KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

GRADUATE SCHOOL
Department of Mechanical Engineering

# A study of partial synchronization in networks of delay-coupled systems

**Libo Su**

Supervisors:
Prof. dr. ir. Wim Michiels
Prof. dr. Henk Nijmeijer
Co-supervisor:
Dr. ir. Erik Steur

Dissertation presented in partial fulfillment of the requirements for the joint degree of Doctor of Engineering Science (PhD): Computer Science at KU Leuven, and Doctor (PhD): Mechanical Engineering at Eindhoven University of Technology.

May 2021

# A study of partial synchronization in networks of delay-coupled systems

**Libo SU**

Examination committee:
Prof. dr. ir. Bob Puers
(chair of preliminary defense)
Prof. dr. Philip de Goey
(chair of public defense)
Prof. dr. ir. Wim Michiels, supervisor
Prof. dr. Henk Nijmeijer, supervisor
Dr. ir. Erik Steur, co-supervisor
Prof. dr. ir. Johan Suykens
Prof. dr. ir. Karl Meerbergen
Prof. dr. Jean-Pierre Richard
Prof. dr. Remco van der Hofstad

Dissertation presented in partial fulfillment of the requirements for the joint degree of Doctor of Engineering Science (PhD): Computer Science at KU Leuven, and Doctor (PhD): Mechanical Engineering at Eindhoven University of Technology (TU/e).

May 2021



This project has received funding from the European Union's EU Framework Programme for Research and Innovation Horizon 2020 under Grant Agreement No 675080.

© 2021 KU Leuven – Faculty of Engineering Science
Uitgegeven in eigen beheer, Libo Su, Celestijnenlaan 200A box 2402, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN 978-90-386-5266-5

# A study of partial synchronization in networks of delay-coupled systems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op dinsdag 11 mei 2021 om 16:00 uur

door

Libo Su

geboren te Zhejiang, China

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

| | |
|---|---|
| voorzitter: | prof.dr. L.P.H. de Goey |
| 1e promotor: | prof.dr. H. Nijmeijer |
| 2e promotor: | prof.dr.ir. W. Michiels (KU Leuven) |
| copromotor(en): | dr.ir. E. Steur |
| leden: | prof.dr. R.W. van der Hofstad |
| | prof.dr. J.P. Richard (Centrale Lille) |
| | prof.dr.ir. J.A.K. Suykens (KU Leuven) |
| | prof.dr.ir. K. Meerbergen (KU Leuven) |

*Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.*

# Preface

This thesis is the result of my four-year doctoral study at KU Leuven and TU/e, and it would not have been possible without the help, support and guidance that I received from many people.

First and foremost, I want to thank my supervisors, Wim Michiels, Henk Nijmeijer and Erik Steur for their valuable guidance and constant support. I feel extremely fortunate and privileged to be guided by them during my PhD. Wim, thank you for offering me this PhD position. Your broad knowledge and enthusiasm for research are really admiring, and I am grateful for the effort and time you spent answering my questions and proofreading my papers (even at weekends). Henk, thank you for being my supervisor again after my master study at the D&C group. I have always admired your eagle-eye view and rigorous attitude, and also have been grateful for your support and patience. Erik, thank you for agreeing to be my co-supervisor. Your sharp mathematical skills and theoretical insight have greatly helped me improve the quality of my work, and I also want to thank your help with the experimental setup. In addition, I would also like to thank the European Union for funding this research.

I would like to thank the members of my examination committee, Johan Suykens, Karl Meerbergen, Jean-Pierre Richard, Remco van der Hofstadfor for accepting the roles, for carefully reading my thesis, and for all the questions, discussions and suggestions. I thank Bob Puers for chairing the preliminary defense and Philip de Goey for chairing the public defense.

During my PhD, I had the chance to visit and collaborate with the Integrated Vehicle Safety group at TNO, Helmond, the Netherlands. This would not be possible without the support and help from Mauro Fusco, Dehlia Willemson and Jan Verhaegh. I would like to thank Mauro and Dehlia for introducing me to the group and for their insightful feedback on our collaborative work. I would also like to thank Jan for the helpful discussion on model predictive control for vehicle platooning. This PhD project also allowed me to spend one month in the Control and Scientific Computing group of the CNRS lab CRIStAL at Centrale Lille, France. I would like to thank Jean-Pierre Richard

and Laurentiu Hetel for arranging this visit.

I would like to express my gratitude to my fellow early stage researchers in the UCoCoS project: Kirill, Haik, Quentin, Deesh and Jijju. It is so lucky of me to be in the same project with you guys. I enjoyed every gathering we had during the past few years. Jijju, thank you for being a great office mate. It was a pity that we only shared our office for one year due to the Corona pandemic. Besides, I am grateful to the colleagues whom I met during my stay at KU Leuven, Alexey, Dan, Francesco, Luca, Pieter, Koen, Marco, Yanling, Simon, Bruno, Pieterjan, Yuya, Shurong, Zhiyu, Fan, and whom I met during my stay at TU/e, Andy, Fahin, Wouter, Carlos, Haitao, Yuzhe, Veronika, Ksenia and Daniël. Thanks for sharing your expertise and experiences during lunch and coffee breaks. A special acknowledgment to Yanling for our collaboration on synchronization. Another special thank goes to Luca for helping me with my driving training. I would also like to extend my gratitude to the staff of the Computer Science Department at KU Leuven (especially Margot and Stella) and the Mechanical Engineering Department at TU/e (especially Geertje) for their administrative support. In addition, I would also like to thank my dear Chinese friends in Leuven, Genghua, Kaili, Chenmin, Shuang, Liqian, Boyuan and Kexin, with whom I shared so many memorable trips and dinners. Without them, my stay in Belgium would not be so enjoyable.

Finally, I would like to dedicate this thesis to my family, especially to my grandmother who passed away recently, my parents, my sister and my wife. Dear Ruoyun, my beloved wife, thank you for standing by me all these years. I could not be able to finish this work without your support and love.

*Libo Su*
*Düsseldorf*
*April 2021*

# Summary

This thesis studies partial synchronization in networks of delay-coupled systems. *Partial synchronization* refers to the phenomenon that coupled systems can be grouped into clusters such that synchrony is only observed within each cluster. The coupling considered here contains a time-delay. A pattern of partial synchronization can be characterized by a *partial synchronization manifold* which is a linear invariant subspace of the state space of the network dynamics, corresponding to partially synchronous motion. The existence and asymptotic stability of such a manifold are required to observe partial synchronization in simulations and experiments. The latter guarantees that the manifold attracts the neighboring solutions. Therefore, the partial synchronization problem can be tackled in two steps: 1) identify partial synchronization manifolds; 2) analyze the stability of the partial synchronization manifolds. This thesis aims to provide an efficient method to characterize all partial synchronization manifolds and tractable stability conditions of partial synchronization manifolds. In return, the method and stability conditions provide insights into the relation between the dynamics of individual systems, coupling parameters, time-delays, and partial synchronization.

First, a method is proposed to identify all partial synchronization manifolds in delay-coupled networks, using the information on network structure and coupling type. The method relies on a recently proposed existence criterion for partial synchronization manifolds. The criterion is expressed as a row-sum check on the ordered adjacency matrix (which represents the network structure). In this thesis, the criterion is extended so that the method is also applicable to networks with different systems, and the computational efficiency is significantly improved for the case where only the transmitted signals in coupling contain a time-delay. In addition, the method allows to automatically decompose the network dynamics into synchronization error dynamics describing the deviations of the system states within the clusters, and dynamics on manifolds describing the network dynamics after synchronization. The method is implemented as a publicly available software tool.

Second, the stability of the partial synchronization manifolds is analyzed by using the second method of Lyapunov. The stability analysis of partial synchronization

manifolds is recast as the stability analysis of an equilibrium (or zero solution) of the synchronization error dynamics. It is shown that the linearized error system can be interpreted as a linear parameter-varying (LPV) time-delay system. By assessing the stability of this LPV time-delay system using a Lyapunov-Krasovskii functional (LKF), the sufficient conditions for the local stability of partial synchronization manifolds are derived and expressed in the form of linear matrix inequalities (LMIs), whose feasibility can be assessed by using standard numerical tools.

Third, the partial synchronization in a practical sense is considered. From an application perspective, the conditions for partial synchronization are too stringent, as inevitable perturbations or uncertainties (considered as additive to the nominal setting) destroy the zero equilibrium of the error system. Consequently, exact partial synchronization, where the states of the systems within each cluster are perfectly synchronized, cannot be achieved. Instead, an approximate synchronization, *practical partial synchronization*, may be observed, where the states of the systems within each cluster converge to each other up to some tolerance, and this tolerance tends to zero if (the size of) the perturbations tend to zero. In this case, the error system in the analysis is viewed as a non-autonomous time-delay system with a bounded additive perturbation. By assessing the practical stability of this error system using LKFs, the conditions for practical partial synchronization is derived and again formulated in term of LMIs. Besides, an explicit relation between the size of perturbation and the bound of the synchronization error is provided. All these conditions (for both exact and practical synchronization) induce restrictions on the coupling strength and time-delay. They can be used to check whether for a given set of parameters partial synchronization occurs and shed a light on the parameter dependence.

Fourth, experiments on a network of electronic neurons are performed to demonstrate the obtained theoretical results. Each electronic neuron is the implementation of a neuron model in a circuit board. A cross correlation-based notion of practical synchronization is introduced to quantify the synchronization between the neurons. The experiments confirm the theory well though the theoretical results tend to be conservative. The synchronization regions in network parameter space determined by the theoretical conditions are a qualitative match of those obtained via experiments, although the former are smaller than the latter due to the conservatism of the theory.

Finally, a control design is presented to achieve vehicle platooning which is a real-life application of controlled network synchronization. The design, grounded in model predictive control (MPC), employs a cooperative adaptive cruise control (CACC) strategy to achieve vehicle platooning. The CACC strategy uses vehicle-to-vehicle communication to allow for inter-vehicle data exchange such that short inter-vehicle distances can be reached while guaranteeing safety. To improve the cohesion of the platoon when encountering disturbance, a bi-directional network topology is used. It is shown that compared to the commonly used unidirectional topology, the bi-directional topology allows the vehicles to maintain shorter distances.

# Samenvatting

Dit proefschrift bestudeert partiële synchronisatie in netwerken van systemen met tijdsvertragingen in de koppeling. Partiële synchronisatie betreft het fenomeen waarbij gekoppelde systemen in clusters kunnen worden ingedeeld, waarbij binnen elke cluster synchroon gedrag wordt waargenomen. Een patroon van partiële synchronisatie wordt gekenmerkt door een zogenaamde variëteit van partiële synchronisatie. Dit is een deelruimte van de toestandsruimte van het gehele netwerk, die invariant is met betrekking tot de systeemdynamica en waarin de trajecten gedeeltelijk synchroon zijn. Het bestaan, alsook de asymptotische stabiliteit van zulke variëteit zijn vereist voor het waarnemen van partiële synchronisatie in simulatie en experiment, wat leidt tot een twee stappen procedure voor de analyse van partiële synchronisatie. Dit proefschrift heeft als doelstelling om efficiënte methodes aan te reiken om alle variëteiten van partiële synchronisatie te berekenen, en om de stabiliteit ervan na te gaan. De stabiliteitsvoorwaarden leveren inzicht in de relatie tussen de dynamica van individuele systemen en parameters van de koppeling, en het optreden van partiële synchronisatie.

