end{aligned} \tag{3.2}$$

Synchronization in (3.2) occurs when the differences between the state variables are all zero, for example  $x_1 - x_2 = y_1 - y_2 = z_1 - z_2 = 0$ .

As in the proof of Theorem 2.1, we introduce the difference variables:  $X_{ij} = x_j - x_i$ ,  $Y_{ij} = y_j - y_i$  and  $Z_{ij} = z_j - z_i$ . If all of the difference variables converge to the origin  $X_{ij} = Y_{ij} = Z_{ij} = 0$ , the original dynamical system has converged to the synchronous state.

To study the evolution of the difference variables, we differentiate them with respect to time to obtain:

$$\begin{aligned}
\dot{X}_{12} &= \dot{x}_2 - \dot{x}_1 \\
&= \sigma(y_2 - x_2) + c_{21}(x_1 - x_2) + c_{23}(x_3 - x_2) - (\sigma(y_1 - x_1) + c_{12}(x_2 - x_1)) \\
&= \sigma((y_2 - y_1) - (x_2 - x_1)) - 2c_{12}X_{12} + c_{23}X_{23} \\
\dot{Y}_{12} &= \dot{y}_2 - \dot{y}_1 \\
&= x_2(r - z_2) - y_2 - (x_1(r - z_1) - y_1) \\
&= \left\{ r - \frac{z_1 + z_2}{2} \right\} X_{12} - Y_{12} - \frac{x_1 + x_2}{2} Z_{12} \\
&= \left( r - U_{12}^{(z)} \right) X_{12} - Y_{12} - U_{12}^{(x)} Z_{12} \\
\dot{Z}_{12} &= \dot{z}_2 - \dot{z}_1 \\
&= x_2 y_2 - b z_2 - (x_1 y_1 - b z_1) \\
&= -b Z_{12} + \frac{1}{2} \{ (y_2 - y_1)(x_1 + x_2) + (y_1 + y_2)(x_2 - x_1) \} \\
&= U_{12}^{(y)} X_{12} + U_{12}^{(x)} Y_{12} - b Z_{12}.
\end{aligned} \tag{3.3}$$

Similarly for the remaining differences, we obtain

$$\begin{aligned}
\dot{X}_{13} &= \sigma(Y_{13} - X_{13}) - c_{23}X_{23} - c_{12}X_{12}, & \dot{Y}_{13} &= \left( r - U_{13}^{(z)} \right) X_{13} - Y_{13} - U_{13}^{(x)} Z_{13} + d_{34}Y_{34}, \\
\dot{Z}_{13} &= U_{13}^{(y)} X_{13} + U_{13}^{(x)} Y_{13} - b Z_{13}, \\
\dot{X}_{14} &= \sigma(Y_{14} - X_{14}) - c_{12}X_{12}, & \dot{Y}_{14} &= \left( r - U_{14}^{(z)} \right) X_{14} - Y_{14} - U_{14}^{(x)} Z_{14} - d_{34}Y_{34}, \\
\dot{Z}_{14} &= U_{14}^{(y)} X_{14} + U_{14}^{(x)} Y_{14} - b Z_{14}, \\
\dot{X}_{23} &= \sigma(Y_{23} - X_{23}) - 2c_{23}X_{23} + c_{12}X_{12}, & \dot{Y}_{23} &= \left( r - U_{23}^{(z)} \right) X_{23} - Y_{23} - U_{23}^{(x)} Z_{23} + d_{34}Y_{34}, \\
\dot{Z}_{23} &= U_{23}^{(y)} X_{23} + U_{23}^{(x)} Y_{23} - b Z_{23}, \\
\dot{X}_{24} &= \sigma(Y_{24} - X_{24}) + c_{12}X_{12} - c_{23}X_{23}, & \dot{Y}_{24} &= \left( r - U_{24}^{(z)} \right) X_{24} - Y_{24} - U_{24}^{(x)} Z_{24} - d_{34}Y_{34}, \\
\dot{Z}_{24} &= U_{24}^{(y)} X_{24} + U_{24}^{(x)} Y_{24} - b Z_{24} \\
\dot{X}_{34} &= \sigma(Y_{34} - X_{34}) + c_{23}X_{23}, & \dot{Y}_{34} &= \left( r - U_{34}^{(z)} \right) X_{34} - Y_{34} - U_{34}^{(x)} Z_{34} - 2d_{34}Y_{34}, \\
\dot{Z}_{34} &= U_{34}^{(y)} X_{34} + U_{34}^{(x)} Y_{34} - b Z_{34},
\end{aligned} \tag{3.4}$$

where  $U_{ij}^{(\xi)} = (\xi_i + \xi_j)/2$  for  $\xi = x, y, z$  are the corresponding sum variables.

Now, proving the global stability of synchronization in (3.2) amounts to proving the global, asymptotic stability of the origin for the difference system ((3.3) and (3.4)). As in the general proof of Theorem 2.1, we will use the Lyapunov function method of proving the global stability of a fixed point. We remind the reader that a Lyapunov function,  $V(\mathbf{x})$  is a positive semi-definite function (with  $V = 0$  only at the fixed point,  $\mathbf{x}^*$ ) whose derivative, with respect to the original system, is negative semi-definite (with  $\dot{V} = 0$  only at the fixed point,  $\mathbf{x}^*$ ). We construct the following Lyapunov function:

$$V = \frac{1}{2}(X_{12}^2 + Y_{12}^2 + Z_{12}^2 + X_{13}^2 + Y_{13}^2 + Z_{13}^2 + X_{14}^2 + Y_{14}^2 + Z_{14}^2 + X_{23}^2 + Y_{23}^2 + Z_{23}^2 + X_{24}^2 + Y_{24}^2 + Z_{24}^2 + X_{34}^2 + Y_{34}^2 + Z_{34}^2). \quad (3.5)$$

Our goal is to obtain conditions under which solutions of the coupled system (3.4) converge to 0 as  $t \rightarrow \infty$ , which implies that the synchronization manifold of system (3.1) is globally asymptotically stable. This is done by proving that the function  $V$  is positive semi-definite, and its derivative with respect to the system (3.1) is negative semi-definite. However, as the function  $V$  is positive semi-definite by construction (with  $V = 0$  only along the synchronization manifold  $X_{ij} = Y_{ij} = Z_{ij} = 0$ ) we only need to prove that its derivative  $\dot{V}$  is negative semi-definite. The corresponding time derivative has the form

$$\dot{V} = X_{12}\dot{X}_{12} + Y_{12}\dot{Y}_{12} + Z_{12}\dot{Z}_{12} + X_{13}\dot{X}_{13} + Y_{13}\dot{Y}_{13} + Z_{13}\dot{Z}_{13} + X_{14}\dot{X}_{14} + Y_{14}\dot{Y}_{14} + Z_{14}\dot{Z}_{14} + X_{23}\dot{X}_{23} + Y_{23}\dot{Y}_{23} + Z_{23}\dot{Z}_{23} + X_{24}\dot{X}_{24} + Y_{24}\dot{Y}_{24} + Z_{24}\dot{Z}_{24} + X_{34}\dot{X}_{34} + Y_{34}\dot{Y}_{34} + Z_{34}\dot{Z}_{34},$$

where the derivatives of the difference variables are given by (3.4). Substituting the derivatives of the difference variables into the equation above, we are able to simplify the derivative of the Lyapunov function  $V$ .

$\dot{V}$

$$\begin{aligned}
&= X_{12}\dot{X}_{12} + Y_{12}\dot{Y}_{12} + X_{13}\dot{X}_{13} + Y_{13}\dot{Y}_{13} + X_{14}\dot{X}_{14} + Y_{14}\dot{Y}_{14} + X_{23}\dot{X}_{23} + Y_{23}\dot{Y}_{23} + \\
&X_{24}\dot{X}_{24} + Y_{24}\dot{Y}_{24} + X_{34}\dot{X}_{34} + Y_{34}\dot{Y}_{34} \\
&= X_{12}\left(\sigma(Y_{12} - X_{12}) - 2c_{12}X_{12} + c_{23}X_{23}\right) + Y_{12}\left(\left(r - U_{12}^{(z)}\right)X_{12} - Y_{12} - U_{12}^{(x)}Z_{12}\right) + \\
&+ Z_{12}\left(U_{12}^{(y)}X_{12} + U_{12}^{(x)}Y_{12} - bZ_{12}\right) + X_{13}\left(\sigma(Y_{13} - X_{13}) - c_{23}X_{23} - c_{12}X_{12}\right) + \\
&+ Y_{13}\left(\left(r - U_{13}^{(z)}\right) \times X_{13} - Y_{13} - U_{13}^{(x)}Z_{13} + d_{34}Y_{34}\right) + Z_{13}\left(U_{13}^{(y)}X_{13} + U_{13}^{(x)}Y_{13} - bZ_{13}\right) + \\
&+ X_{14}\left(\sigma(Y_{14} - X_{14}) - c_{12}X_{12}\right) + Y_{14}\left(\left(r - U_{14}^{(z)}\right)X_{14} - Y_{14} - U_{14}^{(x)}Z_{14} - d_{34}Y_{34}\right) + \\
&+ Z_{14}\left(U_{14}^{(y)}X_{14} + U_{14}^{(x)}Y_{14} - bZ_{14}\right) + X_{23}\left(\sigma(Y_{23} - X_{23}) - 2c_{23}X_{23} + c_{12}X_{12}\right) + \\
&+ Y_{23}\left(\left(r - U_{23}^{(z)}\right)X_{23} - Y_{23} - U_{23}^{(x)}Z_{23} + d_{34}Y_{34}\right) + Z_{23}\left(U_{23}^{(y)}X_{23} + U_{23}^{(x)}Y_{23} - bZ_{23}\right) + \\
&+ X_{24}\left(\sigma(Y_{24} - X_{24}) + c_{12}X_{12} - c_{23}X_{23}\right) + Y_{24}\left(\left(r - U_{24}^{(z)}\right)X_{24} - Y_{24} - U_{24}^{(x)}Z_{24} - d_{34}Y_{34}\right) + \\
&+ Z_{24}\left(U_{24}^{(y)}X_{24} + U_{24}^{(x)}Y_{24} - bZ_{24}\right) + X_{34}\left(\sigma(Y_{34} - X_{34}) + c_{23}X_{23}\right) + Y_{34}\left(\left(r - U_{34}^{(z)}\right)X_{34} - \right. \\
&- Y_{34} - U_{34}^{(x)}Z_{34} - 2d_{34}Y_{34}\left.)\right) + Z_{34}\left(U_{34}^{(y)}X_{34} + U_{34}^{(x)}Y_{34} - bZ_{34}\right) \\
&= \sigma X_{12}Y_{12} - \sigma X_{12}^2 + \left\{r - U_{12}^{(z)}\right\} X_{12}Y_{12} - Y_{12}^2 + U_{12}^{(y)}X_{12}Z_{12} - bZ_{12}^2 + \sigma X_{13}Y_{13} - \sigma X_{13}^2 + \\
&+ \left\{r - U_{13}^{(z)}\right\} X_{13}Y_{13} - Y_{13}^2 + U_{13}^{(y)}X_{13}Z_{13} - bZ_{13}^2 + \sigma X_{14}Y_{14} - \sigma X_{14}^2 + \left\{r - U_{14}^{(z)}\right\} X_{14}Y_{14} - \\
&- Y_{14}^2 + U_{14}^{(y)}X_{14}Z_{14} - bZ_{14}^2 + \sigma X_{23}Y_{23} - \sigma X_{23}^2 + \left\{r - U_{23}^{(z)}\right\} X_{23}Y_{23} - Y_{23}^2 + U_{23}^{(y)}X_{23}Z_{23} - \\
&- bZ_{23}^2 + \sigma X_{24}Y_{24} - \sigma X_{24}^2 + \left\{r - U_{24}^{(z)}\right\} X_{24}Y_{24} - Y_{24}^2 + U_{24}^{(y)}X_{24}Z_{24} - bZ_{24}^2 + \sigma X_{34}Y_{34} - \\
&- \sigma X_{34}^2 + \left\{r - U_{34}^{(z)}\right\} X_{34}Y_{34} - Y_{34}^2 + U_{34}^{(y)}X_{34}Z_{34} - bZ_{34}^2 - 4c_{12}X_{12}^2 - 4c_{23}X_{23}^2 - 4d_{34}Y_{34}^2
\end{aligned}$$

The terms in the derivative of the Lyapunov function that depend explicitly on coupling are:  $-4c_{12}X_{12}^2 - 4c_{23}X_{23}^2 - 4d_{34}Y_{34}^2$ . Observe that the coefficient for each coupling term is the number of oscillators. This will always be the general structure for the coupling terms after simplifying  $\dot{V}$ . We refer to the remaining terms as the “intrinsic terms” or the “uncoupled system”. The coupling terms play a critical role in whether or not  $\dot{V}$  is negative semi-definite. The coupling parameters are free variables that can be increased indefinitely, and the coupling terms are necessarily negative, hence they can nullify positive terms that come

from the individual oscillator dynamics. Therefore, it is using these coupling terms that we are able to prove the stability of the coupled system.

To obtain global synchronization, we compose a new Lyapunov function to help compensate the uncoupled terms as follows:

$$\dot{V} = a_x X_{12}^2 + a_x X_{13}^2 + a_x X_{23}^2 + a_y Y_{34}^2 + \omega_x X_{14}^2 + \omega_y Y_{14}^2 + \omega_x X_{24}^2 + \omega_y Y_{24}^2 - 4c_{12} X_{12}^2 - 4c_{23} X_{23}^2 - 4d_{34} Y_{34}^2.$$

We use the coefficients,  $a_x$  and  $a_y$  for the  $x$ - and  $y$ -edges, respectively, for differences along edges that are present in the given mixed network. The coefficients,  $\omega_x$  and  $\omega_y$ , are used for the edges which do not have an edge in the mixed network.

To stabilize the network through  $x$ ,  $y$  and the mixed component, we have the following auxiliary systems:

$$\begin{aligned}\dot{X}_{ij} &= \sigma(Y_{ij} - X_{ij}) - a_x X_{ij} = \dots - a_x X_{ij}, \\ \dot{Y}_{ij} &= \left( r - U_{ij}^{(z)} \right) X_{ij} - Y_{ij} - U_{ij}^{(x)} Z_{ij} = \dots, \\ \dot{Z}_{ij} &= U_{ij}^{(y)} X_{ij} + U_{ij}^{(x)} Y_{ij} - b Z_{ij} = \dots,\end{aligned}$$

This auxiliary system corresponds to the stability of the differences within the  $x$ -connection graph, which are  $X_{12}, X_{23}, X_{13}$ . Here, for convenience, we have replaced the intrinsic terms, defined by the contribution of the uncoupled systems, by the dots. In terms of Theorem 2.1, these dots correspond to the vector form:  $\left[ \int_0^1 DF(\beta x_j + (1 - \beta)x_i) d\beta \right] \mathbf{X}_{ij}$ , where vector  $\mathbf{X}_{ij} = (X_{ij}, Y_{ij}, Z_{ij})$ . We will be using this compact dot notation throughout the proof.

Similarly, we write the auxiliary system for the stability of the differences within the  $y$ -connection graph, which is just  $Y_{34}$ :

$$\begin{aligned}\dot{X}_{ij} &= \dots, \\ \dot{Y}_{ij} &= \dots - a_y Y_{ij}, \\ \dot{Z}_{ij} &= \dots\end{aligned}$$

Finally, the auxiliary system for the stability of the differences within the mixed connection graph, which are  $X_{14}, Y_{14}, X_{24}, Y_{24}$ , is

$$\begin{aligned}\dot{X}_{ij} &= \dots - \omega_x X_{ij}, \\ \dot{Y}_{ij} &= \dots - \omega_y Y_{ij}, \\ \dot{Z}_{ij} &= \dots\end{aligned}$$

While the coupling terms can compensate for the difference variables along edges that exist in the network (as those are the only difference variables with corresponding coupling terms), the difference variables without edges in the network present an obstacle for synchronization. There are no coupling terms to stabilize these terms!