Vooreerst wordt een methode voorgesteld om alle variëteiten van partiële synchronisatie te bepalen op basis van de netwerkstructuur en het soort koppeling. De methode is gebaseerd op een bestaanscriterium voor variëteiten van partiële synchronisatie, uitgedrukt als rijsomtest op de geordende verbindingsmatrix van het netwerk. In dit proefschrift wordt het criterium uitgebreid zodat het ook toepast kan worden op netwerken van verschillende systemen. Tegelijkertijd wordt de rekenkost gereduceerd wanneer enkel de verzonden signalen tussen de (deel)systemen vertraagd zijn. Bovendien is het mogelijk om de netwerkdynamica automatisch op te splitsen in de foutendynamica, bepaald door onderlinge verschillen tussen de toestanden binnen de clusters, en de dynamica op de variëteit, die het netwerkgedrag na synchronisatie beschrijft. De methode is geïmplementeerd in vrij beschikbare software.

Als tweede bijdrage wordt de stabiliteit van variëteiten van partiële synchronisatie geanalyseerd met behulp van de tweede methode van Lyapunov. De stabiliteitsanalyse van zo'n variëteit kan herleid worden tot de stabiliteitsanalyse van de evenwichtsop-

lossing van de foutendynamica, namelijk de nul-oplossing. Er wordt aangetoond dat de linearisatie geïnterpreteerd kan worden als een lineair parameter-variërend systeem met vertragingen. Door de stabiliteit van dit systeem te onderzoeken met behulp van een Lyapunov-Krasovskii functionaal (LKF), worden voorwaarden voor de lokale stabiliteit van variëteiten van partiële synchronisatie afgeleid, en uitgedrukt in de vorm van standaard lineaire matrixongelijkheden.

Ten derde wordt partiële synchronisatie in een meer praktische setting behandeld. Vanuit toepassingsperspectief zijn de bestaansvoorwaarden voor partiële synchronisatie te streng, aangezien onvermijdelijke verstoringen of modelleringsfouten het nul-evenwicht van het foutensysteem ongedaan maken. Bijgevolg kan exacte partiële synchronisatie niet bereikt worden maar kan een zwakkere notie van synchronisatie, die we praktische partiële synchronisatie nomen, worden waargenomen. Hierbij convergeren de systemen binnen elk cluster op een tolerantie na, waarbij deze tolerantie naar nul gaat als de grootte van de verstoringen naar nul gaat. In de analyse kan de foutendynamica dan geïnterpreteerd worden in termen van een niet-autonoom systeem met een begrensde additieve verstoring. Door de zgn. praktische stabiliteit van dit foutensysteem te beoordelen met behulp van LKFs, worden voorwaarden voor praktische partiële synchronisatie afgeleid, alsook een expliciete relatie tussen de grenzen op de verstoringen en de synchronisatiefouten. De voorwaarden, voor zowel exacte als praktische synchronisatie kunnen worden gebruikt om te voorspellen of partiële synchronisatie plaats vindt, en ze werpen een licht op de parameterafhankelijkheid.

Als vierde bijdrage worden experimenten uitgevoerd op een netwerk van elektronische neuronen om de theoretische resultaten te staven. Elk elektronisch neuron komt overeen met de implementatie van een neuronmodel op een printplaat. Om de synchronisatie te kwantificeren wordt een criterium op basis van de kruiscorrelatie tussen opgemeten tijdsreeksen gebruikt. De gebieden in de parameterruimte overeenkomstig synchronisatie, die worden bepaald door de theorie en de experimenten, komen kwalitatief overeen. In het eerste geval zijn ze kleiner, doordat de theoretisch analyse leidt tot voldoende maar niet nodige voorwaarden voor synchronisatie.

Tenslotte wordt een regelaarsontwerp voorgesteld om een lineaire formatie van voertuigen te stabiliseren, een toepassing van gecontroleerde netwerksynchronisatie. Het ontwerp is gebaseerd op een coöperatieve adaptieve cruise control-strategie, die gebruikt maakt van communicatie tussen de voertuigen. Om de cohesie van de groep na een verstoring te herstellen, wordt een bidirectionele netwerktopologie gebruikt. die, in vergelijking met de algemeen gebruikte unidirectionele topologie, de voertuigen in staat stelt om kortere tussenafstanden aan te houden zonder aan veiligheid in te boeten.

# List of Abbreviations

**CACC** Cooperative Adaptive Cruise Control. 8, 107–109, 111, 113, 114, 118–120, 122

**LKF** Lyapunov-Krasovskii Functional. 14

**LMI** Linear Matrix Inequality. 7, 8, 15, 16, 47, 48, 53, 54, 56, 57, 60, 64, 75, 76, 79–82

**LPV** Linear Parameter Varying. 48, 53, 54, 56, 60

**MPC** Model Predictive Control. 8, 109, 111, 113, 114, 116–120, 122, 159, 160

**NCS** Networked Control System. 128

**PID** Proportional Integral Derivative. 120

**RFDE** Retarded Functional Differential Equation. 12–14

**SCN** Suprachiasmatic Nucleus. 129

# List of Symbols

| | |
|---|---|
| $\mathbb{C}$ | Set of complex numbers |
| $\mathbb{Z}$ | Set of integers |
| $\mathbb{Z}_{\geq 0}$ | Set of non-negative integers |
| $\mathbb{R}$ | Set of real numbers |
| $\mathbb{R}_{+}$ | Set of positive real numbers |
| $\mathbb{R}_{\geq 0}$ | Set of non-negative real numbers |
| $\mathbb{R}^n$ | $n$ dimensional real vector space |
| $\ker(A)$ | Kernel of matrix $A$ |
| $\operatorname{rank}(A)$ | Rank of matrix $A$ |
| $\sigma(A)$ | Spectrum of matrix $A$ (multiset of all eigenvalues of $A$) |
| $\lambda_i(A)$ | $i$th eigenvalue of matrix $A$ |
| $\otimes$ | Kronecker product |
| $I_n$ | $n \times n$-dimensional identity matrix |
| $\mathbf{1}_n$ | $n$-dimensional vector with entries 1 |
| $\mathbf{0}_{n\times m}$ | $n \times m$-dimensional matrix with all entries 0 |
| $\operatorname{Re}(\lambda)$ | Real part of $\lambda \in \mathbb{C}$ |
| $\operatorname{Im}(\lambda)$ | Imaginary part of $\lambda \in \mathbb{C}$ |
| $\|v\|$ | Euclidean norm of vector $v \in \mathbb{R}^n$ |
| $\|A\|_2$ | 2 induced norm of matrix $A$ |

$\|A\|_\infty$ $\infty$ induced norm of matrix $A$

$\mathscr{C}([-\tau,0],\mathbb{R}^n)$ Space of continuous functions mapping the interval $[-\tau,0]$ into $\mathbb{R}^n$

$\|\phi\|_s$ Norm of $\phi \in \mathscr{C}([-\tau,0],\mathbb{R}^n)$, $\|\phi\|_s = \sup_{-\tau\leq\theta\leq 0}\|\phi(\theta)\|$

# Contents

**Summary** iii

**Samenvatting** v

**List of Abbreviations** vii

**List of Symbols** x

**Contents** xi

**List of Figures** xv

**List of Tables** xix

**1 Introduction** 1
- 1.1 The synchronization phenomenon . . . 1
  - 1.1.1 Historical notes . . . 2
  - 1.1.2 Partial synchronization . . . 3
- 1.2 Motivation and contributions . . . 5
- 1.3 Outline . . . 7

**2 Preliminaries** 9
- 2.1 Notation . . . 9
  - 2.1.1 Sets and numbers . . . 9
  - 2.1.2 Vectors, matrices and norms . . . 9
  - 2.1.3 Functions . . . 11
- 2.2 Semipassive systems . . . 11
- 2.3 Time-delay systems . . . 12
- 2.4 Linear matrix inequalities . . . 15
- 2.5 Elementary graph theory . . . 16

**3 Characterization of partial synchronization manifolds 19**
3.1 Introduction . . . 19
3.2 Characterization of partial synchronization manifolds . . . 20
3.2.1 Existence of partial synchronization manifolds . . . 20
3.3 Computation of partial synchronization manifolds . . . 27
3.4 Dynamics decomposition of partially synchronized networks . . . 32
3.4.1 Synchronization error dynamics . . . 33
3.4.2 Dynamics on the synchronization manifolds . . . 37
3.4.3 Software . . . 38
3.5 Numerical example . . . 39
3.6 Conclusions . . . 43

**4 Partial synchronization in networks of delay-coupled systems 47**
4.1 Introduction . . . 47
4.2 Networks . . . 48
4.3 Synchronization error dynamics . . . 51
4.3.1 Networks with invasive coupling . . . 51
4.3.2 Networks with non-invasive coupling . . . 52
4.4 Conditions for partial synchronization . . . 53
4.5 Numerical example . . . 56
4.6 Conclusions . . . 60

**5 Practical partial synchronization in networks of delay-coupled systems 61**
5.1 Introduction . . . 61
5.2 Networks . . . 64
5.3 Practical partial synchronization . . . 70
5.3.1 Practical partial synchronization definition . . . 70
5.3.2 Perturbed synchronization error dynamics . . . 71
5.4 Conditions for practical partial synchronization . . . 75
5.5 Numerical example . . . 79
5.6 Conclusions . . . 82

**6 Experimental synchronization of delay-coupled neurons 87**
6.1 Introduction . . . 87
6.2 Experimental setup . . . 89
6.3 Experimental synchronization . . . 91
6.3.1 Network with same type of neurons . . . 93
6.3.2 Network with different types of neurons . . . 103
6.4 Conclusions . . . 104

**7 A model predictive control approach to truck platooning 107**
7.1 Introduction . . . 107
7.2 Problem formulation . . . 110

7.2.1 Vehicle model . . . 110
7.2.2 Platoon settings . . . 111
7.2.3 Platoon cohesion . . . 113
7.3 Control design . . . 113
7.3.1 Control law design . . . 114
7.3.2 MPC design . . . 114
7.4 Simulation results . . . 118
7.5 Conclusions . . . 122

**8 Conclusions and recommendations 125**
8.1 Conclusions . . . 125
8.2 Recommendations . . . 127

**A Partial synchronization manifolds in networks with multiple weights 131**

**B Construction of the row-sum inequality indicator vector $v_S$ 135**

**C Proof of Theorem 4.4.1 139**

**D Invariant set of coupled Hindmarsh-Rose neurons 143**
D.1 Existence of invariant set . . . 144
D.2 Computation of invariant set . . . 146

**E Proof of Theorem 5.4.1 147**

**F Proof of Theorem 5.4.2 151**

**G Proof of semi-passivity of System** (5.5) **155**

**H Solution of LMIs** (5.28) **in the example of Section 5.5 157**

**I Model predictive control 159**

**Bibliography 163**

**Curriculum Vitae 175**

**List of publications 177**

# List of Figures


1.1 Drawing of two synchronized pendulum clocks by Huygens . . . . . 3

3.1 Stability regions of (3.31)-(3.32) with $\Omega = 1$ and $\varepsilon = 0.02$. . . . . 42
3.2 Simulation results of coupled linear systems (3.31), (3.32) . . . . . 44
3.3 Simulation results of coupled nonlinear systms (3.51), (3.32) . . . . . 45

4.1 Partial synchronization regions via theory and simulations . . . . . 58
4.2 The invariant box and the trajectories of coupled neurons . . . . . 59

5.1 Illustration of partial synchronization . . . . . 62
5.2 Graph of network 5.4 . . . . . 67
5.3 Simulation results of the network corresponding to Example 5.2.2, with adjacency matrix $A_n$ . . . . . 69
5.4 Simulation results of the network corresponding to Example 5.2.2, with adjacency matrix $A$ . . . . . 69
5.5 Synchronization errors of the network corresponding to Example 5.2.2, with $A$ and $A_n$ . . . . . 70
5.6 Partial synchronization region (gray area) in the $(k, \tau)$ space [1]. . . . 77
5.7 The network of Example 5.2.2 with adjacency matrix $A$ . . . . . 80
5.8 The states $x_{i,1}$ of the network with $k = 0.3, \tau = 2$ in the time interval $[0, 100]$ . . . . . 83
5.9 The states $x_{i,1}$ of the network with $k = 0.3, \tau = 2$ in the time interval $[990, 1000]$ . . . . . 84
5.10 Synchronization errors of the network with $A$ (practical synchronization, blue colored) and $A_n$ (exact synchronization, red colored) in the time interval $[0, 100]$ . . . . . 85
5.11 Synchronization errors of the network with $A$ (practical synchronization, blue colored) and $A_n$ (exact synchronization, red colored) in the time interval $[990, 1000]$ . . . . . 85
5.12 Synchronization errors with different sizes of perturbations . . . . . 86

6.1 Experimental setup of the electronic neurons . . . . . 89
6.2 Different types of neuron behaviors due to different values of the parameter $E_i$ (when $u_i \equiv 0$). . . . . 90
6.3 Synchronization regions of the network with the adjacency matrix (6.7) in the $(k,\tau)$ space via experiments, detected by using different values of $\varepsilon$: (a) $\varepsilon = 0.1$, (b) $\varepsilon = 0.01$ and (c) $\varepsilon = 0.001$. . . . . 95
6.4 Outputs of neurons 3 and 4 in the network with the adjacency matrix (6.7) obtained via experiments with: (a) $k = 0.17$ and $\tau = 2$, leading to $Co_{y_3,y_4} = 0.9$; (b) $k = 0.24$ and $\tau = 2$, leading to $Co_{y_3,y_4} = 0.99$; (c) $k = 0.3$ and $\tau = 2$, leading to $Co_{y_3,y_4} = 0.999$. . . . . 96
6.5 Synchronization regions of the network with the adjacency matrix (6.7) in the $(k,\tau)$ space via LMIs. . . . . 96
6.6 Synchronization regions of the network with the adjacency matrix (6.7) in the $(k,\tau)$ space via experiments with (a) $E_i = 2$ and (b) $E_i = 5$, detected by adopting $\varepsilon = 0.001$ as the synchronization threshold. . . . 98
6.7 Synchronization regions of the network with (6.8) in the $(k,\tau)$ space via experiments, detected by adopting $\varepsilon = 0.001$ as the synchronization threshold. . . . . 99
6.8 Behaviors of the neurons in the network with the adjacency matrix (6.8), $k = 0.5$ and $\tau = 0.16$ in simulation. . . . . 100
6.9 Behaviors of the neurons in the network with the adjacency matrix (6.8), $k = 4$ and $\tau = 0.16$ in simulation. . . . . 101
6.10 Synchronization regions of the network with the adjacency matrix (6.10) in the $(k,\tau)$ space via experiments, detected by adopting $\varepsilon = 0.001$ as the synchronization threshold. . . . . 102
6.11 Synchronization regions of the network with the adjacency matrix (6.11) in the $(k,\tau)$ space via experiments (a), simulations (b) and LMIs (c) . . . . . 103
6.12 Behaviors of the neurons in the network with the adjacency matrix (6.11), $k = 5$ and $\tau = 0.1$ in experiment . . . . . 105
6.13 Synchronization errors of the network with the adjacency matrix (6.11), $k = 5$ and $\tau = 0.1$. . . . . 106

7.1 Schematic overview of a heterogeneous vehicle platoon and the bi-directional information flow topology . . . . . 112
7.2 Behaviors of heterogeneous vehicles in platoon during acceleration . . 114
7.3 Block diagram of the CACC Controller . . . . . 115
7.4 Acceleration $a_i$, speed $v_i$, and spacing error $e_i$ response of a four-truck platoon during acceleration using MPC-based CACC. . . . . 120
7.5 Acceleration $a_i$, speed $v_i$ and spacing error $e_i$ response of a four-truck platoon during acceleration using MPC-based CACC from [2]. . . . . 121

7.6 Acceleration $a_i$, speed $v_i$ and spacing error $e_i$ response of a four-truck platoon during acceleration using MPC-based CACC with parameters in Table 7.4. . . . . 123
7.7 Acceleration $a_i$, speed $v_i$ and spacing error $e_i$ response of a four-truck platoon during acceleration using MPC-based CACC with parameters in Table 7.3 and $\varepsilon = 0.01$. . . . . 124

# List of Tables


3.1 Generating the set of all partitions of a number of systems using the original algorithm [3]. . . . . 28
3.2 Generating the set of all partitions of eight systems with row-sum restrictions. . . . . 32

6.1 Synchronization patterns and $v_c$ . . . . . 94

7.1 Values of vehicle and platoon parameters . . . . . 119
7.2 Gear ratios for different speed ranges [4] . . . . . 119
7.3 MPC Parameters used in [2] . . . . . 122
7.4 MPC Parameters after tuning . . . . . 122

# Chapter 1

# Introduction

**Abstract:** This chapter introduces the synchronization phenomenon and gives some historical notes. It continues with an introduction to partial synchronization and relevant concepts. At the end of this chapter, the motivation, main contributions and outline of this thesis are presented.

## 1.1 The synchronization phenomenon

What is synchronization? The word synchronization is derived from the word synchronous which has a Greek origin "σύγχρονος" composed of the prefix "συν" meaning the same or together and "χρονος" meaning time [30]. Literally translated, synchronous means occurring at the same time [30]. Therefore, synchronization refers to the phenomenon that "events" keep occurring simultaneously (for an extended period of time) [5]. This phenomenon has been observed in various fields, ranging from nature, science, engineering to society. Examples include: fireflies flashing synchronously [6], geese flying in flock during migration [7], vehicles moving in platoons [8], audience forming "Mexican waves" during a football match [9], *etc*. Besides its ubiquitousness, synchronization also plays a very important role in life. For example, heartbeats are regulated by clusters of synchronized pacemaker neurons [10], and blood sugar levels are maintained by the insulin secreted by the synchronized $\beta$-cells in the pancreas [11].

### 1.1.1 Historical notes

The scientific community has been investigating synchronization for centuries. The Dutch physicist, mathematician, astronomer and inventor Christiaan Huygens was probably the first scientist who made a record of the synchronization phenomenon. In 1665, he documented the synchronous motions of two pendulum clocks mounted next to each other on a wooden beam [12], see Figure 1.1. The two clocks swung with the same frequency but in opposite directions. He also pointed out accurately that it was the imperceptible motion of the beam that caused the anti-phase synchronization of the two clocks. In the middle of the eighteenth century, another important observation on synchronization was made by Lord Rayleigh in his famous book "The theory of sound". He described the synchronization of two organ pipes which sounded in unison when their outlets were close to each other [13]. With the development of electrical and radio engineering, another progress in the investigation of synchronization was made by Edward Appleton and Balthasar van der Pol. In the 1920s, they theoretically and experimentally studied the synchronization of triode electronic generators [14, 15] after the discovery of this phenomenon by W. H. Eccles and J. H. Vincent when applying for a British patent [16]. These studies were of great importance because of their applications on radio communication systems. Progress was also made in other fields. In 1967, Winfree discussed the synchronization of biological oscillators in [17]. This was one of the first works to analyze synchronization in a large population of nonlinear oscillators. The early work on synchronization focused on two coupled oscillators. Pioneered by the work of Fujisaka and Yamada [18] in 1983, researchers discovered that chaotic systems could synchronize ([19, 20, 21, 22]), which was then a surprising finding. It was previously believed that it was impossible for chaotic systems to be synchronized due to their high sensitivity to initial conditions. Later on, in 1998, Pecora and Carroll showed that the stability of the synchronous state of coupled chaotic systems could be analyzed by using a master stability function in [23], which became a popular tool for analyzing synchronization problems. Another research focus has been synchronization of systems interconnected via time-delay couplings since the 1980s. This is because that in reality, the couplings usually contain time-delays as a result of the finite information transmission and processing speeds among the coupled systems. In 1989, Schuster and Wagner investigated the synchronization (mutual entrainment) of two delay-coupled phase oscillators [24]. In [25], it was shown that the master stability function method could also be applied to analyze the synchronization of delay-coupled chaotic systems. More results on synchronization of delay-coupled systems can be found in [26, 27, 28, 29] and the references therein. Inspired by these seminal works, synchronization has become a focus of study for researchers from various disciplines including physics, biology, mathematics, engineering, and even social science.

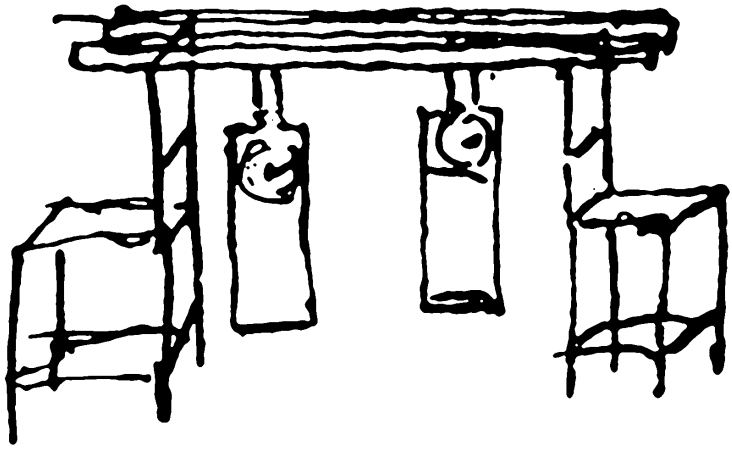


Figure 1.1: The original drawing of two pendulum clocks mounted on a common support by Christiaan Huygens [12]

### 1.1.2 Partial synchronization

Synchronization comes in different forms. One trivial form of synchronization is *full synchronization*, also called *complete synchronization*, which refers to the situation where all the systems inside a network behave identically. Other typical forms of synchronization include *phase synchronization*, *generalized synchronization*, *partial synchronization*, *etc.* Phase synchronization only requires a relation between the phases of the systems, while their amplitudes can be uncorrelated [30]. Generalized synchronization means that the states (outputs) of the coupled systems satisfy a given function [30]. In this thesis, we focus on *partial synchronization* or *cluster synchronization*. Partial synchronization is a form of incomplete synchronization where only some (but not all) systems in the network are synchronized to the same trajectory. When the systems in a network are partially synchronized, they can be grouped into different clusters, and within each cluster, synchrony is observed but it is not present between systems from different clusters. Partial synchronization often occurs in large, complex systems. One typical example concerns the synchronous activities of neurons in parts of the human brain [31]. It is worth noting that full synchronization is not always desirable: synchronization of an excessive amount of neurons can cause brain disorders like epilepsy and Parkinson's disease [32]. Considerable attention has been attracted to partial synchronization recently, see, *e.g.*, [33, 34, 35, 36, 37], and the references therein. Sometimes, there exist time-delays in and between the systems, for instance, due to the transmission time of signals among the systems, which may induce partial synchronization. Some studies have been performed to study partial synchronization of networked systems with delayed couplings, see [38, 39, 40, 3, 41].

Unlike full synchronization, partial synchronization can have many different patterns in a network of coupled systems[1]. For example, a network consisting of three systems, labeled by $1, 2, 3$, can be grouped into clusters (partitioned) in five ways:

- nodes 1 and 2 in one cluster, node 3 being separate;
- nodes 1 and 3 in one cluster, node 2 being separate;
- nodes 2 and 3 in one cluster, node 1 being separate;
- nodes 1, 2 and 3 in one cluster;
- no cluster (nodes 1, 2, 3 are separate from each).