To overcome this obstacle, notice that the difference variables that do not have edges in the network can be re-written as a linear combination of difference variables with edges on the network. For example, we have  $X_{13} = x_3 - x_1 = (x_3 - x_2) + (x_2 - x_1) = X_{23} + X_{12}$ . Next, we use the alternative expressions for the difference variables and the Cauchy-Schwarz inequality to obtain the following bounds on the squares of the difference variables:

$$\begin{aligned}X_{13}^2 &= (X_{12} + X_{23})^2 \leq 2(X_{12}^2 + X_{23}^2) \\ X_{14}^2 &= (X_{12} + X_{23} + X_{34})^2 \leq 3(X_{12}^2 + X_{23}^2 + X_{34}^2) \\ Y_{14}^2 &= (Y_{12} + Y_{23} + Y_{34})^2 \leq 3(Y_{12}^2 + Y_{23}^2 + Y_{34}^2) \\ X_{24}^2 &= (X_{23} + X_{34})^2 \leq 2(X_{23}^2 + X_{34}^2) \\ Y_{24}^2 &= (Y_{23} + Y_{34})^2 \leq 2(Y_{23}^2 + Y_{34}^2).\end{aligned}$$

After making the respective substitutions into the derivative of the Lyapunov function, we obtain:

$$\begin{aligned}\dot{V} \leq & a_x X_{12}^2 + a_x 2(X_{12}^2 + X_{23}^2) + a_x X_{23}^2 + a_y Y_{34}^2 + \omega_x 3((X_{12}^2 + X_{23}^2 + X_{34}^2)) + \\ & \omega_y 3((Y_{12}^2 + Y_{23}^2 + Y_{34}^2)) + \omega_x 2((X_{23}^2 + X_{34}^2)) + \omega_y 2((Y_{23}^2 + Y_{34}^2)) - \\ & 4c_{12} X_{12}^2 - 4c_{23} X_{23}^2 - 4d_{34} Y_{34}^2.\end{aligned}$$

Grouping the terms yields

$$\begin{aligned} \dot{V} \leq & [a_x(1+2) + \omega_x 3]X_{12}^2 + [a_x(1+2) + \omega_x(3+2)]X_{23}^2 + (a_y + \omega_y[2+3])Y_{34}^2 - \\ & 4c_{12}X_{12}^2 - 4c_{23}X_{23}^2 - 4d_{34}Y_{34}^2 + \omega_x[3+2]X_{34}^2 + \omega_y 3Y_{12}^2 + \omega_y[2+3]Y_{23}^2. \end{aligned} \quad (3.6)$$

After grouping the terms according to their corresponding difference variables, the reader should take note of the coefficients for the difference variables in each term. For example,  $[a_x(1+2) + \omega_x]X_{12}^2$ . The known integers in these coefficients come from the use of the Cauchy-Schwarz inequality, and represent the sum of the path lengths of the paths passing through the given edge. For edge 1-2, the  $(1+2)$  in the term  $a_x(1+2)X_{12}^2$  comes from the paths in the  $x$  connection graph passing through the edge 1-2. This would be the path from oscillator 1 to oscillator 2 (through edge 1-2) and the path from oscillator 1 to oscillator 3 (through edge 1-2 and edge 2-3). This gives the dependence of our stability condition on the sum of the path lengths of the paths passing through a given edge in the network that we see in Theorem 2.1.

In the inequality given in (3.6), the ‘‘good terms’’, which are  $X_{12}^2$ ,  $X_{23}^2$  and  $Y_{34}^2$ , can be pairwise compensated for by the coupling terms. However, the differences that correspond to the opposite variable as the type of edge (for example,  $X_{34}$ , where the edge between oscillator 3-4 is through the  $y$  variable) cannot be stabilized through the coupling term. Therefore, to compensate for the term  $+\omega_y 3Y_{12}^2$ , we need an additional stabilizing term  $-\alpha Y_{12}^2$  in the  $X_{12}$  auxiliary system, where  $\alpha > 3\omega_y$ .

To add the additional stabilizing term  $-\alpha Y_{12}^2$ , we must change the differences systems. The modified auxiliary system for  $X_{12}$  becomes:

$$\begin{aligned} \dot{X}_{12} &= \dots - a_x X_{12} - \alpha_{12}^x X_{12} \\ \dot{Y}_{12} &= \dots + 3\omega_y Y_{12} \\ \dot{Z}_{12} &= \dots \end{aligned}$$

The changes are similar for all of the  $X_{ij}$  auxiliary systems, with the addition of the  $-\alpha_{ij}^x X_{ij}$  and  $+\omega_y Y_{ij}$  terms.

It is important for the reader to notice that these changes to the  $X_{ij}$  auxiliary systems worsen the stability criteria for synchronization through the  $x$  connection graph (caused by adding the  $\omega_y Y_{12}$  term to the  $X_{ij}$  auxiliary system). While we let  $\alpha_{ij}$  be large enough to stabilize the  $X_{ij}$  auxiliary system, it requires that such an  $\alpha_{ij}$  exists.

Applying these changes for  $X_{12}$  adds two new terms to the Lyapunov function, which now has a derivative bounded by:

$$\begin{aligned} \dot{V} \leq & [a_x(1+2) + \omega_x 3 + \alpha] X_{12}^2 + [a_x(1+2) + \omega_x(3+2)] X_{23}^2 + (a_y + \omega_y[2+3]) Y_{34}^2 - \\ & 4c X_{12}^2 - 4c X_{23}^2 - 4d Y_{34}^2 + \omega_x[3+2] X_{34}^2 + \omega_y 3 Y_{12}^2 - \omega_y 3 Y_{12}^2 + \omega_y[2+3] Y_{23}^2. \end{aligned}$$

We apply the same reasoning and changes to the  $X_{23}$  and  $Y_{34}$  auxiliary systems to compensate for the terms  $+\omega_x[3+2] X_{34}^2$  and  $+\omega_y[3+2] Y_{23}^2$ . The modified auxiliary systems required to stabilize the differences across each respective edge are given by:

$$\begin{aligned} \dot{X}_{12} &= \dots - a_x X_{12} - \alpha_{12}^x X_{12} \\ \dot{Y}_{12} &= \dots + 3\omega_y Y_{12} \\ \dot{Z}_{12} &= \dots \\ \dot{X}_{34} &= \dots + \omega_x[2+3] X_{34} \\ \dot{Y}_{34} &= \dots - a_y Y_{34} - \alpha_{34}^y Y_{34} \\ \dot{Z}_{34} &= \dots \\ \dot{X}_{23} &= \dots - a_x X_{23} - \alpha_{23}^x X_{23} \\ \dot{Y}_{23} &= \dots + \omega_y[2+3] Y_{23} \\ \dot{Z}_{23} &= \dots \end{aligned} \tag{3.7}$$

Finally, we arrive at:

$$\begin{aligned} \dot{V} \leq & [a_x(1+2) + \omega_x 3 + \alpha_{12}^x] X_{12}^2 - 4c_{12} X_{12}^2 + [a_x(1+2) + \omega_x(3+2) \alpha_{23}^x] X_{23}^2 - 4c_{23} X_{23}^2 + \\ & (a_y + \omega_y[2+3] + \alpha_{34}^y) Y_{34}^2 - 4d_{34} Y_{34}^2. \end{aligned}$$

This derivative,  $\dot{V}$  is negative semi-definite if  $c_{12} > \frac{1}{4}[a_x 3 + \omega_x 3 + \alpha_{12}^x]$ ,  $c_{23} > \frac{1}{4}[a_x 3 + \omega_x 5 + \alpha_{23}^x]$ , and  $d_{34} > \frac{1}{4}[a_y + \omega_y 5 + \alpha_{34}^y]$ . If this derivative is negative semi-definite, the origin is globally,

asymptotically stable for the difference system ((3.3) and (3.4)). The origin of the difference system is given by  $x_i - x_j = y_i - y_j = z_i - z_j = 0$  for the original system (3.2), which is the synchronous state.

This gives the following stability condition for the original system:

$$\text{edge 1-2: } c_{12} > \frac{1}{4}[a_x 3 + \omega_x 3 + \alpha_{12}^x]$$

$$\text{edge 2-3: } c_{23} > \frac{1}{4}[a_x 3 + \omega_x 5 + \alpha_{23}^x]$$

$$\text{edge 3-4: } d_{34} > \frac{1}{4}[a_y + \omega_y 5 + \alpha_{34}^y].$$

### 3.1.2 Computing the Stability Conditions

Here, we discuss stability criteria to clearly walk the reader through the computation of the bounds proven in the previous section.

Synchronization for the four-node network (see Fig. 3.1) is globally stable if for each edge  $k = 1, 2, 3$ :

$$\begin{aligned} c_{ij} \equiv c_k &> \frac{1}{n} \left\{ a_x \cdot b_k^{x\text{-graph}} + \omega_x \cdot b_k^{\text{mixed}} + \alpha_k^x \right\} \\ d_{ij} \equiv d_k &> \frac{1}{n} \left\{ a_y \cdot b_k^{y\text{-graph}} + \omega_y \cdot b_k^{\text{mixed}} + \alpha_k^y \right\}, \end{aligned} \quad (3.8)$$

Computing the bounds in (3.8), is a complicated, multi-step process of balancing the  $x$  and  $y$  coupling terms. We walk the reader through these steps systematically, describing (in detail) how each piece of every term is calculated and where in the previous proof these pieces come from.

$\frac{1}{n}$  is 1 over the number of oscillators in the network. In the proof, this comes from the coefficients of the coupling terms in the original derivative of the Lyapunov function. In this example,  $n = 4$ , because there are four oscillators in the network.

$b_k^{\text{mixed}}$  is the sum of path lengths of all paths passing through edge  $k$  that include both  $x$  and  $y$  edges. In the proof, this term comes from re-writing the difference variables for differences that do not correspond to edges in the network, and then applying the Cauchy-Schwarz inequality to place a bound on their squares. To compute all of the paths that pass through a given edge, it is recommended that the reader algorithmically finds the shortest

path between every pair of oscillators, and take note of the paths that go through edge  $k$ . A description of the computation of shortest paths is given in Appendix B.

Using the network of Fig. 3.1, we compute  $b_{23}^{mixed}$  as an example. There are two paths in the network that pass through edges of both types ( $x$  and  $y$ ) and edge 2-3. These paths are the path from oscillator 1 to oscillator 4,  $P_{14}$  and the path from oscillator 2 to oscillator 4,  $P_{24}$ . The lengths of these paths are  $|P_{14}| = 3$  and  $|P_{24}| = 2$ . Hence, for this example,  $b_{23}^{mixed} = |P_{14}| + |P_{24}| = 3 + 2 = 5$ . For all of the edges, we have:

$$\begin{aligned} b_{12}^{mixed} &= |P_{14}| = 3, \\ b_{23}^{mixed} &= |P_{14}| + |P_{24}| = 3 + 2 = 5, \\ b_{34}^{mixed} &= |P_{24}| = 2. \end{aligned}$$

$b_k^{x-graph}$  and  $b_k^{y-graph}$  are the sums of the path lengths of all paths passing through only  $x$  and  $y$  edges, respectively. Similar to  $b_k^{mixed}$ , this term comes from re-writing the difference variables for differences that do not correspond to edges in the network, and using the Cauchy-Schwarz inequality.

Using the network of Fig. 3.1, we compute  $b_{23}^{x-graph}$  as an example. There are two paths in the network that pass through edge 2-3, and are only part of the  $x$  connection graph. These paths are the path from oscillator 1 to oscillator 3,  $P_{13}$ , and the path from oscillator 2 to oscillator 3,  $P_{23}$ . The lengths of these paths are  $|P_{13}| = 2$  and  $|P_{23}| = 1$ . Hence, for this example,  $b_{23}^x = |P_{13}| + |P_{23}| = 2 + 1 = 3$ . For all of the edges, we have:

$$\begin{aligned} b_{12}^x &= |P_{12}| + |P_{13}| = 1 + 2 = 3, \\ b_{23}^x &= |P_{13}| + |P_{23}| = 2 + 1 = 3, \\ b_{34}^y &= |P_{34}| = 1 \end{aligned}$$

*Remark:* As discussed earlier, the shortest path between two pairs of oscillators is not always the optimal choice of paths when computing  $b_k^{mixed}$ ,  $b_k^{x-graph}$  and  $b_k^{y-graph}$ . Occasionally, many of the shortest paths will use the same ‘‘shortcut’’ edges. This can cause  $b_k^{mixed}$ ,  $b_k^{x-graph}$  or

$b_k^{y-graph}$  to become very large, making the bounds on  $c_{ij}$  and  $d_{ij}$  deceptively large for these shortcut edges.

So far, the stability condition is given by:

$$\begin{aligned} c_{12} &> \frac{1}{4}[a_x 3 + \omega_x 3 + \alpha_{12}^x] \\ c_{23} &> \frac{1}{4}[a_x 3 + \omega_x 5 + \alpha_{23}^x] \\ d_{34} &> \frac{1}{4}[a_y + \omega_y 5 + \alpha_{34}^y] \end{aligned}$$

$a_x$  and  $a_y$  are double the coupling strength necessary to synchronize a network of two oscillators coupled through  $x$  or  $y$ , respectively. These values come from the auxiliary systems that need to be stabilized for a network with only a  $x$  or  $y$  connection graph. If we were to consider a network through only one variable, the modified auxiliary systems in (3.7) would only have the terms from the individual oscillator dynamics (represented by the  $\dots$ ) and the  $a_x$  and  $a_y$  terms. More casually,  $a_x$  and  $a_y$  can be thought of as the means of stabilizing the network through the respective, separate connection graphs.

For the Lorenz system,  $a_x \approx 7.5$  and  $a_y \approx 2.8$ .

$\omega_x$  and  $\omega_y$  are chosen such that they are large enough to stabilize the mixed auxiliary system (corresponding to the mixed connection graph), which is given by:

$$\begin{aligned} \dot{X}_k &= \sigma(Y_k - X_k) - \omega_x X_k, \\ \dot{Y}_k &= \left(r - U_k^{(z)}\right) X_k - Y_k - U_k^{(x)} Z_k - \omega_y Y_k \\ \dot{Z}_k &= U_k^{(y)} X_k + U_k^{(x)} Y_k - b, \end{aligned} \tag{3.9}$$

The effect of  $\omega_x$  and  $\omega_y$  in stabilizing this difference is shown in Fig. 3.2. These terms are necessary in showing that differences along the mixed connection graph are stable. The critical step in the proof requires showing that these terms in other auxiliary systems can be compensated for, but it is necessary that they be large enough to stabilize the auxiliary system shown in (3.9).

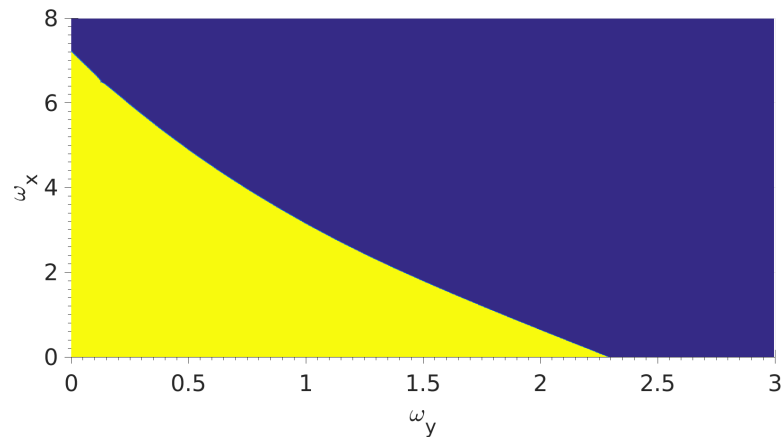


Figure 3.2: Stability diagram for the mixed auxiliary system given in (3.9). The blue (dark) region gives values of  $\omega_x$  and  $\omega_y$  that stabilize the auxiliary system, while the yellow (light) region gives those values that fail to stabilize the mixed auxiliary system. It is using this diagram that  $\omega_x$  and  $\omega_y$  are computed in the stability criteria (2.32).

As  $\omega_x$  and  $\omega_y$  are chosen such that the mixed auxiliary system is stable, we can choose any pair of values in the blue region for Fig. 3.2. For the purpose of illustration, let  $\omega_x = 4$  and  $\omega_y = 1$ .

Substituting the values of  $a_x$ ,  $a_y$ ,  $\omega_x$ , and  $\omega_y$ , we obtain:

$$\begin{aligned} c_{12} &> \frac{1}{4}[22.5 + 12 + \alpha_{12}^x] \\ c_{23} &> \frac{1}{4}[22.5 + 20 + \alpha_{23}^x] \\ d_{34} &> \frac{1}{4}[2.8 + 5 + \alpha_{34}^y] \end{aligned}$$

In the proof of the stability condition for the four oscillator tree, we saw that the stability of synchronization in the network depends on the stability of the modified auxiliary systems in (3.7). These modified auxiliary systems have both stabilizing terms (the additional negative terms for the difference variable that corresponds to the type of edge between the two oscillators) and destabilizing terms (the positive terms for the differences variable that corresponds to the alternative type of edge in the network). Because of these modified auxiliary systems, increasing coupling for one type of edge will simultaneously improve the stability of one auxiliary system and worsen stability of another difference system.