Moreover, the number of possible partitions increases rapidly with the number of systems in a network. This raises the first question considered in this thesis: Which partial synchronization patterns will be observed given a network? In particular, which partial synchronization patterns are supported by the network topology, the coupling function (coupling strength, time-delay, *etc.*), and the dynamics of the individual systems? A pattern of partial synchronization supported by the network structure can be characterized by a *partial synchronization manifold* which is a positively invariant set[2] in the state space of the coupled systems [3]. On a partial synchronization manifold, the systems show a pattern of partially synchronous motion. Then the question can be solved by identifying partial synchronization manifolds and analyzing the stability of such manifolds. When a partial synchronization manifold is identified, its asymptotic stability, which guarantees the convergence of the states of the whole network to it, is also required to observe the corresponding partial synchronization pattern [1]. In practice, due to network imperfections, modeling errors, *etc.*, perturbations are often present in networks of coupled systems. This leads to the second question which is about the robustness of partial synchronization. With the presence of perturbations, the coupled systems may only show an approximate form of synchronization where the differences of the states of the synchronized systems only converge to some (small) bound (referred to as *practical synchronization* [42, 43]), or even worse, the synchrony will disappear. Therefore, it is important to investigate the robustness of partial synchronization. Several studies have been performed to investigate practical synchronization in delay-free coupled systems, see [44, 3, 43, 42]. However, no work has been done on the quantitative analysis of practical partial synchronization of interconnected systems, in particular in the presence of delayed couplings.

[1] In fact, full synchronization can be considered as a special pattern of partial synchronization where all systems are grouped into one cluster.

[2] Given a dynamical system, a subset of its state space is called a positively invariant set if it satisfies the property that when the system's state is inside this set at time $t_0$, the state will remain within this set for all $t > t_0$.

## 1.2 Motivation and contributions

This thesis focuses on partial synchronization in networks of dynamical systems that are interconnected via time-delay couplings. Given a delay-coupled network, the following key questions are addressed: 1) what are possible partial synchronization patterns that can be observed in the network; 2) when will a certain pattern of partial synchronization emerge in the network; 3) how robust is the partial synchronization with respect to perturbations? The network structure, coupling function and the dynamics of the individual systems all play a role in formulating a form of partial synchronization in the network. The challenge here is to find the relation between these factors and partial synchronization.

Consider $N$ dynamical systems of the form

$$\begin{aligned} \dot{x}_i(t) &= f_i(x_i(t), u_i(t)), \\ y_i(t) &= h_i(x_i(t)), \quad i = 1, 2, \ldots, N, \end{aligned} \tag{1.1}$$

with state $x_i$, input $u_i$ and output $y_i$. The systems are coupled as their inputs will depend on the outputs of the connected systems.

Two types of delayed couplings considered are given by

$$u_i(t) = k \sum_{j=1}^{N} a_{ij}[y_j(t-\tau) - y_i(t)], \text{ (invasive coupling)} \tag{1.2}$$

and

$$u_i(t) = k \sum_{j=1}^{N} a_{ij}[y_j(t-\tau) - y_i(t-\tau)], \text{ (non-invasive coupling)} \tag{1.3}$$

where $i = 1, 2, \ldots, N$, $\tau$ is the time-delay, $k > 0$ represents the coupling strength, and $a_{ij} \geq 0$ stands for the weights on the connection from system $j$ to system $i$. If $a_{ij} = 0$, system $j$ is not connected to system $i$. If $a_{ij} > 0$, system $j$ is connected to system $i$. Note that the connection here is not symmetric, *i.e.*, $a_{ij}$ and $a_{ij}$ are not necessarily equal to each other. Actually, the weights $a_{ij}$[3] contain the information on the communication structure of the network. The parameter $k$ seems redundant here as it could be embedded in the weights ($\tilde{a}_{ij} = k a_{ij}$). This parameter $k$ is used because it is assumed that the network structure is given ($a_{ij}$ is fixed) and the synchronization conditions will be checked against $k$. Following the naming conventions in [45, 46, 47, 3], coupling (1.2) is referred to as *invasive coupling* as it does not generally vanish when all systems are synchronized, and coupling (1.3) is referred to as *non-invasive coupling* as it vanishes when all systems are synchronized. The invasive coupling applies to the situation

[3] If the coupling is represented by a graph (see Chapter 2), the matrix $A = (a_{ij})$ corresponds to its adjacency matrix.

where information coming from other systems is subject to some delay due to finite transmission speed. The non-invasive coupling applies to the situation where the sensors and/or actuators have delays, for example, in a car following problem, drivers react with some delays on the distances between their cars.

The delayed type coupling is considered because in some cases time-delay cannot be neglected and time-delay can cause drastic changes to the stability properties of systems. Apart from being delayed, these couplings are also diffusive, *i.e.*, they are defined based on the differences between the (delayed) output signals of the connected systems. Diffusive coupling is an important type of coupling that has been found in many different types of networks, such as networks of coupled neurons [32, 48, 49, 50], biological systems [51, 52, 53], mechanical systems [54, 55, 56] and electrical systems [57, 58].

The objective of this thesis is to investigate and characterize the influence of dynamics of the individual systems, network structures, coupling types, and time-delays on the occurrence and patterns of partial synchronization. The occurrence of partial synchronization is determined by the existence and stability of partial synchronization manifolds, while each partial synchronization manifold corresponds to a pattern of partial synchronization. Hence, the partial synchronization problem is tackled in two steps: 1) identify partial synchronization manifolds; 2) assess the stability of partial synchronization manifolds. Through these two steps, this thesis provides:

i an efficient method to characterize all partial synchronization manifolds for delay-coupled systems (1.1), (1.2) and (1.1), (1.3);

ii tractable stability conditions for partial synchronization manifolds.

The method takes the type of system dynamics, network structures (the adjacency matrix), and the types of couplings as inputs to compute all partial synchronization manifolds of the coupled systems. The stability conditions of the partial synchronization manifolds impose restrictions on the coupling parameter $k$ and time-delay $\tau$. In this way, these results provide insights into the relation between the system dynamics, network structures, coupling parameters, time-delays and partial synchronization. The results obtained from the first step are shown in Chapter 3, and those obtained from the second step are presented in Chapter 4. The robustness of partial synchronization is also investigated by considering the practical form of partial synchronization, and the results are presented in Chapter 5. Besides, as a demonstration of the theoretical results in a practical sense, a series of experiments in a network of coupled electronic neurons are performed, and the results on synchronization of these neurons are presented in Chapter 6. Finally, Chapter 7 gives another demonstration of a real-life application of synchronization in networked systems – vehicle platooning.

## 1.3 Outline

The thesis is organized as follows.

Chapter 2 introduces the notation and concepts used throughout the thesis.

Chapter 3 presents a method to characterize all partial synchronization manifolds in delay-coupled systems. First, existence conditions for partial synchronization in networks of delay-coupled systems are presented. The existence conditions allow for systems to be different in the networks. Second, the method using these conditions is developed to compute all partial synchronization manifolds. The computational efficiency is improved for the invasive coupling case compared to the method in [3]. In addition, the method performs a decomposition of the network dynamics into synchronization error dynamics and dynamics on the manifolds. The synchronization error dynamics will be used for stability analysis of partial synchronization manifolds in the next chapters.

Chapter 4 presents sufficient conditions for partial synchronization in networks of delay-coupled systems. The synchronization is considered in an exact sense, *i.e.*, the differences between the states of the synchronized systems converge to zero. The sufficient conditions for partial synchronization are derived by assessing the stability of the synchronization error dynamics which are derived using the method developed in Chapter 3. It is shown that the synchronization error dynamics, after linearization around the zero equilibrium, can be interpreted as a linear parameter varying time-delay system. By constructing Lyapunov-Krasovskii functionals, the stability conditions for such systems, *i.e.*, the condition for (exact) partial synchronization are acquired and formulated in the form of linear matrix inequalities (LMIs) which can be efficiently solved by several existing numerical tools.

Chapter 5 presents sufficient conditions for *practical* partial synchronization in networks of delay-coupled systems. Recall that practical partial synchronization here refers to a form of approximate synchronization where the states (or outputs) of the systems in each cluster are not identical, but converge to each other up to some (small) bound. It is required that the bound goes to zero if (the size of) the perturbations go to zero. Perfect synchronization is impossible in this case due to perturbations. First, it is shown that the conditions for the existence of partial synchronization manifolds are fragile as infinitesimal perturbations on the networks can lead to violations of these conditions. In this case, partial synchronization in the exact sense cannot emerge, but practical partial synchronization may occur instead. Second, the synchronization error dynamics of the perturbed networks are derived. Due to the perturbations, this error system has no zero equilibrium. Third, the conditions for practical synchronization manifolds are derived through a practical stability analysis of the synchronization error system. This practical stability analysis does not require a zero equilibrium of

the error system. The error system is viewed as a time-delay system affected by an additive perturbation induced term. This term goes to zero when the perturbations become zero, and in the limit, the error system will have a zero equilibrium. By using Lyapunov-Krasovskii functionals, conditions for practical synchronization are derived and formulated in the form of LMIs.

Chapter 6 presents experimental results on synchronization in networks of delay-coupled electronic neurons to demonstrate the theoretical results shown in the previous chapters. An experimental setup consisting of up to sixteen electronic neurons is used. Each electronic neuron is an electric circuit board realization of a mathematical model for biological neurons. First, the model of the electronic neuron and its behavior modes are introduced. Second, a cross correlation-based notion of practical synchronization is introduced to quantify the synchronization of the electronic neurons. Note that the electronic neurons are only expected to be synchronized in a practical sense, that is, the outputs of the synchronized neurons only converge to each other within a small bound. Exact synchronization is not possible as all electronic neurons are slightly different because, for instance, no electrical components are identical. Finally, a series of experiments are performed with different network settings. The experiments confirm the theory well though the theoretical results tend to be conservative. The synchronization regions in network parameter space determined by the theoretical conditions are a qualitative match of those obtained via experiments, although the former are smaller than the latter reflecting the conservatism of the method developed in Chapter 4. However, despite the conservatism, the theoretical results are still able to provide an estimation of bounds of the network parameters between different synchronization patterns, and predict transitions from one pattern to another when the network parameters are changed.

Chapter 7 presents a control design to achieve platooning of vehicles with acceleration limitations as a real-life application of (controlled) network synchronization. The control design, grounded on model predictive model (MPC), employs a cooperative adaptive cruise control (CACC) strategy to achieve vehicle platooning. The CACC strategy uses vehicle-to-vehicle communication to allow for inter-vehicle data exchange such that short inter-vehicle distances can be reached while guaranteeing safety. The MPC technique is chosen as the basis of the design because of its ability to easily accommodate the acceleration limitation. To improve the cohesion of the platoon when encountering disturbances, a bi-directional network topology is used. It is shown that compared to the commonly used unidirectional topology, the bi-directional topology allows the vehicles to maintain shorter distances. This indicates that choosing a proper network topology can improve the performance of synchronization.

Chapter 8 summarizes the conclusions of all chapters and gives recommendations for future research.

# Chapter 2

# Preliminaries

**Abstract:** This chapter introduces the notation and concepts that are used throughout the thesis. Section 2.1 introduces the notation. Section 2.2 presents the concept of semipassive systems. Section 2.3 deals with time-delay systems, focusing on the stability properties. Section 2.4 introduces linear matrix inequality and its connection with stability analysis. Finally, section 2.5 gives an introduction to some basics of graph theory.

## 2.1 Notation

### 2.1.1 Sets and numbers

The symbols $\mathbb{C}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{+}$, $\mathbb{Z}$ and $\mathbb{Z}_{\geq 0}$ denote the sets of complex, real, non-negative real, positive real numbers, integers, and non-negative integers, respectively. $\mathrm{Re}(\lambda)$ and $\mathrm{Im}(\lambda)$ represent the real and imaginary parts of a complex number $\lambda$, respectively. The imaginary unit is denoted by $j = \sqrt{-1}$.

### 2.1.2 Vectors, matrices and norms

The space $\mathbb{R}^n$ denotes the $n$-dimensional real vector space. For a vector $v \in \mathbb{R}^n$, its Euclidean norm is given by $\|v\| := \sqrt{v^\top v}$, where $v^\top$ denotes the transpose of $v$. The space $\mathbb{C}^n$ denotes the $n$-dimensional complex vector space. For a vector $v \in \mathbb{C}^n$, its Euclidean norm is given by $\|v\| := \sqrt{v^H v}$, where $v^H$ denotes the conjugate transpose

of $v$. The $n$-dimensional vector with all entries 1 is denoted by $\mathbf{1}_n$, or simply $\mathbf{1}$ if no confusion arises.

The spectral norm of a matrix $A \in \mathbb{R}^{m\times n}$, denoted by $\|A\|$, is the induced 2-norm $\|A\| := \max_{x\in\mathbb{R}^n} = \frac{\|Ax\|}{\|x\|}$. The induced $\infty$-norm of a matrix $A \in \mathbb{R}^{m\times n}$, denoted by $\|A\|_\infty$, is defined as $\|A\|_\infty := \max_{1\le i\le m} \sum_{1\le j\le n}^{n} |a_{ij}|$, where $|a_{ij}|$ is the absolute value of the $ij$th entry of the matrix A. The notations $\mathrm{rank}(A)$, $\ker(A)$, and $\sigma(A)$ stand for the rank, the right kernel, and the spectrum of a matrix $A$. Note that in this thesis, a spectrum $\sigma(\cdot)$ is considered as a multiset. A multiset is a collection of objects (called elements) where the elements may appear more than once [59]. For example, for the matrix $A$ given by

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

the spectrum $\sigma(A)$ is the multiset $\{1,1,2\}$.

The $n \times n$ identity matrix is denoted by $I_n$, or simply $I$ if no confusion can arise. The $n \times m$ matrix with all entries 0 is denoted by $\mathbf{0}_{m\times n}$, or simply $\mathbf{0}$ if no confusion can arise. The symbol $\otimes$ represents the Kronecker product of two matrices. If $A$ is an $m \times n$ matrix and $B$ is a $p \times q$ matrix, the Kronecker product $A \otimes B$ is defined as:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix} \in \mathbb{R}^{mp\times nq},$$

where $a_{ij}$ is the $ij$th entry of $A$.

The operator $\mathrm{diag}()$ with matrix arguments creates a block-diagonal matrix with the arguments on the main diagonal. Let $A$ and $B$ be two square matrices, then $\mathrm{diag}(A,B)$ is defined as

$$\mathrm{diag}(A,B) = \begin{pmatrix} A & \mathbf{0} \\ \mathbf{0} & B \end{pmatrix}.$$

A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is called positive (negative) definite, denoted by $A > 0$ ($A < 0$), if $x^\top Ax > 0$ ($x^\top Ax < 0$), $\forall x \in \mathbb{R}^n \backslash \{\mathbf{0}\}$. It is positive (negative) definite if and only if all of its eigenvalues are positive (negative). It is called positive semi-definite, denoted by $A \ge 0$, if $x^\top Ax \ge 0, \forall x \in \mathbb{R}^n$. It is semi-definite if and only if all of its eigenvalues are non-negative.

### 2.1.3 Functions

A function $V : X \to \mathbb{R}_{\geq 0}$ defined on a subset $X$ of $\mathbb{R}^n$, is called positive (negative) definite if $V(x) > 0\,(V(x) < 0), \forall x \in X \backslash \{\mathbf{0}\}$ and $V(\mathbf{0}) = 0$. It is called radially unbounded if $X = \mathbb{R}^n$ and $V(x) \to \infty$ as $\|x\| \to \infty$.

Let $r$ be a non-negative integer. A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is called $\mathscr{C}^r$-smooth if it has continuous derivatives up to order $r$.

A continuous function $\alpha : [0, a) \to [0, \infty)$ is of class $\mathscr{K}$ if it is strictly increasing and $\alpha(0) = 0$ [60].

## 2.2 Semipassive systems

This section introduces the concept of semipassivity of dynamical systems. Semipassivity is a weakened version of the passivity property, as clarified in the following definition.

**Definition 2.2.1.** *[61, 62] Consider a system of the form*

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), \\ y(t) &= h(x(t)), \end{aligned} \tag{2.1}$$

*where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^m$, and sufficiently smooth functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}^m$. Suppose that there exist a $\mathscr{C}^r$-smooth, $r \geq 1$ nonnegative storage function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}, V(0) = 0$ and a scalar function $S : \mathbb{R}^n \to \mathbb{R}$ such that the following inequality holds:*

$$\dot{V}(x(t)) \leq y^\top(t) u(t) - S(x(t)). \tag{2.2}$$

*The system* (2.1) *is called*

- *$\mathscr{C}^r$-passive if $S(x)$ in* (2.2) *is nonnegative for all $x \in \mathbb{R}^n$.*
- *strictly $\mathscr{C}^r$-passive if $S(x)$ in* (2.2) *is positive definite.*
- *$\mathscr{C}^r$-semipassive if $S(x)$ in* (2.2) *is nonnegative for $x$ outside some ball $\mathscr{B} = \{x \in \mathbb{R}^n \,|\, \|x\| < R\}$.*
- *strictly $\mathscr{C}^r$-semipassive if $S(x)$ in* (2.2) *is positive for $x$ outside some ball $\mathscr{B} = \{x \in \mathbb{R}^n \,|\, \|x\| < R\}$.*

Basically, when $\|x\|$ is large enough, a semipassive system behaves like a passive system. A nice property of semipassive systems is that their solutions are bounded when the systems are diffusively coupled [62]. This thesis utilizes this property to guarantee the boundedness of the solutions of coupled systems. Many systems, especially physical and biological systems, are semipassive, see [63].

## 2.3 Time-delay systems

The dynamics of the delay-coupled systems will be modeled by time-delay systems. The specific class of time-delay system addressed in this thesis is the retarded type time-delay system. In this section, some basic concepts about time-delay systems, including the initial value problem and stability of equilibra, are introduced. This section only gives a brief introduction to time-delay systems. For more details about the properties, stability analysis and control design of time-delay systems, see [64, 65, 66] and the references therein.

A time-delay system of retarded type can be described by a set of *retarded functional differential equations* which are introduced in what follows by adopting the setting and conventions from [67]. Let $\tau \geq 0$ be a given real number. $\mathscr{C}([-\tau,0],\mathbb{R}^n)$ denotes the Banach space of continuous functions mapping the interval $[-\tau,0]$ to $\mathbb{R}^n$, equipped with the supremum norm $\|\phi\|_s = \sup_{0\leq\theta\leq\tau}\|\phi(\theta)\|, \phi \in \mathscr{C}([-\tau,0],\mathbb{R}^n)$. If $t_0 \in \mathbb{R}, T \geq 0$ and $x \in \mathscr{C}([t_0-\tau,t_0+T],\mathbb{R}^n)$, then for any $t \in [t_0,t_0+T]$, the function $x_t = x_t(\theta) \in \mathscr{C}([-\tau,0],\mathbb{R}^n)$ is defined by $x_t(\theta) = x(t+\theta), -\tau \leq \theta \leq 0$. Let $f : \mathbb{R}\times\mathscr{C}([-\tau,0),\mathbb{R}^n) \to \mathbb{R}^n$ and "$\cdot$" represents the right-hand derivative[1]. Then, the relation

$$\dot{x}(t) = f(t,x_t), \tag{2.3}$$

is called a *retarded functional differential equation*, or RFDE for short.

Let the initial time instant be $t_0$, then the initial condition of (2.3) is a function $\phi \in \mathscr{C}([-\tau,0],\mathbb{R}^n)$ specifying the value of $x(t)$ at the time interval $[-\tau+t_0,t_0]$, *i.e.*, $\phi(\theta) = x(t_0+\theta), \theta \in [-\tau,0]$. The function $x_t$ is the state of (2.3). A function $x$ is a solution of (2.3) on $[t_0-\tau,t_0+T], T \geq 0$ if $x \in \mathscr{C}([-\tau+t_0,t_0+T),\mathbb{R}^n)$ and $x(t)$ satisfies (2.3) for almost all $t \in [t_0,t_0+T)$. The function $f$ is assumed to be continuous in all arguments and Lipschitz in the second argument, which guarantees existence and uniqueness of the solution. For given initial time instant $t_0$ and initial condition $\phi$, $x(t;t_0,\phi)$ denotes the solution of (2.3) through $(t_0,\phi)$.

Next, the notions of stability of an equilibrium of a time-delay system (*retarded functional differential equation*) in the sense of Lyapunov are presented. An equilibrium

[1] The right-hand derivative of a function $x(t)$ is $\dot{x}(t) = \lim_{\Delta t\to 0+}\frac{x(t+\Delta t)-x(t)}{\Delta t}$.

is a constant function $[-\tau,\ 0] \to x^*$ such that $f(t,x^*) = 0$ for all $t$. Without losing generality, it is assumed that $x^* = 0$.

**Definition 2.3.1.** *([67, 68]) Consider the RFDE,*

$$\dot{x} = f(t, x_t), \tag{2.4}$$

*and suppose that $f(t,0) = 0$ for all $t \in \mathbb{R}$. Then the zero solution of* (2.4) $x \equiv 0$ *is*

- stable *if for any $t_0 \in \mathbb{R}$ and number $\varepsilon > 0$ there is $\delta = \delta(t_0, \varepsilon) > 0$ such that $\|\phi\|_s < \delta$ implies $\|x(t;t_0,\phi)\| < \varepsilon$ for all $t \geq t_0$;*
- uniformly stable *if it is stable and the number $\delta$ can be chosen independently of $t_0$;*
- locally asymptotically stable *if it is stable and there exists a number $\delta(t_0) > 0$ such that $\|\phi\|_s < \delta$ implies $x(t;t_0,\phi) \to 0$ as $t \to \infty$.*
- globally asymptotically stable *if it is stable and $x(t;t_0,\phi) \to 0$ as $t \to \infty$.*
- locally uniformly asymptotically stable *if it is uniformly stable and there exists a number $\delta > 0$ such that for every $\varepsilon > 0$ there is a $T = T(\varepsilon, \delta) > 0$ such that $\|\phi\|_s < \delta$ implies $\|x(t;t_0,\phi)\| < \varepsilon$ for all $t \geq t_0 + T$.*
- globally uniformly asymptotically stable *if it is uniformly stable and for all $\delta > 0$ and $\varepsilon > 0$ there is a $T = T(\varepsilon, \delta) > 0$ such that $\|\phi\|_s < \delta$ implies $\|x(t;t_0,\phi)\| < \varepsilon$ for all $t \geq t_0 + T$.*
- locally uniformly exponentially stable *if it is uniformly stable and there are constants $m, \alpha, \delta > 0$ such that $\|\phi\|_s < \delta$ implies $\|x(t;t_0,\phi)\| \leq m e^{-\alpha(t-t_0)}\|\phi\|_s$ for all $t \geq t_0$.*
- globally uniformly exponentially stable *if it is uniformly stable and there are constants $m, \alpha > 0$ such that $\|x(t;t_0,\phi)\| \leq m e^{-\alpha(t-t_0)}\|\phi\|_s$ for all $t \geq t_0$.*

Similarly to the case of systems without delay, the direct Lyapunov method can also be used to analyze the stability of time-delay systems. However, special attention should be paid to the Lyapunov function $V$. In fact, it will become a functional as the state variable $x_t$ in a time-delay system is already a function. Let $V : \mathbb{R} \times \mathscr{C}([-\tau, 0], \mathbb{R}) \to \mathbb{R}$ be continuous and $x(t;t_0,\phi)$ be a solution of (2.4) through $(t_0,\phi)$. The upper right-hand derivate of $V(t,\phi)$ along the solution of (2.4) is defined as

$$\dot{V}(t,\phi) = \lim_{\Delta t \to 0^+} \sup \frac{1}{\Delta t}[V(t+\Delta t, x_{t+\Delta t}(t_0,\phi)) - V(t,\phi)].$$

Then, the following theorem can be used to assess the stability of time-delay systems.