For example, increasing the coupling  $d_{34}$  along the  $y$ -edge in network (3.2) will improve the stability of the auxiliary system for edge 3-4 in (3.7) by making the  $\dot{Y}_{34}$  more negative (through  $\alpha_{34}^y$ ). At the same time, the modified auxiliary systems for the 1-2 and 2-3 edges become more unstable, because the  $\dot{X}_{12}$  and  $\dot{X}_{23}$  equations will become more positive (through  $3\omega_y$ ).

The auxiliary systems that must be stabilized are given by:

$$\begin{aligned}\dot{X}_k &= \sigma(Y_k - X_k) - (a_x + \alpha_k^x)X_k, \\ \dot{Y}_k &= \left(r - U_k^{(z)}\right)X_k - Y_k - U_k^{(x)}Z_k + \omega_x b_k^{\text{mixed}}Y_k \\ \dot{Z}_k &= U_k^{(y)}X_k + U_k^{(x)}Y_k - b,\end{aligned}\tag{3.10}$$

$$\begin{aligned}\dot{X}_k &= \sigma(Y_k - X_k) + \omega_y b_k^{\text{mixed}}X_k \\ \dot{Y}_k &= \left(r - U_k^{(z)}\right)X_k - Y_k - U_k^{(x)}Z_k - (a_y + \alpha_k^y)Y_k, \\ \dot{Z}_k &= U_k^{(y)}X_k + U_k^{(x)}Y_k - bZ_{ij},\end{aligned}\tag{3.11}$$

where (3.10) corresponds to the differences that are part of the  $x$  connection graph, and (3.11) corresponds to the differences that are part of the  $y$  connection graph.

As  $a_x$  and  $a_y$  are taken to be large enough to stabilize their respective auxiliary systems, without the additional terms, we can ignore the ‘‘intrinsic terms’’ when computing  $\alpha_k^x$  and  $\alpha_k^y$ . Therefore,  $\alpha_k^x$  [ $\alpha_k^y$ ] must be large enough to stabilize (3.10) [(3.11)] in the presence of  $\omega_x b_k^{\text{mixed}}$  [ $\omega_y b_k^{\text{mixed}}$ ]. The complex relationship between these terms in regard to stabilizing (3.10) [(3.11)] is shown in the left [right] figure in Fig. 3.3. Notice the coefficient  $\beta_x$  [ $\beta_y$ ] on the  $\omega_x b_k^{\text{mixed}}\beta_x$  [ $\omega_y b_k^{\text{mixed}}\beta_y$ ] axis. Because of the Cauchy-Schwarz inequality,  $b_k^{\text{mixed}}$  provides an over estimate for the terms added into the auxiliary system. To compensate for this overestimate, we introduce the scaling factors  $\beta_x$  and  $\beta_y$  to the stability of (3.10) and (3.11).

For our example, we will take  $\beta_x = \beta_y = 0.1$ . For edge 1-2, this gives  $\omega_x b_k^{\text{mixed}}\beta_x = 4 \cdot 3 \cdot 0.1 = 1.2$ . Using the left plot in Fig. 3.3, we obtain  $\alpha_x \approx 15$ .

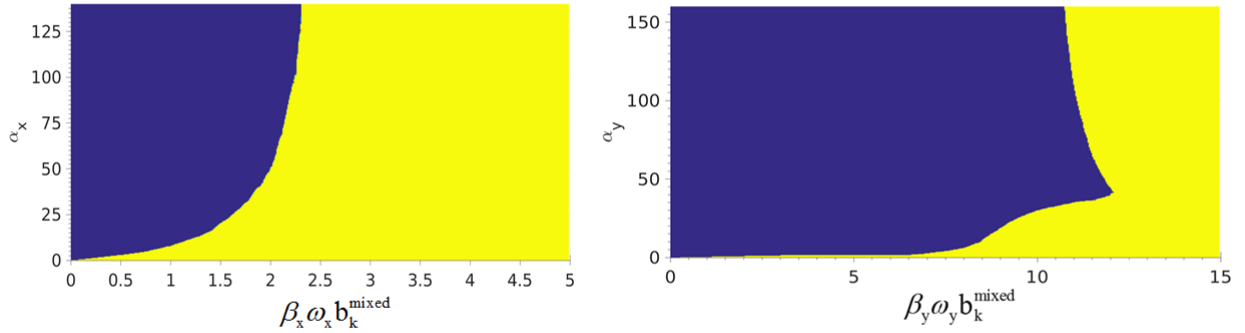


Figure 3.3: Stabilization of the  $X$  (left) and  $Y$  (right) auxiliary systems.

$$c_{12} > \frac{1}{4}[22.5 + 12 + 15] = 12.375,$$

$$c_{23} > \frac{1}{4}[22.5 + 20 + 15] = 14.375.$$

Similarly we can obtain a theoretical bound on the edge coupled through  $y$  (edge 3-4) using the right plot in Fig.3.3 and selecting  $\beta_y = 0.1$ . This gives  $\omega_y b_k^{mixed} \beta_y = 1 \cdot 1 \cdot 0.1 = 0.1$ . Using the stability diagram, we obtain  $\alpha_y \approx 5$ . And the theoretical bound would be

$$d_{34} > \frac{1}{4}[2.8 + 5 + 5] = 3.2.$$

We have the follow bounds on the coupling strengths necessary to synchronize the network:

$$\begin{aligned} c_{12} &> 12.375 \\ c_{23} &> 14.375 \\ d_{34} &> 3.2. \end{aligned} \tag{3.12}$$

Furthermore, if we are considering a uniform coupling strength,  $\varepsilon$ , we let  $\varepsilon = \max\{c_{12}, c_{23}, d_{34}\}$ , which gives a uniform coupling strength of  $\varepsilon = 14.375$  for the network. We compare this theoretically computed value of  $\varepsilon$  to the numerically computed value of  $\varepsilon$

necessary to synchronize the network,  $\varepsilon = 13.38$ . Notice that the theoretically computed bound is still conservative relative to the numerically computed bound.

With the derivation and application of the stability criteria (3.8) more clear, we turn our attention to the synchronization properties of more complex mixed networks.

### 3.2 Other Networks

In this section, we will use various mixed complex networks to reveal some of the properties of mixed networks, and uncover surprising insights into the mechanisms underlying the behavior of mixed networks. Inspired by the theory we have presented, we conjecture that the synchronizability of a mixed network depends on the stability of the differences across the most highly loaded edge (i.e. the edge with the most paths passing through it). Let's now consider another mixed network consisting of three coupled Lorenz oscillators as shown below.

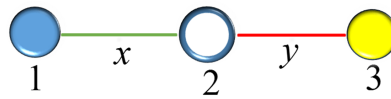


Figure 3.4: Example of three oscillators coupled through  $x$  and  $y$ .

Using our general Theorem (Theorem 2.1), we arrive at the following bound for each respective edge.

1.  $x$ -edge :  $c_{12} > \frac{1}{3}(1 \cdot a_x + 2 \cdot \omega_x + \alpha_{12}^x)$
2.  $y$ -edge :  $d_{12} > \frac{1}{3}(1 \cdot a_y + 2 \cdot \omega_y + \alpha_{12}^y)$ .

The auxiliary system to stabilize edge 2-3 is

$$\begin{aligned}\dot{X}_{23} &= \sigma Y_{23} - \sigma X_{23} + 2\omega_x X_{23} \\ \dot{Y}_{23} &= r X_{23} - Y_{23} - (U^{(z)} X_{23} + U^{(x)} Z_{23}) - a_y Y_{23} - \alpha_{23}^y Y_{23} \\ \dot{Z}_{23} &= -b Z_{23} + (U^{(x)} Y_{23} + U^{(y)} X_{23}).\end{aligned}$$

To find where synchrony starts across edge 2-3, we first find the synchronization threshold for two oscillators coupled through one variable at a time. The double coupling strengths are  $a_x \approx 7.5$  and  $a_y \approx 2.8$  if connected through  $x$  and  $y$  respectively.

Now let's use the double coupling strengths in place of the three Lorenz oscillators. We'll start with the network coupled uniformly through  $x$  since the double coupling strength through  $x$  was larger than through  $y$ . Below shows this network structure. The coupling

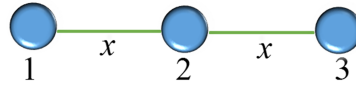


Figure 3.5: Three oscillators uniformly coupled through  $x$ .

strength value for this network gives  $c_{thr} \approx 7.5$ .

Now replacing edge 2-3 with  $y$  which is the original network, produces a lower synchronization threshold of 7. Therefore, replacing the  $x$ -edge with a  $y$ -edge improves synchronizability. Is this phenomena valid for any chain mixed-network? We will address this question in the next example.

### 3.2.1 Ten Oscillator Chain Network

Consider ten Lorenz oscillators shown in Fig. 3.6 for which each edge is uniformly coupled through  $x$ .

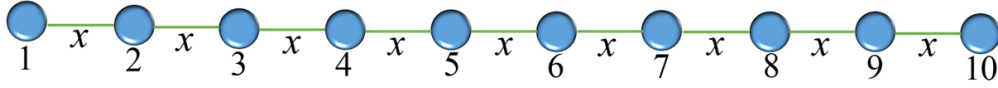


Figure 3.6: A network of ten Lorenz oscillators uniformly coupled through  $x$ .

In this regime, we test our previous conjecture done with three oscillators (Fig.3.5) to this network containing ten oscillators. Hence, let's replace edge 5-6 with a  $y$ -edge to determine if the auxiliary system can be stabilized through  $y$ .

Fig. 3.7 shows the modified network and (3.13) gives the auxiliary system used to stabilize this network through the  $y$ -edge.

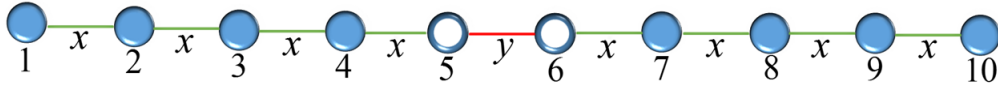


Figure 3.7: A network of ten Lorenz oscillators coupled through both  $x$  and  $y$ .

$$\begin{aligned}
 \dot{X}_{56} &= \sigma Y_{56} - \sigma X_{56} + \omega_x b_{56}^{xy-graph} X_{56} \\
 \dot{Y}_{56} &= r X_{56} - Y_{56} - (U^{(z)} X_{56} + U^{(x)} Z_{56}) - a_y Y_{56} - \alpha_{56}^x Y_{56} \\
 \dot{Z}_{56} &= -b Z_{56} + (U^{(x)} Y_{56} + U^{(y)} X_{56}).
 \end{aligned} \tag{3.13}$$

In Fig. 3.7, we see the effect of replacing different edges in the network with edges of the opposite type (replacing  $x$ -edges with  $y$ -edges). When a secondary edge (in terms of the number of paths passing through an edge), such as edge 9-10, is replaced, we see a drop in the synchronization threshold,  $d^*$ . This drop is because the  $x$ -edge is replaced with a more favorable link (in the sense that two  $y$ -coupled Lorenz oscillators require a weaker coupling to synchronize) and  $b_k^{x-graph}$  for all of the edges decreases. Hence, the synchronization threshold decreases. As more central edges are successively replaced, we see the synchronization threshold,  $d^*$ , increase and eventually no longer exist (meaning that the

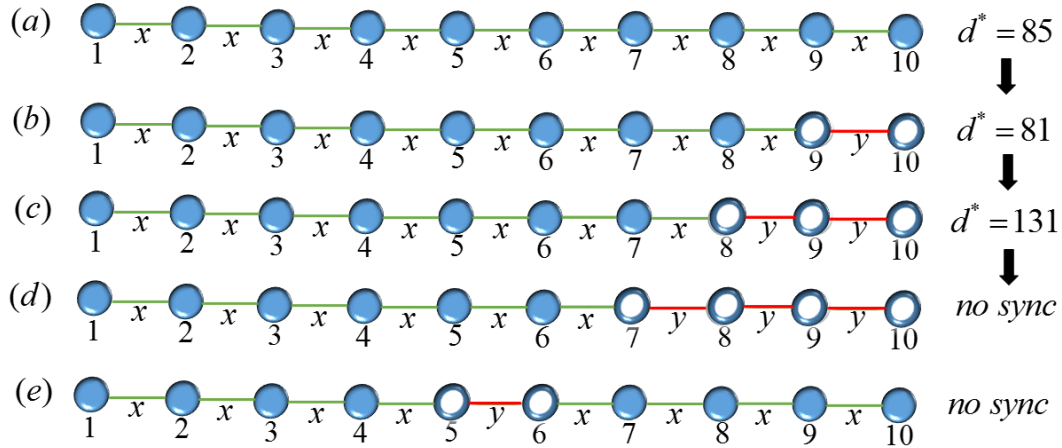


Figure 3.8: Networks of Lorenz systems with uniform coupling strength  $d$  for both  $x$ - and  $y$ -edges. The dependence of the synchronization threshold  $d^*$  on the location and type of edges (i.e. which edges are  $x$ -edges and which edges are  $y$ -edges). Notice that replacing a secondary  $x$ -edge with a presumably better converging  $y$ -coupling improves synchronization, as expected (b). Replacing edges that have more paths passing through them, causes the synchronization threshold to increase (c), and then no longer be attainable (d,e).

network is not synchronizable for any coupling  $d$ ). From the stability criteria (3.8) given in the previous chapter, this increase (and eventual break) in the synchronization threshold is likely caused by the give-and-take in the stabilization of the auxiliary systems.

This example leads us to the following conjecture: replacing a lightly loaded (in terms of paths passing through an edge) link with a stronger converging coupling via another variable may improve synchronizability as shown in Fig.3.8, but replacing a more highly loaded link can cause synchrony to no longer be attainable.

### 3.2.2 Six Oscillator Tree Network

Given the “synchrony break” phenomenon that occurs with the ten oscillator example, we wish to consider smaller examples to unveil the mechanism underlying synchrony break. We consider the 6-oscillator mixed network in Fig. 3.9. We begin with the network shown

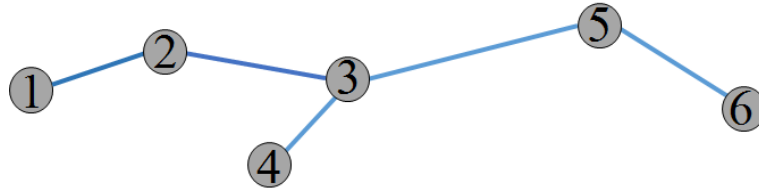


Figure 3.9: Network of six  $x$ -coupled Lorenz oscillators with uniform coupling. The blue edges represent  $x$ -coupling.

in Fig. 3.9. As explained in the four-oscillator chain network, we can find each edge's traffic load for the  $x$  connection graph in the same fashion. Thus,

$$\begin{aligned}
 b_{12}^x &= |P_{12}| + |P_{13}| + |P_{14}| + |P_{15}| + |P_{16}| = 13 \\
 b_{23}^x &= |P_{23}| + |P_{24}| + |P_{13}| + |P_{25}| + |P_{26}| + |P_{14}| + |P_{15}| + |P_{16}| = 20 \\
 b_{34}^x &= |P_{34}| + |P_{24}| + |P_{14}| + |P_{45}| + |P_{46}| = 11 \\
 b_{35}^x &= |P_{35}| + |P_{45}| + |P_{46}| + |P_{36}| + |P_{25}| + |P_{15}| + |P_{26}| + |P_{16}| = 20 \\
 b_{56}^x &= |P_{56}| + |P_{36}| + |P_{46}| + |P_{26}| + |P_{16}| = 13.
 \end{aligned} \tag{3.14}$$

With these preliminary “traffic load” values, we consider the following cases for study:

- (i) One edge-replacement, in which we replace only one edge in the  $x$  connection graph with a  $y$  edge. This results in multiple networks of four  $x$  edges and one  $y$  edge.
- (ii) Two edge-replacement, in which we replace two edges in the  $x$  connection graph with  $y$  edges. This results in multiple networks of three  $x$  edges and two  $y$  edges.
- (iii) Successive edge-replacement, in which we replace the edges in the order of their “traffic load” (from smallest to largest) as defined above.

We begin with the case (i), one edge replacement. We start with  $x$  connection graph shown in Fig. 3.9 then replace each edge with a  $y$ -edge in the order of the traffic load computed in (3.14). After computing the coupling threshold required to synchronize the new network, this edge reverts back to being an  $x$ -edge, as shown in Fig. 3.10. As the

network is no longer composed of just a  $x$  component, we must consider the sum of the path lengths in the  $x$  connection graph (given by  $b_k^{x-graph}$ ), the  $y$  connection graph (given by  $b_k^{y-graph}$ ), and the mixed connection graph (given by  $b_k^{mixed}$ ). The values of  $b_k^{mixed}$  for the different network configurations are given by:

$$\begin{aligned}
 b_{12}^{mixed} &= |P_{13}| + |P_{14}| + |P_{15}| + |P_{16}| = 12 \\
 b_{23}^{mixed} &= |P_{13}| + |P_{14}| + |P_{15}| + |P_{16}| + |P_{24}| + |P_{25}| + |P_{26}| = 19 \\
 b_{34}^{mixed} &= |P_{24}| + |P_{14}| + |P_{45}| + |P_{46}| = 10 \\
 b_{35}^{mixed} &= |P_{36}| + |P_{45}| + |P_{25}| + |P_{15}| + |P_{26}| + |P_{16}| + |P_{46}| = 19 \\
 b_{56}^{mixed} &= |P_{46}| + |P_{36}| + |P_{26}| + |P_{16}| = 10.
 \end{aligned}$$

The effect of replacing a single edge is shown in Fig. 3.10. When secondary edges (edges with fewer paths passing through them) are replaced, there is little to no effect on the synchronization threshold. However, when “bottleneck” edges (edges with many paths passing through them) are replaced, we observe the “synchrony break” phenomenon as shown in the example of ten Lorenz oscillators. It should be noted that the phenomenon is present when replacing both edge 2-3 and edge 3-5, because the network is symmetric. Not only were these the edges with the highest traffic load in the network connected only through  $x$ , but these are also the edges that have the most mixed paths that pass through them (the highest  $b_k^{mixed}$ ).

The role of “bottleneck” edges seems clear in regard to replacing one edge in a network (replacing bottleneck edges dramatically increases the synchronization threshold, and can even make synchronization unattainable). Now, we focus our attention on the role that these bottleneck edges play in a more complex network structure. To this end, we consider the two edge-replacement scheme (case (ii)).

Using the network labels in Fig. 3.11, we compute the  $b_k^{mixed}$  values:

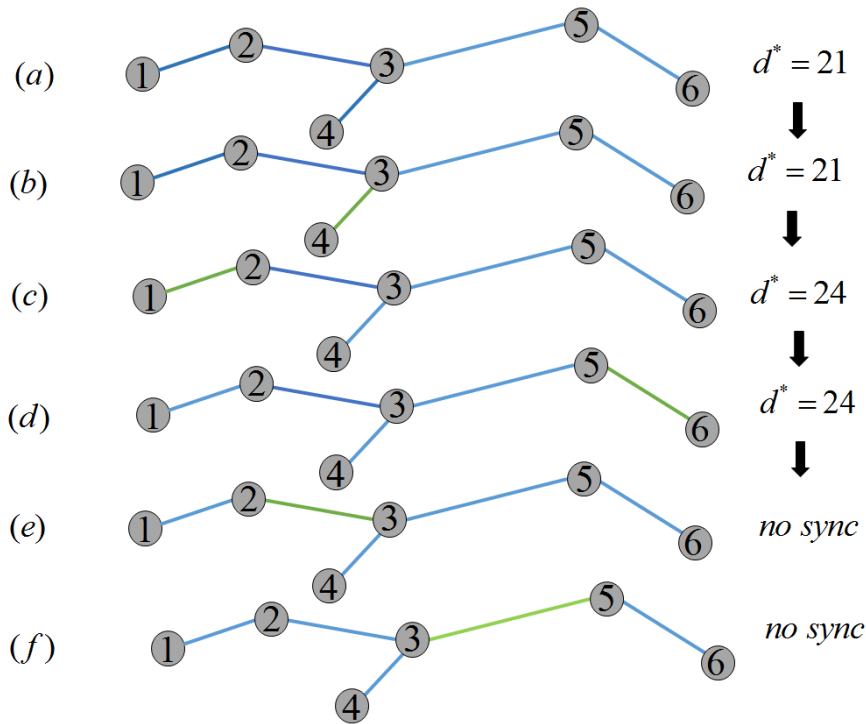


Figure 3.10: Network of six Lorenz oscillators shown in (a)-(f) with one  $x$ (blue)-edge replaced with a  $y$ (green)-edge, according to the one edge-replacement scheme (case (i)).  $d^*$  indicates the threshold on coupling required for the given network to synchronize. Replacing secondary  $x$ -edges with  $y$ -edges has almost no effect on the synchronization threshold ((b)-(d)). When  $x$ -edges contained in many paths in the connection graph are replaced by  $y$ -edges, synchronization is impossible, regardless of the magnitude of coupling ((e) and (f)).

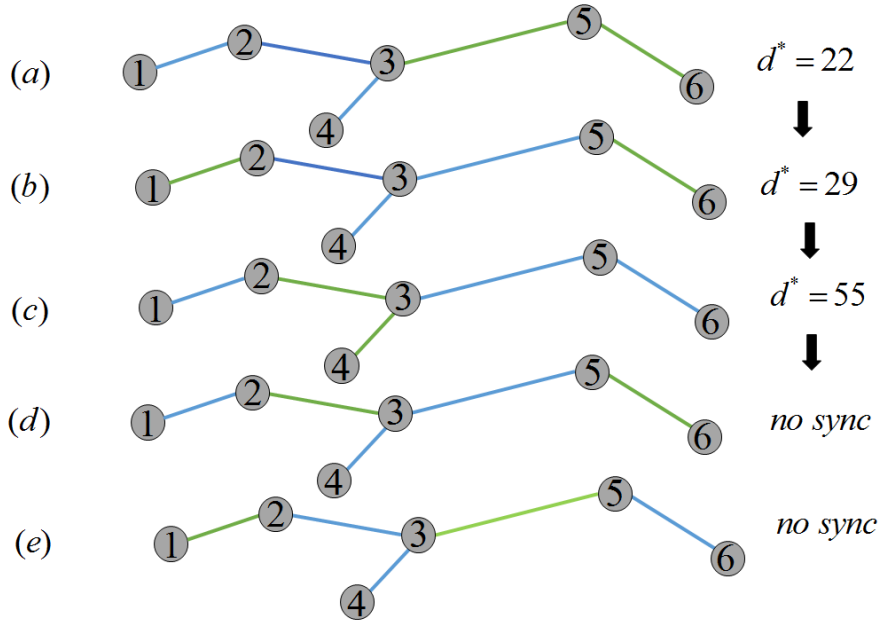


Figure 3.11: Network of six Lorenz oscillators shown in (a)-(e) with each  $x$ (blue)-edge replaced two at a time with a  $y$ (green)-edge. The coupling strength shown on the right of each figure depicts instability in the network as the highly loaded  $x$ -edges are replaced with  $y$ -edges.

	(a)	(b)	(c)	(d)	(e)
$b_{12}^{mixed}$	7	12	12	12	12
$b_{23}^{mixed}$	12	15	17	19	17
$b_{34}^{mixed}$	5	6	8	8	8
$b_{35}^{mixed}$	17	15	17	17	19
$b_{56}^{mixed}$	10	12	10	12	12

In Fig. 3.11, we see the role of replacing additional  $x$ -edge with  $y$ -edges. In (a), the two  $y$  edges are edge 3-5 and edge 5-6, a highly loaded and secondary edge, respectively. Even though a bottleneck edge is replaced for this network, synchrony does not break, because the graph is partitioned into an  $x$  component and a  $y$  component. This partitioning results in lower values of  $b_k^{mixed}$  (see the above table), and the partitions act as a whole, so the coupling threshold benefits from the favorable  $y$  coupling. In (b), the two  $y$  edges are edge 1-2 and edge 5-6, both of which are secondary edges. While the edges do not form a larger

$y$  subgraph, replacing secondary edges lowers the  $b_k^{mixed}$  values compared to even case (a) (see the above table). Despite the lower values for  $b_k^{mixed}$ , the coupling threshold  $d^*$  is larger for network (b) than network (a). This shows that  $b_k^{mixed}$  does not tell the whole story. Synchronization depends on  $b_k^{mixed}$ , but also on  $b_k^{x-graph}$  and  $b_k^{y-graph}$ . In (c), the two  $y$  edges are edge 2-3 and edge 3-4, a bottleneck edge and a secondary edge. This case differs from (a) in that the connected  $y$  subgraph divides the graph in to three partitions: a single  $x$ -edge, a  $x$  subgraph, and a  $y$  subgraph. This increases the maximum  $b_k^{mixed}$  and removes the benefit of having the partitioned graph. As a result, the synchronization threshold  $d^*$  is much larger than in networks (a) and (b). In cases (d) and (e), the two  $y$  edges are on a bottleneck edge and a secondary edge. In both examples we see a spike in  $b_k^{mixed}$  and synchrony is not attainable for any value of  $d$ .

The last replacement scheme we consider is successive replacement (iii). Successively replacing edges in the original six oscillator network (Fig. 3.9) dramatically changes the values of the traffic loads through the different connection graphs. It is through these dramatic changes that we hope to gain insight into the roles of these different network properties in regard to synchronization. We successively replace each  $x$ -edge with a  $y$ -edge as shown in Fig. 3.12. The successive edge replacement is done according to the ranking of each edge's traffic load for the  $x$  connection graph. All  $x$ -edges are replaced with one and not reverted back to a  $y$ -edge. Therein, turning the original  $x$ -connection graph to a  $y$  connection graph. The traffic load will now be composed of mixed paths as each  $x$ -edge has been replaced with a  $y$ -edge. Recall each  $x$ -edge is replaced based on the order of the traffic load in the  $x$ -connection graph. The table below shows all  $b_k^{mixed}$  values as the network topology changes due to each  $x$ -edge successively replaced with a  $y$ -edge:

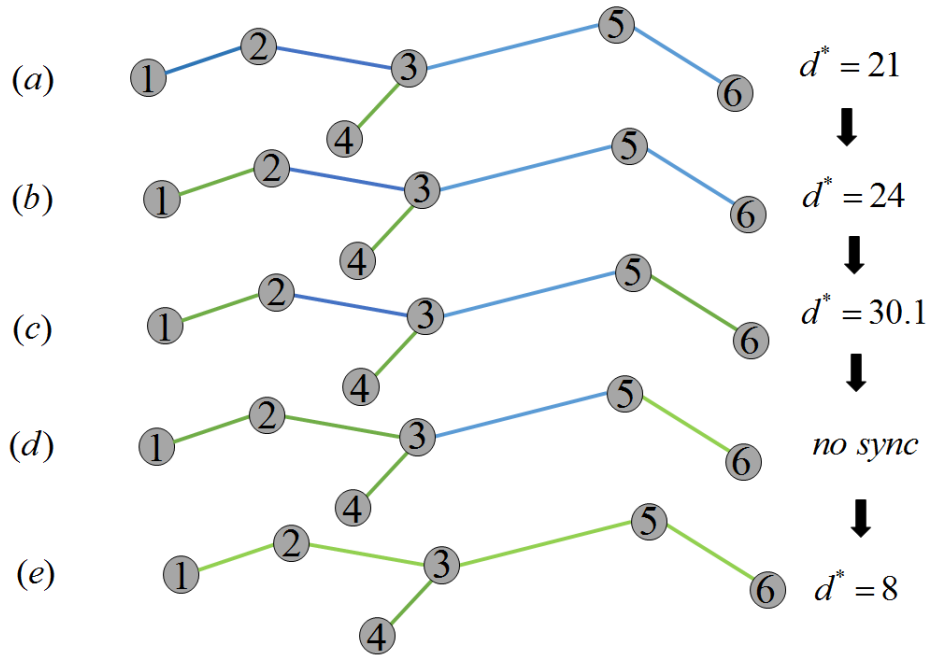


Figure 3.12: Network of six Lorenz oscillators shown in (a)-(e) with each  $x$ (blue)-edge successively replaced with a  $y$ (green)-edge. The coupling strength shown on the right of each figure depicts instability in the network as the highly loaded  $x$ -edges are replaced with  $y$ -edges.

	(a)	(b)	(c)	(d)	(e)
$b_{12}^{mixed}$	3	12	12	7	0
$b_{23}^{mixed}$	5	14	17	12	0
$b_{34}^{mixed}$	10	10	10	5	0
$b_{35}^{mixed}$	10	12	17	19	0
$b_{56}^{mixed}$	3	7	12	12	0

In Fig. 3.12, we see the effect that changing the network topology according to the successive replacement schema on the coupling threshold  $d^*$ . When successively replacing secondary edges (see networks (a), (b) and (c) in Fig. 3.12) we more-or-less see the same results from the one edge-replacement networks. Replacing secondary edges has a very minor impact on the coupling threshold  $d^*$ . However, we see the synchrony break phenomenon when every edge except for edge 3-5 has been replaced. This is because the network is the inverted

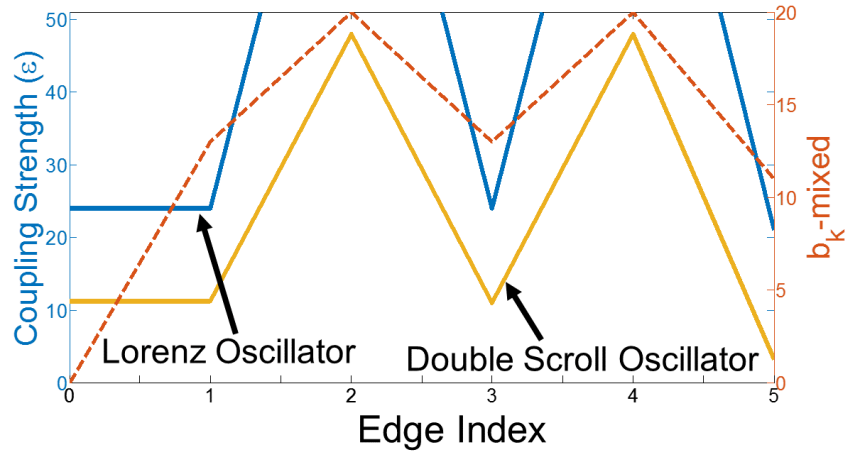


Figure 3.13: Coupling thresholds for six Lorenz and Chua (Double-Scroll) oscillators after replacing one  $x$ -edge to a  $y$ -edges and their mixed traffic load. The blue and yellow lines represent Lorenz and Chua coupling strengths, respectively. The red line represents the traffic load for the mixed edges.

version of network (e) in Fig. 3.13, so the same reasoning for synchrony break is used here. The  $b_k^{mixed}$  values are much larger when replacing a bottleneck edge, and the auxiliary systems are no longer able to be stabilized. Lastly, when the final edge has been replaced, we see synchrony return, because now the network is composed of just a  $y$  connection graph.

When doing the one edge replacement, we see that synchrony breaks when edge 2-3 and edge 4-5 are replaced, as shown in Fig. 3.13. On the other hand, in Fig. 3.14, the region where synchrony breaks is after the 4<sup>th</sup> edge replacement. The reason for the break in synchrony occurring in different places is because of how the replacements affect  $b_k^{mixed}$ .

Now that we used numerical examples to get a feel for the role of the mixed network structure on the synchronization threshold, we will compare the theoretical thresholds computed using (3.8) to the corresponding numerical thresholds. This is to gain insight into how we can overcome the

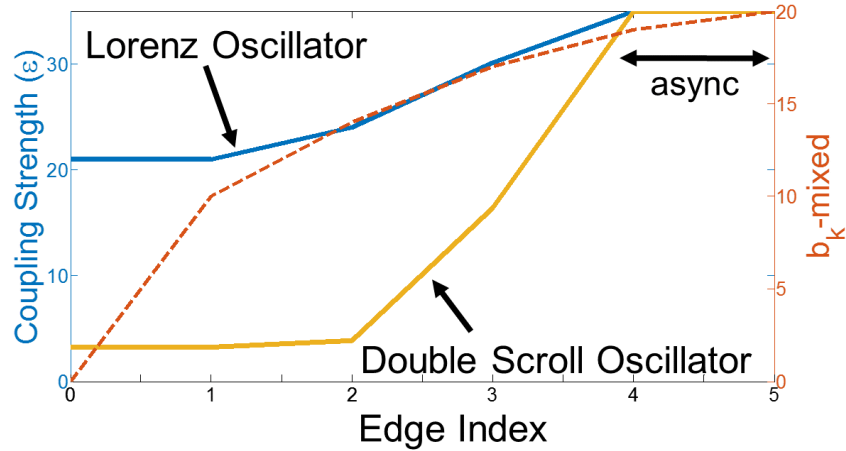


Figure 3.14: Coupling thresholds for six Lorenz and Chua (Double-Scroll) oscillators after replacing each  $x$ -edge to  $y$ -edges successively and the mixed traffic load. The blue and yellow lines represent Lorenz and Chua coupling strengths, respectively. The red line represents the traffic load for the mixed edges.