**Theorem 2.3.2.** *(Lyapunov-Krasovskii stability theorem, [68]) Consider the RFDE* (2.4) *and suppose* $f : \mathbb{R} \times \mathscr{C}([-\tau, 0], \mathbb{R}^n) \to \mathbb{R}^n$ *is completely continuous*[2] *and* $u, v, w : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *are continuous nondecreasing functions,* $u(s)$ *and* $v(s)$ *are positive for* $s > 0$*, and* $u(0) = v(0) = 0$*. If there is a continuous functional* $V : \mathbb{R} \times \mathscr{C}([-\tau, 0], \mathbb{R}^n) \to \mathbb{R}$ *such that*

$$u(\|\phi(0)\|) \leq V(t, \phi) \leq v(\|\phi\|_s), \tag{2.5}$$

$$\dot{V}(t, \phi) \leq -w(\|\phi(0)\|), \tag{2.6}$$

*then the zero solution* $x \equiv 0$ *of* (2.3) *is uniformly stable. If* $w(s) > 0$ *for* $s > 0$ *the solution* $x \equiv 0$ *is uniformly asymptotically stable. If, in addition,* $\lim_{s \to \infty} u(s) = \infty$*, the solution* $x \equiv 0$ *is globally uniformly asymptotically stable.*

The functional $V(t, \phi)$ is called Lyapunov-Krasovskii functional (LKF). Sometimes, Lyapunov-Krasovskii functionals $V(t, \phi, \dot{\phi})$ that also depends on the state derivatives are used, see [68, 69]. In this case, one of the following norms is needed to replace the supremum norm $\|\cdot\|_s$ in (2.5):

$$\|\phi\|_W = \max\{\|\phi\|_s, \|\dot{\phi}\|_s\}, \tag{2.7}$$

or

$$\|\phi\|_W = \max_{\theta \in [-\tau, 0]} \|\phi(\theta)\| + \left[\int_{-\tau}^{0} \|\dot{\phi}(\theta)\| \mathrm{d}\theta\right]^{1/2}, \tag{2.8}$$

The conclusions in Theorem 2.3.2 still hold when $\|\phi\|_s$ is replaced by $\|\phi\|_W$, see [70, 69].

For linear time-invariant time-delay systems, the stability analysis can be also be done by using frequency domain methods, for example, by inspecting the characteristic roots of the time-delay systems.

Consider a linear time-delay systems of the form

$$\dot{x}(t) = A_0 x(t) + A_1 x(t - \tau), \tag{2.9}$$

where $x(t) \in \mathbb{R}^n$ and $A_i \in \mathbb{R}^{n \times n}, i = 0, 1$ and time delay $\tau > 0$.

The characteristic equation of (2.9) is given by

$$\det \Delta(\lambda) = 0, \tag{2.10}$$

where

$$\Delta(\lambda) := \lambda I - A_0 - A_1 e^{-\lambda \tau}.$$

The roots of (2.10) are called the *characteristic roots* of (2.9).

[2]The mapping $f : \mathbb{R} \times \mathscr{C}([-\tau, 0], \mathbb{R}^n) \to \mathbb{R}^n$ is completely continuous if it is continuous and maps bounded sets in $\mathbb{R} \times \mathscr{C}([-\tau, 0], \mathbb{R}^n)$ into bounded sets in $\mathbb{R}^n$.

**Theorem 2.3.3.** *([65]) The zero solution $x(t) \equiv 0$ of* (2.9) *is asymptotically stable if and only if all of its characteristic roots are located in the open left half-plane.*

The characteristic equation (2.10) has an infinite number of characteristic roots. However, given any finite real number $\gamma$, the number of characteristic roots in the plane $\{\lambda \in \mathbb{C} | \mathrm{Re}(\lambda) > \gamma\}$ is finite. In other words, the number of characteristic roots located on the right side of any vertical line is finite.

An important property of the characteristic roots of (2.9) is that they behave continuously with respect to the variations of the time delay and the system matrices [65]. That is, when the time delay $\tau$, the matrix $A_0$ or $A_1$ varies, the characteristic roots may transit from the left half-plane to the right half-plane (or vice versa) only by crossing the imaginary axis. Hence, the stability regions in a parameter space can be determined by checking for the imaginary axis crossings of the roots with respect to the changes of the parameters.

## 2.4 Linear matrix inequalities

In this thesis, many criteria are formulated in the form of linear matrix inequalities (LMIs). This section introduces linear matrix inequalities and their connection with stability analysis. There exist several off-the-shelf software tools that can solve LMIs efficiently, for instance, MOSEK [71], SEDUMI [72], SDPT3 [73], and the MATLAB LMI toolbox [74].

A linear matrix inequality (LMI) is a constraint of the form

$$\mathscr{A}(x) = A_0 + \sum_{i=1}^{m} x_i A_i < 0, \tag{2.11}$$

where $x = [x_1, \ldots, x_m]^\top \in \mathbb{R}^m$ is the vector of the decision variables $x_i$ and the symmetric matrices $A_i = A_i^\top \in \mathbb{R}^{n \times n}, i = 0, \ldots, m$, are given. If there exists a vector $x^* \in \mathbb{R}^m$ such that the inequality (2.11) holds, the LMI is called *feasible*.

The variables can be also matrices, which is the case for most control applications. A typical example is the Lyapunov inequality

$$A^\top P + PA < 0, \tag{2.12}$$

where $A \in \mathbb{R}^{n \times n}$ is given and the symmetric matrix $P \in \mathbb{R}^{n \times n}$ is the design variable.

Let $E_1, \ldots, E_m \in \mathbb{R}^{n \times n}$ be $m$ linearly independent symmetric matrices, where $m = n(n+1)/2$. Then the symmetric matrix $P$ can be expressed by $P = \sum_{i=1}^{m} x_i E_i$, where

$x_1, \ldots, x_m$ are the decision variables. In this way, (2.12) can be seen as a special form of (2.11):

$$A^\top \left( \sum_{i=1}^{m} x_i E_i \right) + \left( \sum_{i=1}^{m} x_i E_i \right) A = x_1 \left( A^\top E_1 + E_1 A \right) + \ldots + x_m \left( A^\top E_m + E_m A \right) < 0. \tag{2.13}$$

When using the direct Lyapunov method to analyze the stability of dynamical systems, the requirements on the Lyapunov function(al)s can in many cases be transformed into LMIs. Improved stability conditions can be obtained by using a so-called descriptor transformation, see [69] and the references therein.

## 2.5 Elementary graph theory

In this thesis, the coupling between systems is presented by graphs. This section briefly introduces some basic concepts in graph theory. The notations and terminology are adopted from [75, 76]. A *graph* $\mathscr{G}$ consists of a pair $(\mathscr{E}, \mathscr{V})$, where $\mathscr{V}$ is the set of vertices and $\mathscr{E}$ is the set of edges. Let $i, j \in \mathscr{V}$ be two nodes, then $(i, j) \in \mathscr{E}$ denotes the directed edge from node $i$ to node $j$. The edge can also be undirected if $(i, j) \in \mathscr{E}$ implies $(j, i) \in \mathscr{E}$. A graph with directed edges is called a *directed* graph, or *digraph*. Otherwise, it is an *undirected* graph. In the remainder of this section, only directed graphs are considered. An edge joining a node to itself is called a *self-loop*. If two or more edges connecting an ordered pair of nodes, the edges are called *multiple edges*. A graph is *simple* if it has neither self-loops nor multiple edges. A graph is *strongly connected* if, for any two nodes $i$, $j$ in it, there exist a directed path from $i$ to $j$.

If two nodes are connected by an edge, they are called *adjacent*. For a graph $\mathscr{G}$ with $n$ vertices, the *adjacency matrix* $A \in \mathbb{R}^{n \times n}$ is then defined as $A = (a_{ij})$ where

$$a_{ij} = \begin{cases} 1, & \text{if } (j,i) \in \mathscr{E}(\mathscr{G}), \\ 0, & \text{otherwise.} \end{cases}$$

It is also possible to assign weights to the edges, thus the graph becomes a weighted graph. Let $w_{ij}$ be the weight on the edge $j,i$, then the *weighted adjacency matrix* is defined as $A := (a_{ij})$ with

$$a_{ij} = \begin{cases} w_{ij}, & \text{if } (j,i) \in \mathscr{E}(\mathscr{G}), \\ 0, & \text{otherwise.} \end{cases}$$

To include the information on weights, a graph $\mathscr{G}$ is now extended as a triple-tuple $(\mathscr{V}, \mathscr{E}, A)$.

The *degree matrix* $D \in \mathbb{R}^{n\times n}$ of a graph is a diagonal matrix with the degree $d_i = \sum_j a_{ij}$ as entries on its diagonal, and the *Laplacian matrix* $L$ is defined by $L := D - A$.

# Chapter 3

# Characterization of partial synchronization manifolds

**Abstract:** In this chapter, a method to characterize all partial synchronization manifolds in networks of delay-coupled systems is presented. The method is developed based on a recently proposed algorithm for computing partial synchronization manifolds with improved computational efficiency and also applicable to networks of different systems. In addition, the method performs a decomposition of network dynamics into the synchronization error dynamics and the dynamics on the manifolds in a systematic way, which is important in the context of stability analysis of partial synchronization manifolds. The results presented in this chapter are published in [77] and [78].

## 3.1 Introduction

This chapter focuses on partial synchronization in networks of systems interacting via diffusive time-delay couplings. From a mathematical point of view, a prerequisite for partial synchronization is the existence of so-called *partial synchronization manifolds*, which are positively invariant sets in the state space of the delay-coupled systems. On such a manifold, the systems show partially synchronized motion. Note that the existence of a partial synchronization manifold, by itself, does not imply that in experiments synchronous behavior within each cluster of the corresponding partition will appear. The latter also requires the manifold to be asymptotically (practically) stable, which will be dealt with in the next chapters.

In [33], such manifolds are identified by exploiting the symmetry under permutation of the networks with identical systems coupled through diffusion. In [40], a so-called equitable partition is defined to identify partial synchronization manifolds. More equivalent existence conditions for partial synchronization manifolds for networks of identical systems interacting via linear diffusive time-delay couplings have been presented in [3]. Using one of the criteria, which utilizes the block structure of the reordered adjacency matrix of a network, a method for computing partial synchronization manifolds has been developed in [3]. In this chapter, an improved method is presented which is more computationally efficient when the couplings are invasive, *i.e.*, they do not vanish when full synchronization emerges. Most work on partial synchronization considers networks consisting of identical systems. In some research, systems with different dynamics are also considered, see, *e.g.*, [38, 79]. In this chapter, the method proposed in [3] is also extended to allow for networks with (some but not all) different systems at the nodes. In addition, the extended method in this chapter performs an automatic decomposition of the network dynamics into the synchronization error dynamics and the dynamics on the manifolds, which serves as a foundation for stability analysis in the subsequent chapters.

The remainder of this chapter is organized as follows. Section 3.2 shows how to characterize and compute partial synchronization manifolds. Section 3.4 presents the decomposition of the network dynamics into the synchronization error dynamics and the dynamics on the manifolds. Section 3.5 uses a numerical example to demonstrate the usage of the method to analyze partial synchronization of delay-coupled systems.

## 3.2 Characterization of partial synchronization manifolds

In this section, some relevant concepts on partial synchronization manifolds are introduced, and the existence conditions for partial synchronization manifolds in [3] are extended to allow for networks with different systems at the nodes. As an extension of [3], the settings and conventions in [3] are adopted throughout this section.

### 3.2.1 Existence of partial synchronization manifolds

We consider a network consisting of $N$ systems whose coupling is represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where

- $\mathcal{V} = \{1, 2, \ldots, N\}$ is a finite set of nodes with cardinality $|\mathcal{V}| = N$;

- $\mathscr{E} \subset \mathscr{V} \times \mathscr{V}$ is the set of *directed edges*, where the edge $(i, j)$ points from node $i$ to node $j$;
- $A = (a_{ij}) \in \mathbb{R}^{N\times N}$ is the weighted adjacency matrix, where $a_{ij} > 0$ represents the weight of edge $(j,i)$ when $(j,i) \in \mathscr{E}$, and $a_{ij} = 0$ when $(j,i) \notin \mathscr{E}$.

Note that the graphs considered here are *simple* and *strongly connected*.

Every node hosts a dynamical system of the form

$$\begin{cases} \dot{x}_i(t) = f_i(x_i(t)) + B_i u_i(t), \\ y_i(t) = C_i x_i(t), \end{cases} \tag{3.1}$$

where $x_i(t) \in \mathbb{R}^n$ are the states, $f_i : \mathbb{R}^n \to \mathbb{R}^n$ are sufficiently smooth functions, $u_i(t) \in \mathbb{R}^m$ are the inputs, $y_i(t) \in \mathbb{R}^m$ are the outputs, $B_i \in \mathbb{R}^{n\times m}$ are the input matrices, $C_i \in \mathbb{R}^{m\times n}$ are the output matrices, and $i = 1,\ldots,N$. Here, we assume that all $C_iB_i$ are similar to positive definite matrices, *i.e.*, there exist invertible matrices $M_i \in \mathbb{R}^{m\times m}$ and positive definite matrices $W_i \in \mathbb{R}^{m\times m}$ such that $C_iB_i = M_i^{-1}W_iM_i$. This assumption guarantees that all the systems are left-invertible [80]. A dynamical system is called left-invertible when the input-output map of the system is injective [81]. Left-invertibility is introduced to render the sufficient condition presented later in this section for the existence of partial synchronization manifolds necessary, *cf.* [3]. One common choice of the matrices $B_i$ and $C_i$ is $B_i = C_i = I$, which corresponds to the case of full state feedback.

The systems (3.1) interact via either one of the following two types of diffusive couplings:

*invasive coupling*

$$u_i(t) = k \sum_{j\in\mathscr{N}_i} a_{ij}[y_j(t-\tau) - y_i(t)] \tag{3.2}$$

or

*non-invasive coupling*

$$u_i(t) = k \sum_{j\in\mathscr{N}_i} a_{ij}[y_j(t-\tau) - y_i(t-\tau)], \tag{3.3}$$

where $\mathscr{N}_i$ is the neighboring set of node $i$, *i.e.*, $\mathscr{N} := \{j \in \mathscr{V} \,|\, (j,i) \in \mathscr{E}\}$, $a_{ij}$ are the entries of the weighted adjacency matrix $A$, $\tau$ is the time-delay, and $k$ is the coupling strength.

For the coupled systems (3.1), (3.2) or (3.1), (3.3), a solution denoted by $x(\cdot) = \left[x_1(\cdot)^\top \cdots x_N(\cdot)^\top\right]^\top$ is a partially synchronous solution if there exist nodes $i, j \in \mathscr{V}$ with $i \neq j$ such that

$$x_i(t) = x_j(t), \forall t \geq t_0 \text{ whenever } x_i(\theta) = x_j(\theta) \text{ for } \theta \in [t_0 - \tau, t_0]. \tag{3.4}$$

Here $x_i(\cdot)$ is the component of the solution corresponding to node $i$ in the network.

When a partially synchronous solution exists, the nodes can be grouped into clusters such that the systems at the nodes inside each cluster have equal states. In order to describe this clustering of nodes in a graph $\mathscr{G}$, the concept of partition is used. A partition $\mathscr{P}$ is a set of nonempty, disjoint subsets of $\mathscr{V}$ whose union is $\mathscr{V}$. These subsets are referred to as *parts* of the partition. In this way, a part represents a cluster of the nodes. The number of parts is denoted by $\kappa$. A partition is parameterized by a permutation matrix $\Pi$. The permutation matrix $\Pi$ is associated with an equivalence relation $\sim$, which is such that $i \sim j$ if the $ij$-th entry of $\Pi$ is equal to 1. Note that $\kappa = \dim\ker(I_N - \Pi)$.

**Example 3.2.1.** *Consider a graph $\mathscr{G}$ with $\mathscr{V} = \{1,2,3,4,5\}$. The partition $\mathscr{P} = \{\{1,3,4\},\{2,5\}\}$ with $\kappa = 2$ parts can be presented by the permutation matrix*

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

*It is easy to verify that* $\dim\ker(I_5 - \Pi) = 2$.

We denote the state space of the coupled systems by $\mathscr{C}([-\tau,0],\mathbb{R}^{Nn})$ and the states of the coupled systems by $x_t \in \mathscr{C}([-\tau,0],\mathbb{R}^{Nn})$ with

$$x_t(\theta) := \begin{pmatrix} x_{t,1}(\theta) \\ x_{t,2}(\theta) \\ \vdots \\ x_{t,N}(\theta) \end{pmatrix} = \begin{pmatrix} x_1(t+\theta) \\ x_2(t+\theta) \\ \vdots \\ x_N(t+\theta) \end{pmatrix}, \quad \theta \in [-\tau,0].$$

Then, with a permutation matrix $\Pi$, the condition of form (3.4) can be expressed as

$$x_t(\theta) = (\Pi \otimes I_n)x_t(\theta), \quad \forall \theta \in [-\tau,0], \quad \forall t \geq t_0, \tag{3.5}$$

or equivalently, $x_t \in \mathscr{M}(\Pi)$ for all $t \geq t_0$, where

$$\mathscr{M}(\Pi) := \{\phi \in \mathscr{C}([-\tau,0],\mathbb{R}^{Nn}) \,|\, \phi(\theta) = \mathrm{col}(\phi_1(\theta),\phi_2(\theta),\ldots,\phi_N(\theta)),$$
$$\phi_i(\theta) \in \mathbb{R}^n, i = 1,\ldots,N, \phi(\theta) = (\Pi \otimes I_n)\phi(\theta) \forall \theta \in [-\tau,0]\}.$$

is the set of partially synchronous states induced by the permutation matrix $\Pi$.

Note that $\mathscr{M}(\Pi)$ is a linear subspace of $\mathscr{C}([-\tau,0],\mathbb{R}^{Nn})$.

The set $\mathscr{M}(\Pi)$ is called a partial synchronization manifold if it is a positively invariant set for the coupled systems (3.1), (3.2) or (3.1), (3.3), *i.e.*, $x_{t_0} \in \mathscr{M}(\Pi)$ implies $x_t \in \mathscr{M}(\Pi)$ for all $t > t_0$. This brings the following definition.

**Definition 3.2.2.** *([3]) The set $\mathscr{M}(\Pi)$ with permutation matrix $\Pi$ for which $1 < \kappa < N$ is a partial synchronization manifold for the coupled systems* (3.1)*,* (3.2)*, or* (3.1)*,* (3.3)*, if it holds that $x_t \in \mathscr{M}(\Pi), \forall t > t_0$ whenever $x_{t_0} \in \mathscr{M}(\Pi)$.*

A partition $\mathscr{P}$ associated with the permutation matrix $\Pi$ is called *viable* if the corresponding set $\mathscr{M}(\Pi)$ is a partial synchronization manifold. However, identifying a viable partition can be challenging since a direct verification of the invariance property is difficult. With the help of another permutation matrix $R$ introduced in what follows, the viability of a partition can be checked by using some algebraic conditions on the adjacency matrices.

Given a partition $\mathscr{P}$, the nodes can be relabeled such that the first $\kappa_1$ nodes belong to cluster 1, the second $\kappa_2$ nodes belong to cluster 2, and so on:

$$\{\{1,\ldots,\kappa_1\},\{\kappa_1+1,\ldots,\kappa_1+\kappa_2\},\ldots\},$$

where $\kappa_i, i = 1,\ldots,\kappa$ are the sizes of parts of $\mathscr{P}$. Mathematically, this can be done by using another permutation matrix $R$, which is referred to as the *reordering matrix*. An $N \times N$ permutation matrix $R$ is a reordered matrix if it satisfies

$$R^\top \Pi R = \begin{bmatrix} \Pi_C(\kappa_1) & & & 0 \\ & \Pi_C(\kappa_2) & & \\ & & \ddots & \\ 0 & & & \Pi_C(\kappa_\kappa) \end{bmatrix}, \quad \sum_{i=1}^{\kappa} \kappa_i = N, \tag{3.6}$$

*i.e.*, $R^\top \Pi R$ is a block diagonal matrix with $\kappa$ blocks $\Pi_C(\kappa_i)$, each of which is a $\kappa_i \times \kappa_i$ cyclic permutation matrix with

$$\Pi_C(\kappa_i) = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{\kappa_i \times \kappa_i}, \tag{3.7}$$

and $\Pi_C(1) = 1$.

Using $R$, the *reordered adjacency matrix* $A_R$ can be constructed as follows:

$$A_R = R^\top A R = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1\kappa} \\ A_{21} & A_{22} & \cdots & A_{2\kappa} \\ \vdots & \ddots & \ddots & \vdots \\ A_{\kappa 1} & A_{\kappa 2} & \cdots & A_{\kappa\kappa} \end{bmatrix}, \quad A_{ij} \in \mathbb{R}^{\kappa_i \times \kappa_j}. \tag{3.8}$$

The reordered adjacency matrix has a block-structure where the diagonal blocks describe the interactions between the systems inside the same clusters and the off-diagonal blocks describe the interactions between the systems in different clusters. Now, we can introduce the existence conditions for partial synchronization manifolds. To allow for systems to have different dynamics ($f_i(\cdot) \neq f_j(\cdot)$, $B_i \neq B_j$ and $C_i \neq C_j$, for some $i, j \in \mathscr{V}$), we extend Theorems 3 and 4 in [3] and give the existence conditions in the following theorem:

**Theorem 3.2.3.** *Given an adjacency matrix $A$ and a permutation matrix $\Pi$ of the same dimension, and assuming that $C_iB_i$, $i = 1,\ldots,N$ are similar to positive definite matrices, the following statements are equivalent:*

*(1) $\mathscr{M}(\Pi)$ is a partial synchronization manifold for (3.1) and (3.2), respectively (3.1) and (3.3);*

*(2) all blocks, respectively all off-diagonal blocks, of the reordered adjacency matrix* (3.8)*, partitioned in blocks of size $\kappa_i \times \kappa_j$ have constant row-sums and, in addition, $\mathscr{F}$, $\mathscr{B}$ and $\mathscr{C}$ defined by*

$$\mathscr{F} := \begin{bmatrix} f_1(\cdot) \\ f_2(\cdot) \\ \vdots \\ f_N(\cdot) \end{bmatrix}, \mathscr{B} := \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix}, \mathscr{C} := \begin{bmatrix} C_1^\top \\ C_2^\top \\ \vdots \\ C_N^\top \end{bmatrix}$$

*satisfy the conditions*

$$\mathscr{F} = (\Pi \otimes I_n)\mathscr{F},\ \mathscr{B} = (\Pi \otimes I_n)\mathscr{B},\ \mathscr{C} = (\Pi \otimes I_n)\mathscr{C}. \tag{3.9}$$

Here, the conditions $\mathscr{F} = (\Pi \otimes I_n)\mathscr{F}$, $\mathscr{B} = (\Pi \otimes I_n)\mathscr{B}$ and $\mathscr{C} = (\Pi \otimes I_n)\mathscr{C}$ indicate that all the nodes in the same cluster host systems with the same input-output dynamics, but the nodes in different clusters can host systems with different input-output dynamics. Thus, this theorem can be also applied to the situation where some (but not all) systems in the networks are different, thus marking it as an extension of Theorems 3 and 4 of [3] where only networks of identical systems are considered.