### 3.3 Numerical vs Theoretical Synchronization Thresholds

In the subsections above, we discovered how much the highly loaded edge impacts the synchronization threshold for the network (in the six and ten oscillator networks). In this section, we see the same phenomena in the 20-oscillator mixed network, shown in Fig. 3.15. We consider two different network schemas: (i) one edge replacement and (ii) successive edge replacement. For one edge replacement (i), we begin with the 20-oscillator mixed network topology from Fig. 3.15. We replace an edge with the opposite connection (for example replacing a  $x$  (black) edge with a  $y$  (gray) edge), we compute the synchronization threshold, and then revert the edge to its original coupling type. In Fig. 3.16, we summarize the results of this investigation.

For one edge replacement (i), we first compute the maximum value of  $b_k^{mixed}$  for each edge replacement.  $b_k^{mixed}$  (the sum of the mixed paths passing through an edge) is computed using the  $b_k^{mixed}$  code found in Appendix D. While the value of  $b^{mixed}$  depends heavily on the choice of paths from oscillator to oscillator, we use the natural choice of the shortest paths

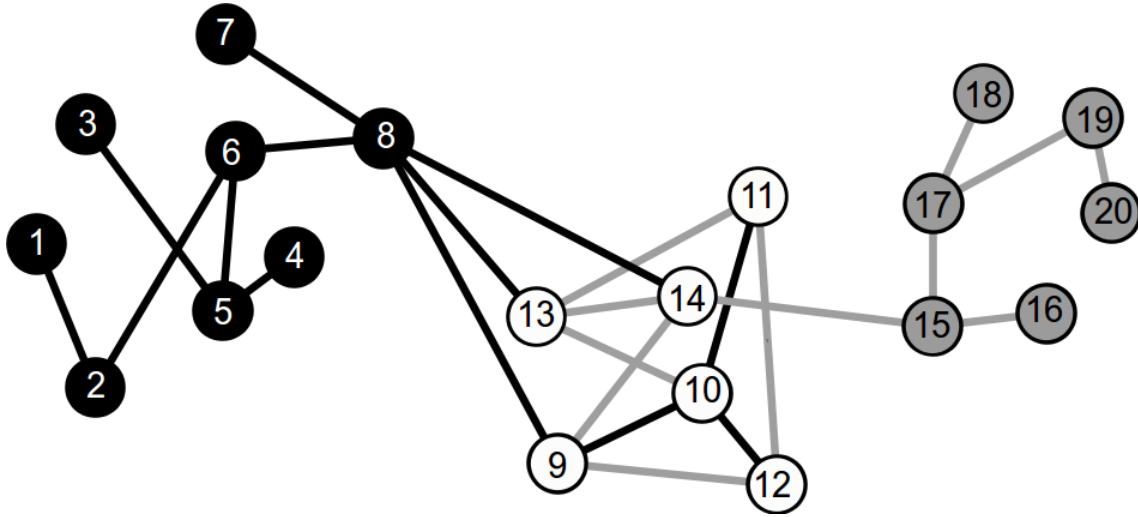


Figure 3.15: 20 oscillator mixed-network with an  $x$  connection graph (black edges), and a  $y$  connection graph (gray edges). Moreover the grey oscillators belong to the  $x$ -graph, black oscillators belong to the  $y$ -graph and white oscillators belong to the mixed subgraph.

(computed using Dijkstra's Algorithm). It is of vital importance that the choice of path is consistent. Then, we index the edges according to this value  $\max b_k^{mixed}$ . The values of  $\max b_k^{mixed}$  range from 0 (for edge 11-12, for which there are alternative, more favorable paths avoiding this edge) to  $\approx 400$  (for edge 14-15, for which every path from the  $x$  component to the  $y$  component must pass through).

After indexing the edges, we compute the synchronization threshold for replacing the corresponding edge with the opposite connection type. In Fig. 3.16, we see that replacing secondary edges (edges with low  $\max b_k^{mixed}$  values) has little effect on the synchronization threshold for both the Lorenz and Double-Scroll networks (see the blue and red curves, respectively). However, as soon as an edge that is sufficiently highly loaded (the number of paths passing through an edge exceeds a certain threshold), synchrony is no longer attainable (see the region of asynchrony that begins with replacing edge 19). The dip in in the region of asynchrony (corresponding to a return of synchronization for that edge replacement) is caused by replacing an edge that is a bottleneck if we only consider the shortest path, but

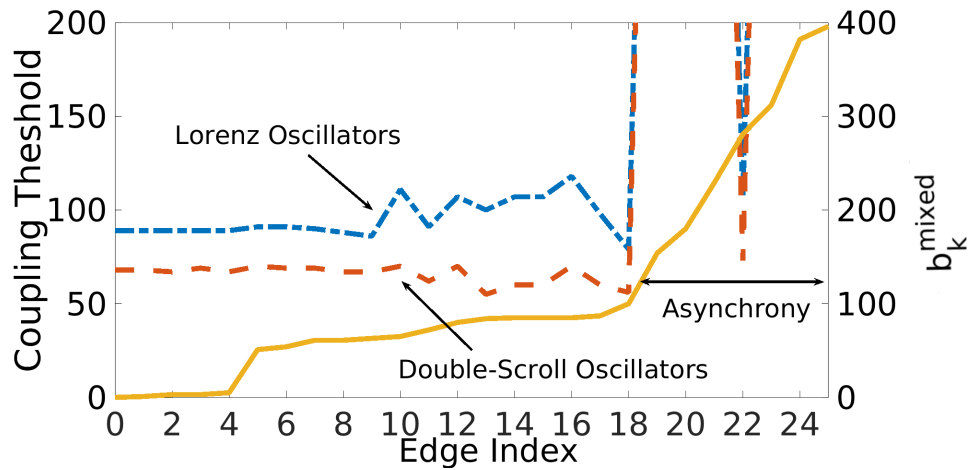


Figure 3.16: The relationship between coupling strength and traffic load. The coupling thresholds (strength) on the left  $y$ -axis is the coupling strength necessary to synchronize the 20-oscillator network in Fig. 3.15 after replacing each  $x$ -edge with a  $y$ -edge one at a time; the edge index was determined by the traffic load passing through the edges comprising of mixed paths. When an edge with a traffic load higher than a certain threshold (beginning with edge 18), the network becomes unsynchronizable. The phenomena is consistent for both a Lorenz network (blue) and double scroll network (red). The yellow curve depicts the traffic load for the respective edges.

there are alternative (slightly longer) paths that do not use the edge, meaning it is not a bottleneck edge for other path choices.

For successive edge replacement (ii), we apply the same edge indexing as in the one edge replacement schema (i). Next, we begin with the network in Fig. 3.15, except where all of the edges are  $x$  (black) edges. Starting with edge 1, we successively replace edge  $k$  with a  $y$  edge and compute the coupling threshold for synchronization, until the entire network is coupled through  $y$ . Once again, in Fig. 3.17 we see that replacing secondary edges with the opposite type of coupling has almost no effect on the synchronization threshold. When edge 12 is replaced, we see that synchronization becomes unattainable, until the network is completely coupled through  $y$ . A significant observation from this experiment is that there is no synchrony when the network is composed of 24  $y$  edges and one  $x$  edge which is the most highly loaded edge. This is exactly the phenomenon that we saw in the six and ten oscillator networks, except this network is much more complex. Replacing just one edge in a

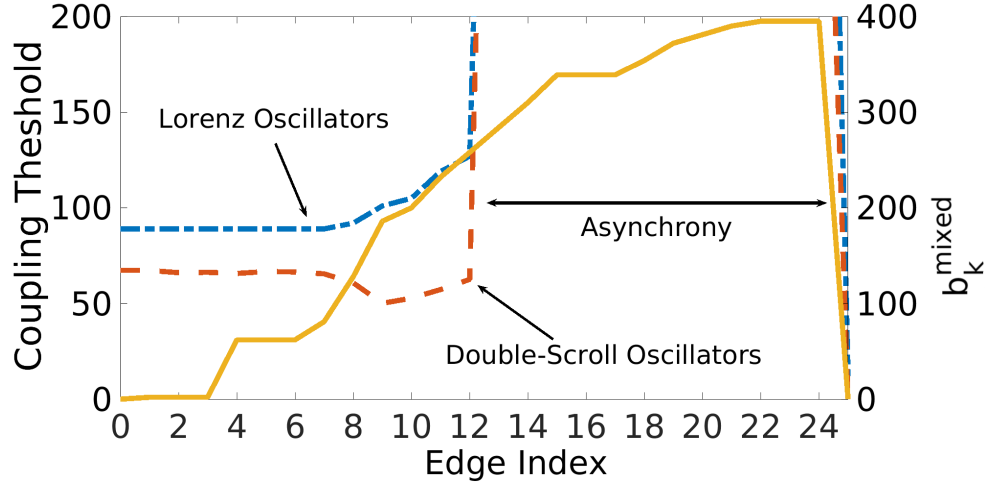


Figure 3.17: The relationship between coupling strength and traffic load due to successive  $y$ -edge replacement. The coupling strength on the left  $y$ -axis is the coupling thresholds(strength) necessary to synchronize the 20-oscillator network in Fig. 3.15 after successively replacing each  $x$ -edge with a  $y$ -edge; the edge index was determined by the traffic load passing through the edges comprising of mixed paths. When an edge with a traffic load higher than a certain threshold (beginning with edge 12), the network can no longer be synchronized. The phenomena is consistent for both a Lorenz network (blue) and double scroll network (red). The yellow curve depicts the traffic load for the respective edges.

20 oscillator network can destroy the network's ability to synchronize. We see similar results for networks of other oscillators and when reversing the roles of the two edge types.

To summarize, for both cases we see that when an edge with sufficiently high  $b^{mixed}$  is replaced with the opposite edge type, synchronization is no longer possible in the network, regardless of the coupling strength. This is a highly unexpected phenomenon. If we replace an  $x$  edge with a more favorable  $y$  edge, if the edge has too many mixed paths traversing through it, this edge replacement 'breaks' synchronization in the network. With this surprising result in mind, we turn to our newly proven Mixed Connection Graph method for insights into this non-intuitive 'synchrony break' phenomenon.

In (2.32), the stability criterion for a network are stated as  $c_{ij} \equiv c_k > \frac{1}{n} \left\{ a_x \cdot b_k^{x-graph} + \omega_x \cdot b_k^{mixed} + \alpha_k^x \right\}$  for  $x$ -coupled edges, and  $d_{ij} \equiv d_k > \frac{1}{n} \left\{ a_y \cdot b_k^{y-graph} + \omega_y \cdot b_k^{mixed} + \alpha_k^y \right\}$  for  $y$ -coupled edges. We consider uniform coupling and without loss of generality we consider

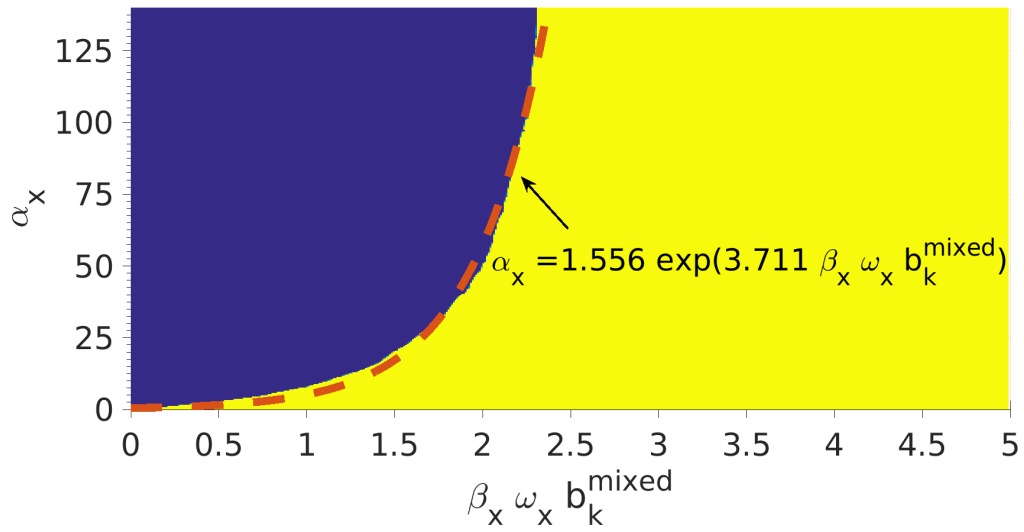


Figure 3.18: Principle diagram indicating the dependence of stability of the  $x$ -auxiliary system (3.10) on the mixed connection graph. Blue (dark) indicates stability, while yellow (light) represents instability. The dependence of  $\alpha_x$  on  $b^{\text{mixed}}$ ,  $\omega_x$ , and  $\beta_x$  is estimated by the exponential function  $1.556 \exp(3.711 \beta_x \omega_x b_k^{\text{mixed}})$  (dashed curve), and is used for the theoretical fit in Fig 3.3(left). The stability diagram for the  $y$ -auxiliary system in (3.11) is similar, but the vertical asymptote occurs at  $\beta_y b_k^{\text{mixed}} \omega_y \approx 12$ .

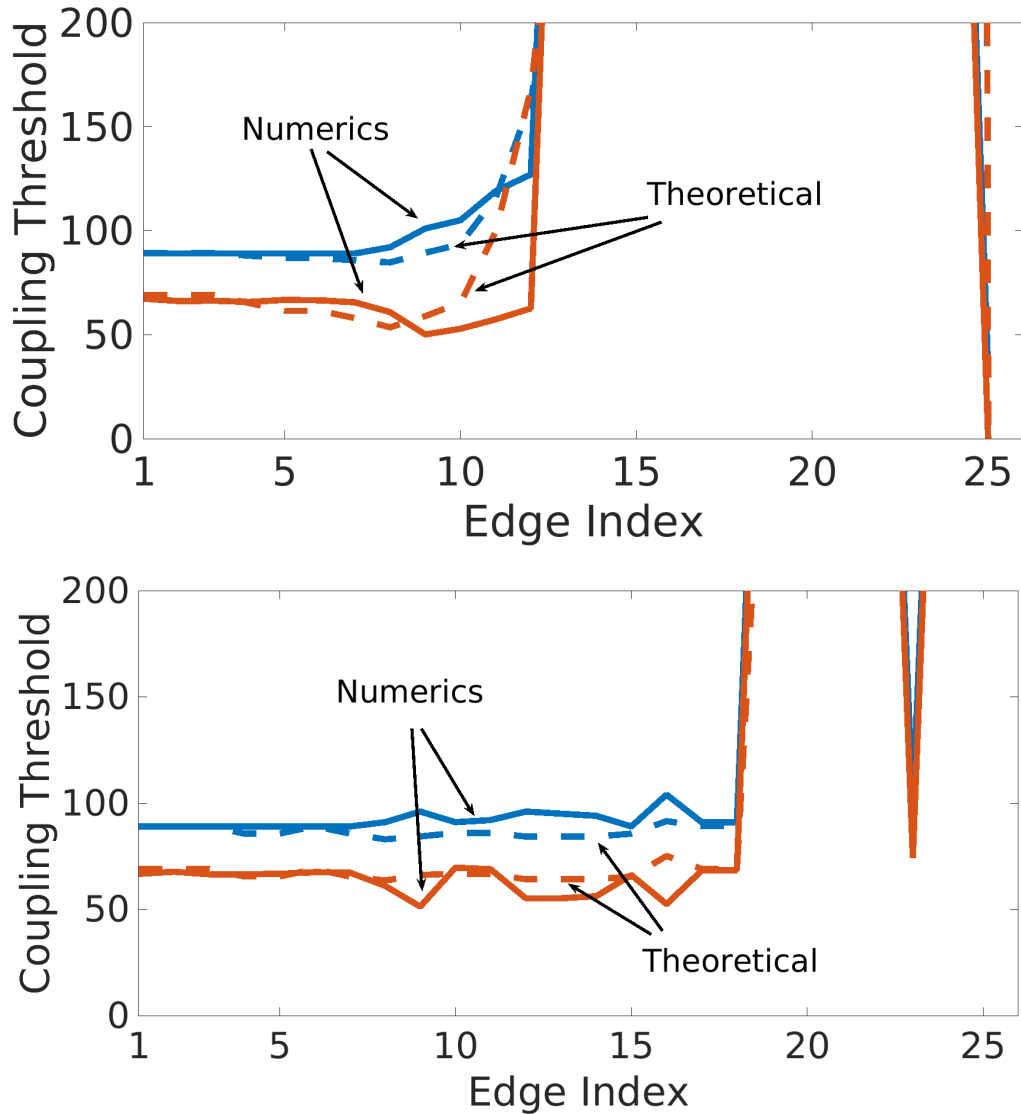


Figure 3.19: Synchronization coupling thresholds for the network of Lorenz (solid light blue lines) and Double-Scroll (solid dark blue lines) oscillators compared to the theoretical bounds using the stability criterion in (2.32). In (a), the network is uniformly coupled through  $x$ , with edges successively replaced with  $y$  edges. In (b), the network topology is given by Fig. 3.15, and edges are replaced with the opposite type of coupling, then reverted after computing the coupling threshold.

only the bound for  $x$ -coupled edges,  $c_{ij}$  (as  $x$ -coupling is less favorable for synchronization than  $y$ -coupling). Let  $\varepsilon = \max\{c_{ij}\}$ . Computing  $a_x$ ,  $a_y$ ,  $b_k^{x-graph}$ , and  $b_k^{y-graph}$  is straight forward.  $a_x$  and  $a_y$  are twice coupling strength needed to synchronize a network of two  $x$  or  $y$  coupled oscillators, respectively.  $b_k^{x-graph}$  and  $b_k^{y-graph}$  are the sum of the path lengths of the  $x$  or  $y$  connection graph, respectively, passing through edge  $k$ .  $\omega_x$  and  $\omega_y$  are chosen such that the mixed auxiliary system (3.9) is stable (for our example, let  $\omega_x = 6$  and  $\omega_y = 0.5$ ).

The last step in the method is to find the values of  $\alpha_x$  and  $\alpha_y$  necessary to stabilize the auxiliary systems given in (3.10) and (3.11), respectively. Here, because the  $x$ -coupling is the primary obstacle for synchronization, we will only compute  $\alpha_x$ , however, in general both  $\alpha_x$  and  $\alpha_y$  are necessary when implementing the method. In Fig. 3.18 we show the role of  $\alpha_x$  in stabilizing (3.10), and its dependence on  $\omega_x$  and  $b^{mixed}$ . In general  $b^{mixed}$  causes the bound on the coupling threshold to be too conservative. To compensate for this, we include the scaling factor  $\beta_x$ . Subsequently, we can compute  $\alpha_x$  as a function of  $\omega_x$ ,  $b^{mixed}$ , and the scaling factor,  $\beta_x$ . This means that the stability of the auxiliary systems explicitly depends on the mixed connection graph. Furthermore, we see in Fig. 3.18 asymptotic behavior for  $\alpha_x$ , implying that if  $b_k^{mixed} > b_k^*$ , the auxiliary system *cannot* be stabilized. This means that the original network cannot be synchronized for any coupling value.

Using the values described above, and the approximation of the function for  $\alpha_x$ ,  $\alpha_x = 1.556 \exp(3.711\beta_x\omega_x b_k^{mixed})$ , we are able to compute the coupling threshold necessary to synchronize the network created in each step of case (a) and (b). These thresholds computed using the method (adjusted for the overestimates given by  $b_k^x$  and  $b^{mixed}$ ) are shown in Fig. 3.19. Not only does the method predict the ‘break’ of synchronization observed by the numerical computation of the coupling thresholds, but the theoretical fit for the coupling thresholds matches the numerical values remarkably well.

### 3.3.1 Another Twenty Oscillator Network

Lastly, not all networks produced this phenomena. We consider the network topology of 20 oscillators shown in Fig. 3.20.

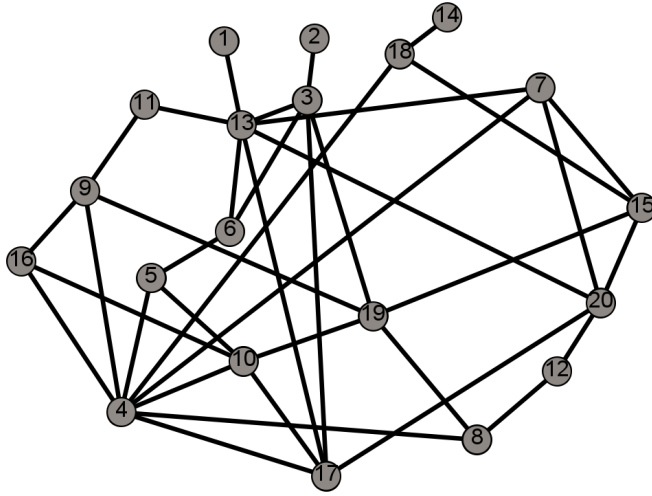


Figure 3.20: A complex network of 20 oscillators in which the network topology contains cycles.

In the same vein as our previous numerical examples, we consider two cases: (i) one edge replacement (in which we only replace one  $x$  edge in the network with a  $y$  edge at a time) and (ii) successive replacement (in which we successively replace  $x$  edges with  $y$  edges until the entire network is coupled only through  $y$ ). In this example, we also index the edges according to the value of  $\max_k \{b_k^{mixed}\}$  in case (i).

For one edge replacement, case (i), which is summarized in Fig. 3.21 (top), we see that replacing any one  $x$  edge in the network with a  $y$  edge has almost no effect on the synchronization threshold in the network. This is the case, regardless of the number of paths passing through the edge being replaced. The network is so highly connected that there are many alternate paths from one oscillator to another that are not much longer than the shortest path, so the magnitude and impact of  $b_k^{mixed}$  is much less dramatic.

For successive replacement, case (ii), which is summarized in Fig. 3.21 (bottom), we see more complicated behavior than for one edge replacement. After a certain number of secondary  $x$  edges are replaced with  $y$  edges, the synchronization threshold *drops*. This is because the edges being replaced form a small  $y$  subgraph in the network, and  $y$  edges

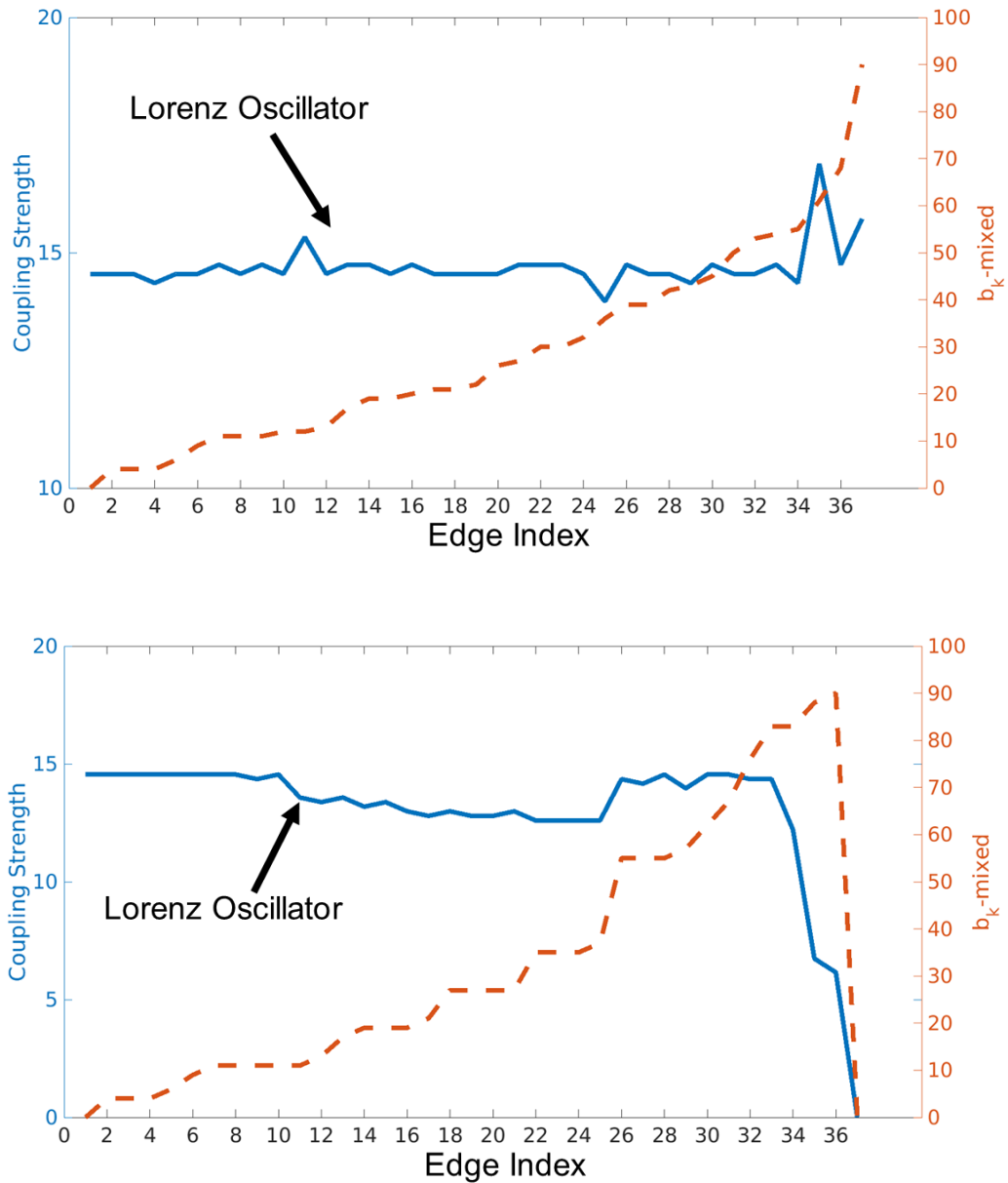


Figure 3.21: The relationship between Lorenz coupling thresholds (blue) and traffic load (dashed line) for the network in Fig. 3.20 for both the one edge (top) and successive (bottom) replacement schemas. (top) Replacing only one  $x$  edge in the network with a  $y$  edge has no noticeable effect on the threshold for coupling required to synchronize the network. (bottom) Replacing  $x$  edges successively with  $y$  edges has no effect on the synchronization threshold for the first 10 secondary edges. Then, as the  $y$  edges form a subgraph, the synchronization threshold drops (between index 11 and 25). Then, when a bottleneck edge (edge 26) is replaced both  $b_k^{mixed}$  and the synchronization threshold spike. Lastly, as the network becomes coupled only through  $y$ , the synchronization threshold plummets.

inherently have lower synchronization thresholds. Then, after replacing edge 26, we see the synchronization threshold jump, along with  $b_k^{mixed}$ , which indicated that a bottleneck edge in the network has been replaced. Lastly, as the network finally transforms into a network that is only coupled through  $y$ , the coupling threshold plummets, as  $y$  synchronize networks at lower thresholds than  $x$  edges.

Using the numerical examples in this section, we have gained a wealth of insight into the interplay between the topology of the mixed network and synchronization. Furthermore, we have shown that using a scaling factor to compensate for the overestimates in the general stability criteria we can predict the numerical synchronization thresholds quite well. In general, in sparse networks replacing secondary edges that have few pass that pass through them has almost no effect on the synchronization threshold. Whereas in sparse networks replacing bottleneck edges will dramatically increase the synchronization threshold, or even make the network unsynchronizable. For more highly connected edges, we see less impact from the “traffic load” on the edges, because of the existence of alternate paths that are not usually much more expensive to take.

## Chapter 4

### CONCLUSIONS

While the study of synchronization in dynamical networks has gained significant momentum, the vast majority of rigorous mathematical analysis focus on the case of networks whose oscillators are connected via the same variables. Despite significant interest from both theoretical and application view points, synchronization in networks with mixed coupling has not been fully understood. This is in particular due to the inability of the existing eigenvalue-methods to give detailed insight into the stability condition of synchronization as the eigenvalues, corresponding to connection graphs composing a mixed network, must be calculated via simultaneous diagonalization of two or more connectivity matrices. Evidently, this simultaneous diagonalization is impossible in general, unless the matrices commute. As a result, synchronization in mixed network is usually studied numerically on a case to case basis and therefore, the exact role of network topology and the addition or exchange of edges remains unclear.

In this thesis, we have closed this gap and developed a new stability method, called the Mixed Graph Stability method. This method handles mixed dynamical networks, which are out of reach for the existing approaches, unless a restricted class of network topologies is considered. This method links the Lyapunov function approach with graph theoretical quantities. It contributes to understanding synchronization properties of mixed dynamical networks and reveals a number of striking, unexpected effects not seen in networks with one connection graph. In particular, using the method, we have demonstrated that replacing a lightly loaded link with a stronger pairwise converging coupling (a “good” link) via another variable may improve synchronizability, as one would expect. At the same time, such a replacement of a highly loaded link may essentially worsen synchronizability and make the network unsynchronizable, turning the pairwise stabilizing “good” link into a trouble maker.

The method is applicable to small or large networks with arbitrary topologies. A key ingredient of the method is the calculation of traffic loads via a given edge on the mixed connection graph. In small networks, these traffic loads can be explicitly calculated by hand with only moderate effort. In large networks, for example in the one displayed in Fig.(4.1), this calculation should be performed by our algebraic algorithms, implemented as MATLAB and PYTHON codes and given in Appendices B-E.

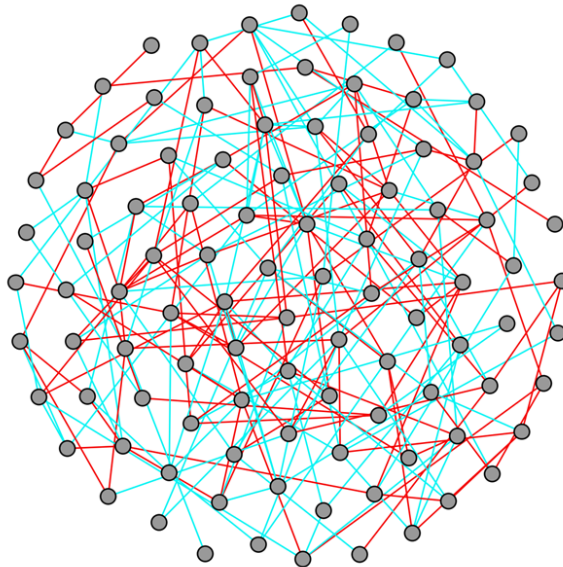


Figure 4.1: Mixed network with 100 oscillators where  $x$ -edges are colored in cyan and  $y$ -edges are colored in red.

In this thesis, we have limited our attention to undirected networks with two connection graphs. In the future, we plan to extend our general method to *directed* mixed networks composed of more than two subnetworks. This can be done by adapting the Generalized Connection graph method [45] which relies on transforming a directed graph into an undirected one. This is done by symmetrizing and augmenting the graph and associating a weight to each path. This weight involves the node unbalance of the two nodes. This extension is not straightforward. It requires overcoming a number of technically challenging issues and remains a subject of future study.

## REFERENCES

- [1] D. Carter, Jr. and S. Pandit, “*A Linear Variation Of Parameters Formula and Comparison Theorems For Three Dimensional Hyperbolic Problems,*” *Dynamic Systems and Applications*, vol. 5, no. 3, pp 423-430, 1996
- [2] D. Carter, Jr., S. Pandit and Y. Valaulikar, “*Monotone Iterative Schemes For Initial-Boundary and Periodic-Boundary Value Hyperbolic Problems in Three Independent Variables,*” *Nonlinear World*, vol. 4, no. 1, pp. 19–30, 1997
- [3] D.J. Watts and S. Strogatz, “*Collective dynamics of small-world networks,*” *Nature*, vol. 393, pp. 409-410, 1998.
- [4] S.H. Strogatz, “*Exploring complex networks,*” *Nature*, vol. 410, pp. 268-276, 2001.
- [5] A.L. Barabási and R. Albert, “*Emergence of scaling in random networks,*” *Science*, vol. 286, pp. 509-512, 1999.
- [6] R. Albert and A.-L. Barabási, “*Statistical mechanics of complex networks,*” *Review of Modern Physics*, vol. 74, pp. 49-98, 2002.
- [7] M. Girvan and M.E.J. Newman, “*Community structure in social and biological networks,*” *Proc. Natl. Acad. Sci. USA*, vol. 99, pp. 7821-7826, 2002.
- [8] M.E.J. Newman, “*The structure and function of complex networks,*” *SIAM Review*, vol. 45, pp. 167-256, 2003.
- [9] J. Buck and E. Buck, “*Synchronous fireflies,*” *Scientific American*, vol. 234, pp. 74-85, 1976.
- [10] R. Mirollo and S. Strogatz, “*Synchronization of pulsed-coupled biological oscillators,*” *SIAM Journal on Applied Mathematics*, vol. 50, pp. 1645-1662, 1990.

- [11] J.J. Collins and I. Stewart, “*Coupled nonlinear oscillators and the symmetries of animal gaits*,” *Science*, 3, pp. 349-392, 1993.
- [12] M.R. Guevara, A. Shrier, and L. Glass, “*Phase-locked rhythms in periodically stimulated heart cell aggregates*,” *American Journal of Physiology*, vol. 254, H1-H10, 1988.
- [13] J. Honerkamp, “*The heart as a system of coupled nonlinear oscillators*,” *Journal of Mathematical Biology*, v. 19, pp. 69-88, 1983.
- [14] V. Torre, “*A theory of synchronization of two heart pacemaker cells*,” *Journal of Theoretical Biology*, v. 61, pp. 55-71, 1976.
- [15] T.I. Netoff and S.J. Schiff, “*Decreased neuronal synchronization during experimental seizures*,” *Journal of Neuroscience*, vol. 22, pp. 7297–7307, 2002.
- [16] K. Schindler, C.E. Elger, K. Lehnertz, “*Increasing synchronization may promote seizure termination: Evidence from status epilepticus*” *Clin. Neurophysiol.*, vol. 118, pp. 1955-68, 2007.
- [17] E. W. Dijkstra, “*A Note on Two Problems in Connexion with Graphs*,” *Numerische Mathematlk I*, pp. 269 - 271, 1959.
- [18] G. Chen and Y. Xinghuo, “*Chaos Control Theory and Applications Series: Lecture Notes in Control and Information Sciences*,” Springer Series vol. 292, 2003.
- [19] L. Glass and M.C. Mackey, “*From Clocks to Chaos: The Rhythms of Life*,” Princeton, NJ: Princeton University Press, 1988.
- [20] I. Belykh, E. de Lange, M. Hasler, “*Synchronization of bursting neurons: what matters in the network topology*,” *Phys. Rev. Lett.*, vol. 94, 188101, 2005.
- [21] I. Belykh and A. Shilnikov, “*When weak inhibition synchronizes strongly desynchronizing networks of bursting neurons*,” *Phys. Rev. Lett.*, vol. 101, 078102, 2008.

- [22] A.E. Motter, S.A. Myers, M. Anghel, T. Nishikawa, “*Spontaneous synchrony in power-grid networks,*” *Nature Phys.* vol. 9, no. 3, pp. 191-197, 2013.
- [23] F. Sorrentino and E. Ott, “*Adaptive synchronization of dynamics on evolving complex networks,*” *Phys. Rev. Lett.*, vol. 100, 114101, 2008.
- [24] P. So, B. Cotton, and E. Barreto, “*Synchronization in interacting populations of heterogeneous oscillators with time-varying coupling,*” *Chaos*, vol. 18, 037114, 2008.
- [25] P. So and E. Barreto, “*Generating macroscopic chaos in a network of globally coupled phase oscillators,*” *Chaos*, vol. 21, 033127, 2011.
- [26] F. Sorrentino, “*Synchronization of hypernetworks of coupled dynamical systems,*” *New J. Phys.*, vol. 14, 033035, 2012.
- [27] D. Irving and F. Sorrentino, “*Synchronization of dynamical hypernetworks: dimensionality reduction through simultaneous block-diagonalization of matrices,*” *Phys. Rev. E*, vol. 86, 056102, 2012.
- [28] Ranta, E., Kaitala, V., Lindström, J. & Lindén, H. “*Synchrony in population dynamics,*” *Proceedings of the Royal Society of London B262*, 113-118, 1995.
- [29] Earn, D., Levin, S. & Rohani, P., “*Coherence and conservation,*” *Science*, vol. 270, 1360-1364, 2000.
- [30] Colombo, A., Dercole, F. & Rinaldi, S., “*Remarks on metacommunity synchronization with application to prey-predator systems,*” *The American Naturalist*, vol. 171 (4), 430-442, 2008.
- [31] Blasius, B., Huppert, A., & Stone, L., “*Complex dynamics and phase synchronization in spatially extended ecological systems,*” *Nature*, vol. 399, 354-359, 1999.
- [32] Belykh, I., Piccardi, C. & Rinaldi, S., “*Synchrony in tritrophic food chain metacommunities,*” *J. Biol. Dynamics*, vol. 3, pp. 497-514, 2009.

- [33] I. Hanski and I.P. Woiwod, “*Spatial synchrony in the dynamics of moth and aphid populations,*” J. Animal Ecology, vol. 62, 656668, 1993.
- [34] D.M. Johnson, A.M. Liebhold, O.N. Bjørnstad, and M.L. McManus, “*Circumpolar variation in periodicity and synchrony among gypsy moth populations,*” J. Animal Ecology, vol. 74, 882-892, 2005.
- [35] K. Higgins, A. Hastings, J.N. Sarvela, and L.W. Bostford, “*Stochastic dynamics and deterministic skeletons: population behavior of Dungeness crab,*” Science, vol. 276, pp. 1431-1435, 1997.
- [36] Cattadori, I.M., Hudson, P.J., Merler, S. & Rizzoli, A., “*Synchrony, scale and temporal dynamics of rock partridge (*Alectoris graeca saxatilis*) populations in the dolomites,*” J. Animal Ecology, vol. 68, 540-549, 1999.
- [37] C.S. Elton, and M. Nicholson, “*The ten-year cycle in numbers of the lynx in Canada,*” J. Animal Ecology, vol. 11, 215244, 1942.
- [38] S. Boccaletti, V. Latora, Y. Moreno, M. Chaovez, and D.-U. Hwanga, “*Complex networks: structure and dynamics,*” Physics Reports, vol. 424, pp. 175-308, 2006.
- [39] C.W. Wu, “*Algebraic connectivity of directed graphs,*” Linear and Multilinear Algebra, vol. 53, pp. 203-223, 2005.
- [40] L.M. Pecora and T.L. Carroll, “*Master stability function for synchronized coupled systems,*” Phys. Rev. Lett., vol. 80, pp. 2109-2112, 1998.
- [41] M. Barahona and L.M. Pecora, “*Synchronization in small-world systems,*” Phys. Rev. Lett., vol. 89, 0112023, 2002.
- [42] V. Belykh, I. Belykh, and M. Hasler, “*Connection graph stability method for synchronized coupled chaotic systems,*” Physica D, vol. 195, pp. 159-187, 2004.

- [43] I. Belykh, M. Hasler, M. Lauret, and H. Nijmeijer, “*Synchronization and graph topology*,” Int. Journal of Bifurcation and Chaos, vol. 15, pp. 3423-3433, 2005.
- [44] I. Belykh, V. Belykh, and M. Hasler, “*Synchronization in asymmetrically coupled networks with node balance*,” Chaos, vol. 16, 015102, 2006.
- [45] I. Belykh, V. Belykh, and M. Hasler, “*Generalized connection graph method for synchronization in asymmetrical networks*,” Physica D, vol. 224, pp. 42-51, 2006.
- [46] T. Nishikawa and A.E. Motter, “*Network synchronization landscape reveals compensatory structures, quantization, and the positive effect of negative interactions*,” Proc. Natl. Acad. Sci. USA, vol. 107, 10342, 2010.
- [47] A. Pikovsky, M. Roseblum, and J. Kurths, “*Synchronization, A Universal Concept in Nonlinear Sciences*,” Cambridge: Cambridge University Press, 2001.
- [48] C.W. Wu, “*Synchronization and convergence of linear dynamics in random directed networks*,” IEEE transactions on Automatic control, vol. 51, pp. 1207-1210, 2006.
- [49] C.W. Wu and L.O. Chua, “*Synchronization in an array of linearly coupled dynamical systems*,” IEEE Trans. Circuits Syst., I: Fundam. Theory Appl. 43, pp. 430-447, 1995.
- [50] J.F. Heagy, L.M. Pecora, and T.L Carroll, “*Short Wavelength Bifurcation and Size Instabilities in Coupled Oscillator Systems*,” Phys. Rev. Lett., vol. 74, pp.4185 - 4188, 1995.
- [51] Z. Li and G. Chen, “*Global synchronization and asymptotic stability of complex dynamical networks*” IEEE Transactions on Circuits and Systems II, 53, pp. 28-33, 2006.
- [52] S. Boccaletti, J. Kurths, G. Osipov, D.L. Valladares, and C.S. Zhou, “*The synchronization of chaotic systems*,” Physics Reports, vol. 366, pp. 1-101, 2002.
- [53] S.V. Buldyrev, R. Parshani, G. Paul, H.E. Stanley, and S. Havlin, “*Catastrophic cascade of failures in interdependent networks*,” Nature, vol. 464, pp. 1025-1028, 2010.

- [54] S.H. Strogatz, *Nonlinear Dynamics and Chaos* (Westview Press, Boulder, 2015).
- [55] I.V. Belykh, V.N. Belykh, K.V. Nevidin, and M. Hasler, “*Persistent clusters in lattices of coupled nonidentical chaotic systems*,” *CHAOS*, vol. 13, p.165-178, 2003.
- [56] I. Belykh, D. Carter Jr., and R. Jetter, “*Synchronization in mixed networks: when good links go bad*,” *Phys. Rev. Lett.* (submitted).
- [57] I. Belykh, D. Carter Jr., and R. Jetter, “*Mixed connection graph method for synchronization in hyper-networks*,” (to be submitted to *SIAM J. Applied Dynamical Systems*).

## Appendix A

### SYNCHRONIZATION CONDITION FOR TWO $X, Y$ -COUPLED LORENZ SYSTEMS

In this appendix, we show how to derive  $a_x$ ,  $a_y$ ,  $\omega_x$ , and  $\omega_y$  for a two-node network (2.1) of  $x, y$ -coupled Lorenz systems [42]. In this case the general model (2.1) turns into

$$\begin{cases} \dot{x}_i = \sigma(y_i - x_i) + c(x_j - x_i), \\ \dot{y}_i = rx_i - y_i - x_iz_i + d(y_j - y_i) \\ \dot{z}_i = -bz_i + x_iz_i, \quad i, j = 1, 2, \end{cases} \quad (\text{A.1})$$

where  $c$  and  $d$  are uniform  $x$  and  $y$  coupling strengths, respectively. In relation to synchronization in this network, the set of thresholds  $\omega_x$ , and  $\omega_y$  is a set of double coupling strengths ( $c^*$ ,  $d^*$ ) that guarantees global stability of synchronization.

The bound on  $a_x$  for  $x$ -coupled Lorenz systems was derived in the original paper [42] which introduced the Connection Graph Method. The extension to  $x, y$ -coupled Lorenz systems is straightforward; yet, we find it instructive to show some intermediate steps before presenting the final bounds.

It follows from [42] that the trajectories of the coupled Lorenz systems are bounded by

$$|\varphi_i| < b(r + \sigma)/2\sqrt{b - 1}, \quad \varphi = x_i, y_i, (z_i - r - \sigma), \quad i, j = 1, 2. \quad (\text{A.2})$$

Similar to the proof given in Chapter 2, we construct the auxiliary system for the difference variables  $X_{ij} = x_j - x_i$ ,  $Y_{ij} = y_j - y_i$ , and  $Z_{ij} = z_j - z_i$  of (A.1):

$$\begin{aligned} \dot{X}_{ij} &= \sigma(Y_{ij} - X_{ij}) - \omega_x X_{ij} \\ \dot{Y}_{ij} &= \left( r - U_{ij}^{(z)} \right) X_{ij} - Y_{ij} - U_{ij}^{(x)} Z_{ij} - \omega_y Y_{ij} \\ \dot{Z}_{ij} &= U_{ij}^{(y)} X_{ij} + U_{ij}^{(x)} Y_{ij} - bZ_{ij}, \quad i, j = 1, 2, \end{aligned} \quad (\text{A.3})$$

where  $U_{ij}^{(\xi)} = (\xi_i + \xi_j)/2$  for  $\xi = x, y, z$  are the corresponding sum variables. Notice new terms  $-\omega_x X_{ij}$  and  $-\omega_y Y_{ij}$  that account for the coupling terms (A.1), such that  $\omega_x = 2c$  and  $\omega_y = 2d$ .

In system (A.3), we were able to eliminate the cross terms due to

$$\xi_j \eta_j - \xi_i \eta_i = U^{(\eta)}(\xi_j - \xi_i) + U^{(\xi)}(\eta_j - \eta_i).$$

As in the proof of Theorem 2.1 in Chapter 2, we consider the Lyapunov functions

$$W_{ij} = X_{ij}^2/2 + Y_{ij}^2/2 + Z_{ij}^2/2, \quad i, j = 1, 2. \quad (\text{A.4})$$

Their derivatives with respect to system (A.3) are

$$\dot{W}_{ij} = -[(\omega_x + \sigma)X_{ij}^2 + (U^{(z)} - r - \sigma)X_{ij}Y_{ij} + \omega_y Y_{ij}^2 - U^{(y)}X_{ij}Z_{ij} + bZ_{ij}^2]. \quad (\text{A.5})$$

To show negative definiteness of (A.5), we apply the Silvester criterion such that  $\omega_x + \sigma > 0$ ,  $\omega_y > 0$ ,

$$\left| \begin{array}{cc} \omega_x + \sigma & \frac{U^{(z)} - r - \sigma}{2} \\ \frac{U^{(z)} - r - \sigma}{2} & \omega_y \end{array} \right| > 0, \quad \text{and} \quad \left| \begin{array}{ccc} \omega_x + \sigma & \frac{U^{(z)} - r - \sigma}{2} & -\frac{U^{(y)}}{2} \\ \frac{U^{(z)} - r - \sigma}{2} & \omega_y & 0 \\ -\frac{U^{(y)}}{2} & 0 & b \end{array} \right| > 0. \quad (\text{A.6})$$

Therefore, we have

$$(\omega_x + \sigma)\omega_y - \frac{(U^{(z)} - r - \sigma)^2}{4} > 0 \quad (\text{A.7})$$

and

$$b(\omega_x + \sigma)\omega_y - \frac{(U^{(z)} - r - \sigma)^2 b}{4} - \omega_y \frac{(U^{(y)})^2}{4} > 0 \quad (\text{A.8})$$

Applying the estimate (A.2) to bound coordinates  $U^{(y)}$  and  $U^{(z)}$ , we obtain

$$\begin{aligned} (\omega_x + \sigma)\omega_y &> \frac{(b(r+\sigma)/2\sqrt{b-1}-r)^2}{4} \\ b(\omega_x + \sigma)\omega_y &> \frac{b(b(r+\sigma)/2\sqrt{b-1}-r)^2}{4} + \omega_y \frac{(b(r+\sigma)/2\sqrt{b-1})^2}{4} \end{aligned} \quad (\text{A.9})$$

These conditions give explicit bounds on  $\omega_x$  and  $\omega_y$ . Given the parameters of the individual Lorenz oscillator  $r, \sigma$ , and  $b$ ,  $\omega_x$  and  $\omega_y$  can be directly calculated via (A.5).

Notice that conditions (A.5) yield for  $x$ - coupled network with  $\omega_x = a_x$  and  $\omega_y = 0$  the bound:

$$a_x > \frac{b(b+1)(r+\sigma)^2}{16(b-1)} - \sigma \quad (\text{A.10})$$

for  $y$ - coupled network with  $\omega_x = 0$  and  $\omega_y = a_y$  the bound:

$$a_y > (b(r + \sigma)/2\sqrt{b-1} - r)^2 / (\sigma - (r + \sigma)/2\sqrt{b-1})^2. \quad (\text{A.11})$$

Combining bounds (A.9), (A.10), and (A.11) gives the set  $a_x, a_y, \omega_x$ , and  $\omega_y$  to be used in the condition of Theorem 2.1 in Chapter 2.

## Appendix B

### COMPUTING THE SHORTEST PATH

Computing  $b_k^{x\text{-graph}}$ ,  $b_k^{y\text{-graph}}$ , and  $b_k^{\text{mixed}}$  requires computing all of the paths from a source oscillator to a target oscillator that pass through edge  $k$ . This amounts to finding an optimal, algorithmic way to traverse a graph, which is a well-studied topic in both Graph Theory and Computer Science. Many algorithms, of varying degrees of complexity and utility, that find the shortest path between two oscillators have been created. Here we present one of the most well known and straight forward shortest path algorithms, Dijkstra’s Algorithm [17].

Dijkstra’s Algorithm is used to find the shortest path from an initial, “source” oscillator to another, “target” oscillator in the graph. The algorithm is:

1. For the first step, one assigns a distance to each oscillator. The initial distance is set to  $\infty$  for every oscillator except for the source oscillator, which has a distance of 0. These distances are updated throughout the algorithm and ensure that the shortest path is attained.
2. The graph is then partitioned into three sets: the current oscillator, the oscillators that have not been visited, and the oscillators that have been visited. For the first iteration of the algorithm, the current oscillator is the source, the visited set is empty, and the unvisited set contains all oscillators except for the source oscillator.
3. Compute the distance to all of the neighbors of the current oscillator that are in the set of unvisited oscillators. Let the current oscillator be oscillator  $i$ , a neighbor in the unvisited set be oscillator  $k$ , and the distance to oscillator  $k$  be  $d_k$ . Compute the distance  $d_k = \min \{d_k, d_i + w(i, j)\}$ ,  $d_k$  is either its current value, or the distance to oscillator  $i$  plus the distance from oscillator  $i$  to oscillator  $j$ , given by the weight of the edge from oscillator  $i$  to oscillator  $j$  ( $w(i, j)$ ), whichever is smaller.  $d_i + w(i, j)$  gives the distance from the source oscillator to oscillator  $k$  passing *through* oscillator  $i$ .

Whenever  $d_k$  is updated to  $d_i + w(i, j)$ , define the *previous oscillator*,  $P$  for oscillator  $k$  to be  $P(k) = i$ .  $P(k)$  indicates the last oscillator traveled through before  $k$  in the path for which  $d_k$  is obtained. As  $d_k$  has been updated to  $d_k = d_i + w(i, j)$ ,  $i$  is the previous oscillator from oscillator  $k$ . This is because the shortest path from the source oscillator to oscillator  $k$  passes through oscillator  $i$  immediately before reaching oscillator  $k$ .

Finding  $P(k)$  for each visited edge allows for the shortest path to oscillator  $k$  to be tracked, not just the distance of the shortest path.

4. After computing the distance for all of the neighbors of the current oscillator, the current oscillator is added to the set of visited oscillators, and the unvisited oscillator with the smallest distance,  $d_k$  is removed from the set of unvisited oscillators, becoming the current oscillator.

Note: The algorithm terminates under two conditions: (a) if the target oscillator is added to the set of visited oscillator, or (b) if the smallest distance in the set of unvisited oscillators is  $\infty$ . Condition (a) indicates that the algorithm has successfully found the shortest path from the source oscillator to the target oscillator.  $d_{target}$  gives the distance to the target from the source, and  $P(target)$  gives the previous oscillator from the target. The chain of  $P(k)$  values gives the reverse order of the path taken from the target oscillator to the source oscillator (which is the shortest path). Condition (b) indicates that there is not a path from the source oscillator to the target oscillator. In undirected graphs, this implies that the graph is disconnected. For directed graphs, this implies that there is no directed route to the target from the source.

*Remark 1: Consistency in the choice of shortest path is vital to the success of the Mixed Connection Graph method. The values  $b_k^{x-graph}$ ,  $b_k^{y-graph}$ , and  $b_k^{mixed}$  require that the same choice of paths is used for all three values. To this end, it is all but necessary to choose the paths algorithmically (such as using a “shortest path” algorithm like Dijkstra’s algorithm presented above).*

*Remark 2: While computing the shortest path between any two oscillators is intuitive way to arrive at  $b_k^{x-graph}$ ,  $b_k^{y-graph}$ , and  $b_k^{mixed}$ , it does not always provide the best bound in Theorem (3.8). Occasionally, the bound can be reduced by considering longer paths that prevent any one edge from becoming a “bottleneck” edge, an edge where most paths pass through, which causes  $b_k^{x-graph}$ ,  $b_k^{y-graph}$ , or  $b_k^{mixed}$  to be large.*

## Appendix C

### MATLAB CODE FOR SYNCHRONIZATION THRESHOLD

```

tInitial = 0;
tStep = .01;
tFinal = 2000;
global Xcoup Ycoup epsilon_star
filename = 'csvlist_X.csv';
Xcoup = csvread(filename); %This is your connectivity matrix for
the x coupling
filename2 = 'csvlist_Y.csv';
Ycoup = csvread(filename2); %This is your connectivity matrix for
the y coupling
n = length(Xcoup); % number of oscillators
initialConditions=rand(3*n,1);

LE = 0; % Left endpoint of the interval
RE = 100; % Right endpoint of the interval
MP = (LE+RE)/2; % Midpoint of the interval
tol = 0.001; % Allowed tolerance for synchrony
delta =.001;

while abs(RE-LE)>tol % To determine whether tolerance level for
    % epsilon_star is met
    epsilon_star=MP; % Critical epsilon value we seek as the interval
shrinks per each iteration

```

```

disp(epsilon_star);
[time,solutions] = ode45('modified_netmixedlorenzODE45',
,[tInitial:tStep:tFinal],initialConditions);%Integrate the
lorenz system across the new interval
disp(['Done with integration'])
for k = 1 : n-1
    synchrony_error_vector=
    sqrt(((solutions(:,k+1)-solutions(:,k)).^2+
+(solutions(:,k+n+1)-solutions(:,k+n)).^2)+
+(solutions(:,k+2*n+1)-solutions(:,k+2*n)).^2);
    % L-2 Norm for accuracy of synchrony error
end
L = length(synchrony_error_vector);
synchrony_error = sum(synchrony_error_vector((L-100):L))/100;
% L-2 Norm for accuracy of synchrony error

if synchrony_error<delta % First bound to assist us in finding
    where synchrony starts
RE = MP; % The midpoint gets re-assigned as the right endpoint
if the first bound is false
    else % Second bound to assist us in finding where synchrony
        starts
        LE = MP; % The midpoint gets re-assigned as the left
        endpoint if the second bound is false
    end
MP = (LE+RE) / 2; % The midpoint gets re-calculated since the
interval has shrunk
end
end

```

```

RE = epsilon_star; % The critical value for epsilon_star will now
be the right endpoint
disp(RE)
%plot(epsilon_star , synchrony_error , 'r')
%view(0,90)
%xlabel('epsilon_star')
%ylabel('synchrony_error')

function dU = modified_netmixedlorenzODE45(t,U)
global epsilon_star;
global Xcoup Ycoup;

n = length(U)/3;
X = reshape(U,[n 3]);
sigma = 10;
r = 28;
b = 8/3;

dX(:,1) = sigma.*(X(:,2)-X(:,1))+epsilon_star*Xcoup*X(:,1);
dX(:,2) = r.*X(:,1) - X(:,2) - X(:,1).* X(:,3)
+epsilon_star*Ycoup*X(:,2);
dX(:,3) = -b.*X(:,3) + X(:,1).*X(:,2);

dU = reshape(dX, [n*3 1]);

Xcoup=[-1 1 0 0;
        1 -2 1 0;
        0 1 -1 0 0;

```

$0 \ 0 \ 0 \ 0 \ 0]$  $Y_{\text{coup}}=[0 \ 0 \ 0 \ 0;$  $0 \ 0 \ 0 \ 0;$  $0 \ 0 \ -1 \ 1;$  $0 \ 0 \ 1 \ -1]$

## Appendix D

### MATLAB CODE FOR MIXED BK

```

xGraph = [9 12; 9 14; 11 12; 11 13; 10 13; 13 14; 14 15;
          15 16; 15 17; 17 18; 17 19; 19 20;];
% x-graph for 20 node network

yGraph = [1 2; 2 6; 5 6; 3 5; 4 5; 6 8; 7 8; 8 9; 8 13;
          8 14; 9 10; 10 12; 10 11;];
% y-graph for 20 node network

[edgeList , bkList]=bk_Mixed(xGraph,yGraph)
% Mixed bk values along with each respective adjacency lists

% To use this MATLAB code, you input the adjacency lists of
% the x-graph and y-graph into the function bk_Mixed. It
% returns the adjacency list of the mixed network, and the
% bk-mixed for the corresponding edges.
function [mixedGraphList , bkMixedList] =
    bk_Mixed(x_graph , y_graph)
    if (size(x_graph,2) ~= 2 || size(y_graph,2)~=2)
        disp('You did not input adjacency lists.')
        disp('Try again.')
    return
end

```

```

mixedGraphList = [x_graph;y_graph];
numberOfNodes = max(mixedGraphList (:));
numberOfEdges = size(mixedGraphList,1);
bkMixedList = zeros(numberOfEdges,1);
mixedGraphMatrix =
    listToMatrix(mixedGraphList, 'undirected');

for i = 1:numberOfNodes
    for j = i+1:numberOfNodes
        [cost, route] = dijkstra(mixedGraphMatrix,j,i);
        routeList = routeToList(route);
        pathHasXEdge = false;
        pathHasYEdge = false;
        isMixedPath = false;
        for k = 1:size(routeList,1)
            routeEdge = routeList(k,:);
            if(containsEdge(x_graph,routeEdge))
                pathHasXEdge = true;
            else
                if(containsEdge(y_graph,routeEdge))
                    pathHasYEdge = true;
                end
            end
        end
        if(pathHasXEdge && pathHasYEdge)
            break
        end
    end
end
end

```

```

    if (pathHasXEdge && pathHasYEdge)
        isMixedPath = true;
    end
    if (isMixedPath)
        for k = 1:size(routeList,1)
            routeEdge = routeList(k,:);
            index =
                edgeIndex(mixedGraphList, routeEdge);
            bkMixedList(index) =
                bkMixedList(index)+cost;
        end
    end
end
end
end
end

```

```

function anEdge = containsEdge(adjacencyList, edge)
    if (find(ismember(adjacencyList, edge, 'rows'), 1) > 0)
        anEdge = true;
    else
        anEdge = false;
    end
end
end

```

```

function adjacencyMatrix=
listToMatrix(adjacencyList, directedOrUndirected)
    directed = false;
    if (strcmp(directedOrUndirected, 'directed'))

```

```

        directed = true;
elseif(strcmp(directedOrUndirected, 'undirected'))
    directed = false;
else
disp('You did not correctly specify if the graph is directed or
        undirected ');
    return
end

numberOfNodes = max(adjacencyList (:));
numberOfEdges = size(adjacencyList,1);
adjacencyMatrix = zeros(numberOfNodes,numberOfNodes);
for i = 1:numberOfEdges
adjacencyMatrix(adjacencyList(i,1),adjacencyList(i,2)) = 1;
    if not(directed)
adjacencyMatrix(adjacencyList(i,2),adjacencyList(i,1)) = 1;
    end
end
end

function routeList = routeToList(route)
    routeList = zeros(length(route)-1,2);
    for i = 2:length(route)
        if(route(i-1) < route(i))
            routeList(i-1,1) = route(i-1);
            routeList(i-1,2) = route(i);
        else
            routeList(i-1,1) = route(i);

```

```

        routeList(i-1,2) = route(i-1);
    end
end
end

function index = edgeIndex(edgeList,edge)
    index = find(ismember(edgeList,edge,'rows'));
end

function [e L] = dijkstra(A,s,d)

if s==d
    e=0;
    L=[s];
else

A = setupgraph(A,inf,1);

if d==1
    d=s;
end
A=exchangennode(A,1,s);

lengthA=size(A,1);
W=zeros(lengthA);
for i=2 : lengthA
    W(1,i)=i;
    W(2,i)=A(1,i);
end
end
end

```

```

end

for i=1 : lengthA
    D(i,1)=A(1,i);
    D(i,2)=i;
end

D2=D(2:length(D),:);
L=2;
while L<=(size(W,1)-1)
    L=L+1;
    D2=sortrows(D2,1);
    k=D2(1,2);
    W(L,1)=k;
    D2(1,:)=[];
    for i=1 : size(D2,1)
        if D(D2(i,2),1) > (D(k,1)+A(k,D2(i,2)))
            D(D2(i,2),1) = D(k,1)+A(k,D2(i,2));
            D2(i,1) = D(D2(i,2),1);
        end
    end
end

for i=2 : length(A)
    W(L,i)=D(i,1);
end

end

if d==s
    L=[1];

```

```

else
    L=[d];
end
e=W(size(W,1),d);
L = listdijkstra(L,W,s,d);
end

function G = exchangenode(G,a,b)

%Exchange element at column a with element at column b;
buffer=G(:,a);
G(:,a)=G(:,b);
G(:,b)=buffer;

%Exchange element at row a with element at row b;
buffer=G(a,:);
G(a,:)=G(b,:);
G(b,:)=buffer;

function L = listdijkstra(L,W,s,d)

index=size(W,1);
while index>0
    if W(2,d)~=W(size(W,1),d)
        L=[L s];
        index=0;
    else
        index2=size(W,1);
        while index2>0

```

```

        if W(index2 ,d)<W(index2 -1,d)
            L=[L W(index2 , 1)];
            L=listdijkstra(L,W,s,W(index2 , 1));
            index2=0;
        else
            index2=index2 -1;
        end
        index=0;
    end
end
end

function G = setupgraph(G,b,s)

if s==1
    for i=1 : size(G,1)
        for j=1 : size(G,1)
            if G(i ,j)==0
                G(i ,j)=b;
            end
        end
    end
end

end

if s==2
    for i=1 : size(G,1)
        for j=1 : size(G,1)
            if G(i ,j)==b
                G(i ,j)=0;
            end
        end
    end
end

```

end

end

end

end

## Appendix E

## PYTHON CODE FOR UNIFORM BK

```

import sys , networkx
import numpy as np
fA=open(sys.argv[1])
fB=open(sys.argv[2])
N=0
for line in fA:
    k=[int(i) for i in line.split(" ")]
    N=max([N]+k)
for line in fB:
    k=[int(i) for i in line.split(" ")]
    N=max([N]+k)
fA.seek(0)
fB.seek(0)
A=np.zeros((N,N))
B=np.zeros((N,N))
edgesA=[]
edgesB=[]
for line in fA:
    k=[int(i)-1 for i in line.split(" ")]
    A[k[0],k[1]]=1
    A[k[1],k[0]]=1
    edgesA.append(k)
for line in fB:

```

```

k=[int(i)-1 for i in line.split(" ")]
B[k[0],k[1]]=1
B[k[1],k[0]]=1
edgesB.append(k)
def stability(A,B):
    g=networkx.from_numpy_matrix(np.maximum(A,B))
    print "\t v1 v2 bk bk_mixed"
    for i1 in xrange(A.shape[0]):
        for j1 in xrange(i1):
            k=[i1, j1]
if A[k[0],k[1]] ==0 and B[k[0],k[1]]==0:
    continue
bk=0
bkmixed=0
shortest_paths=[]
N=A.shape[0]
    for i in xrange(N):
        for j in xrange(i):
            try:
                a=[i]
shortest_paths.append([q for q in networkx.shortest_path(g,i,j)])
            except:
                continue
        #print shortest_paths
for p in shortest_paths:
    unmixed=True
    usesk=False
    l=0

```

```

for i in xrange(1, len(p)):
    q=[p[i-1], p[i]]
    if (q[0]==k[0] and q[1]==k[1]) or (q[0]==k[1] and
                                         q[1]==k[0]):
        usesk=True
        if A[q[0], q[1]]!=A[p[0], p[1]] and
        B[q[0], q[1]]!=B[p[0], p[1]]:
            unmixed=False
            if usesk:
                bk+=len(p)-1
                if not unmixed:
                    bkmixed+=len(p)-1
                    print "\t", k[0], k[1], bk, bkmixed
                    stability(A, B)
                    print "----cut----"
                    for k in edgesA:
                        Anew=np.ones((N, N))
                        Anew[k[0], k[1]]=0
                        Anew[k[1], k[0]]=0
                        Bnew=np.zeros((N, N))
                        Anew=np.minimum(Anew, A)
                        Bnew[k[0], k[1]]=1
                        Bnew[k[1], k[0]]=1
                        Bnew=np.maximum(Bnew, B)
                    print "switched edge ", k[0], ", ", k[1]
                    stability(Anew, Bnew)
                    for k in edgesB:
                        Bnew=np.ones((N, N))

```

```
Bnew[k[0],k[1]]=0
Bnew[k[1],k[0]]=0
Anew=np.zeros((N,N))
Bnew=np.minimum(Bnew,B)
Anew[k[0],k[1]]=1
Anew[k[1],k[0]]=1
Anew=np.maximum(Anew,A)
print "switched edge ", k[0], ", ", k[1]
stability(Anew,Bnew)
```