From Theorem 3.2.3, it can be found that only the topology (the adjacency matrix) and, to a very limited extent, the types of dynamics of the systems play a role on the existence of partial synchronization manifolds, while the other parameters such as delay $\tau$ and coupling strength $k$ have no impact. The intuition behind Theorem 3.2.3 is the following. The row-sum conditions on the adjacency matrix guarantee that every system inside the same cluster receives the same (amount of) interactions from other systems, and the restriction on the dynamics type of the systems ensures that all systems in the same cluster have the same input-output dynamics. Under

those conditions, the solutions will be able to stay on $\mathscr{M}(\Pi)$ if they start on $\mathscr{M}(\Pi)$. Note that the row-sum conditions for networks with non-invasive coupling are less restrictive than the ones with invasive coupling. For the non-invasive coupling case, only off-diagonal blocks of $A_R$ are required to have constant row-sums. This is because that the interactions between the systems in the same clusters (represented by the diagonal blocks) will not affect the invariance of the set $\mathscr{M}(\Pi)$ for networks with non-invasive coupling. The non-invasive coupling of the systems in the same clusters will vanish when these systems are synchronized. For example, suppose that systems (1) and (2) are in the same cluster, then the coupling terms $a_{12}[y_2(t-\tau)-y_1(t-\tau)]$ and $a_{21}[y_1(t-\tau)-y_2(t-\tau)]$ vanish independently of the values of $a_{12}$ and $a_{21}$ when the systems (1) and (2) are synchronized.

The results presented in Theorem 3.2.3 can be extended to networks with systems of a more general form

$$\begin{cases} \dot{x}_i(t) = \bar{f}_i(x_i(t), u_i(t)), \\ y_i(t) = h_i(x_i(t)), \end{cases}$$

where functions $\bar{f}_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, and functions $h_i : \mathbb{R}^n \to \mathbb{R}^m$, $i = 1, \ldots, N$. A prerequisite here is that these systems are left-invertible. Then the only change is that the conditions (3.9) will be replaced by

$$\begin{bmatrix} \bar{f}_1(\cdot,\cdot) \\ \bar{f}_2(\cdot,\cdot) \\ \vdots \\ \bar{f}_N(\cdot,\cdot) \end{bmatrix} = (\Pi \otimes I_n) \begin{bmatrix} \bar{f}_1(\cdot,\cdot) \\ \bar{f}_2(\cdot,\cdot) \\ \vdots \\ \bar{f}_N(\cdot,\cdot) \end{bmatrix}, \begin{bmatrix} g_1(\cdot) \\ g_2(\cdot) \\ \vdots \\ g_N(\cdot) \end{bmatrix} = (\Pi \otimes I_n) \begin{bmatrix} g_1(\cdot) \\ g_2(\cdot) \\ \vdots \\ g_N(\cdot) \end{bmatrix}.$$

Additionally, the results presented in Theorem 3.2.3 can also be extended to networks with multiple time-delays. This is done by decomposing the networks into subnetworks according to the time-delays and then imposing the row-conditions and conditions (3.9) on all subnetworks. Consider a network with $p$ distinct delays, whose corresponding coupling functions are

$$u_i(t) = \sum_{j \in \mathscr{N}_i} a_{ij}[y_j(t-\tau_{ij}) - y_i(t)], \tag{3.10}$$

or

$$u_i(t) = \sum_{j \in \mathscr{N}_i} a_{ij}[y_j(t-\tau_{ij}) - y_i(t-\tau)], \tag{3.11}$$

where $\tau_{ij} \in \{\tau_1, \tau_2, \ldots, \tau_p\}$ with $0 < \tau_1 < \tau_2 < \ldots < \tau_p = \tau$. Define the matrices $A_\ell, \ell = 1, \ldots, r$ as

$$\tilde{A}_\ell = (\tilde{a}_{\ell,ij}), \text{ with } \tilde{a}_{\ell,ij} = \begin{cases} a_{ij} & \text{if } \tau_{ij} = \tau_\ell, \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

Therefore, the network $\mathscr{G}$ can be decomposed as

$$\mathscr{G}(\mathscr{V},\mathscr{E},A) = \tilde{\mathscr{G}}_1(\mathscr{V},\tilde{\mathscr{E}}_1,\tilde{A}_1) \oplus \cdots \oplus \tilde{\mathscr{G}}_p(\mathscr{V},\tilde{\mathscr{E}}_r,\tilde{A}_p), \tag{3.13}$$

where $\tilde{\mathscr{E}}_\ell \subset \mathscr{V} \times \mathscr{V}$ presents the set of edges corresponding to $\tau_\ell, \ell = 1,\ldots,p$.
Then, we have the following condition for existence of partial synchronization manifolds for this type of networks.

**Theorem 3.2.4.** *Consider a graph $\mathscr{G}$ that decomposes as in* (3.13)*. Given an adjacency matrix $A$ and a permutation matrix $\Pi$ of the same dimension, and assuming that $C_iB_i$, $i = 1,\ldots,N$ are similar to positive definite matrices, the following statements are equivalent:*

*(1) $\mathscr{M}(\Pi)$ is a partial synchronization manifold for (3.1) and (3.2), respectively (3.1) and (3.3);*

*(2) for each $\ell = 1,\ldots,p$, all blocks, respectively all off-diagonal blocks, of each block-structured matrices $R^\top \tilde{A}_\ell R$, partitioned in blocks of size $\kappa_i \times \kappa_j$ according to* (3.6)*, have constant row-sums, and, in addition, $\mathscr{F}$, $\mathscr{B}$ and $\mathscr{C}$ defined by*

$$\mathscr{F} := \begin{bmatrix} f_1(\cdot) \\ f_2(\cdot) \\ \vdots \\ f_N(\cdot) \end{bmatrix}, \mathscr{B} := \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix}, \mathscr{C} := \begin{bmatrix} C_1^\top \\ C_2^\top \\ \vdots \\ C_N^\top \end{bmatrix}$$

*satisfy the conditions*

$$\mathscr{F} = (\Pi \otimes I_n)\mathscr{F},\ \mathscr{B} = (\Pi \otimes I_n)\mathscr{B},\ \mathscr{C} = (\Pi \otimes I_n)\mathscr{C}. \tag{3.14}$$

Like Theorem 3.2.3, this is also an extension of Theorems 5 and 6 in [3] to allow for different systems at the nodes of the networks.

In addition to multiple time-delays, networks may have physically distinct communication channels that have different characteristic interaction weights. Similarly, these networks can be decomposed into subnetworks according to their weights. Then we can apply the row-sum criterion in each subnetwork for identifying partial synchronization manifolds in the whole networks, see Appendix A for more information.

It should be emphasized that the existence of a partial synchronization manifold is only necessary for the networks to be partially synchronized. In order to observe partial synchronization of a network, the stability of a partial synchronization manifold is also required. It will be shown in the subsequent sections the coupling parameters $k$ and $\tau$ affect the stability of a partial synchronization manifold.

## 3.3 Computation of partial synchronization manifolds

This section introduces a method for computing all partial synchronization manifolds for networks with identical/non-identical dynamical systems at the nodes. It is an improved version of the algorithm in [3] which only works for networks with identical systems. The improvements are two-fold: 1) increasing computational efficiency for the networks with invasive couplings; 2) allowing for networks with different systems at the nodes.

The original algorithm for identifying all partial synchronization manifolds in [3] consists of two ingredients: generating all possible partitions, and checking the viability of each partition (*i.e.*, checking whether or not it corresponds to a partial synchronization manifold). For the latter, the algorithm checks the row-sum criteria in Theorem 3 or 4 in [3]. However, this algorithm only considers networks with identical systems. Moreover, it scales badly with the network size due to its combinatorial nature. The number of possible partitions of $N$ systems grows with $N$ in an exponential-like way. To see that, we first review how all the partitions are generated using the original algorithm in [3]. In this algorithm, each partition of a network with $N$ systems is presented by a $N$-digit code $d_1 \cdots d_N$, which is defined in such a way that the nodes $i, j \in \{1, \ldots, N\}$ belong to the same cluster if and only if $d_i = d_j$. Table 3.1 demonstrates the generation of all possible partitions using this coding system in the original algorithm from [3]. System 1 is always assigned code 0. When adding a second system, there will be two possible partitions: one coded by 00 when systems 1 and 2 belong to the same cluster, the other one coded by 01 when system 2 belongs to a new cluster. Now a third system is added. Based on every possible partition of systems $\{1,2\}$, the partitions of systems $\{1,2,3\}$ are generated. Starting with partition 00, the third system can be put into the same cluster of systems 1 and 2 or a new cluster, coded by 000 or 001 respectively. Similarly, for partition 01, there are three possibilities: 010 (systems 1 and 3 in the same cluster), 011 (systems 2 and 3 in the same cluster) or 012 (systems 3 in a new cluster). Continuing this procedure, all possible partitions for networks with $N$ systems are generated. Note that the number of partitions generated in this way is known as the Bell number. In combinatorial mathematics, Bell numbers $\{B_0, B_1, \ldots\}$, introduced in [82], count the numbers of partitions of a set. The $N$-th Bell number $B_N$ is the number of partitions of a set of size $N$. Starting with $B_0 = 1$, the Bell numbers can be generated by using the following recurrence relation [83]:

$$B_n = \sum_{k=1}^{n-1} \binom{n-1}{k} B_k, \text{ for } n \geq 1,$$

where $\binom{n-1}{k}$ define the binomial coefficients. It can be shown that the Bell

numbers increase in an exponential-like way with $N$. The first few Bell numbers are $1,1,2,5,15,52,203,877,4140,\ldots$

| System 1 | Systems 1,2 | Systems 1,2,3 | Systems 1,2,3,4 | $\cdots$ |
|---|---|---|---|---|
| 0 | 00 | 000 | 0000 | |
| | | | 0001 | |
| | | 001 | 0010 | |
| | | | 0011 | |
| | | | 0012 | |
| | | | | $\ddots$ |
| | 01 | 010 | 0100 | |
| | | | 0101 | |
| | | | 0102 | |
| | | 011 | 0110 | |
| | | | 0111 | |
| | | | 0112 | |
| | | 012 | 0120 | |
| | | | 0121 | |
| | | | 0122 | |
| | | | 0123 | |
| | | | | $\ddots$ |

Table 3.1: Generating the set of all partitions of a number of systems using the original algorithm [3].

The improved method proposed in this section includes the same ingredients with some modifications to the first ingredient. The second ingredient remains the same and uses the row-sum criteria in Theorem 3.2.3 for the viability check. Therefore, the second ingredient will not be elaborated on in this chapter.

The first ingredient now involves two $N$-dimensional integer vectors $v_F, v_S \in \mathbb{Z}^N$. The first vector $v_S = (S_i), i = 1,\ldots,N$ will be used to limit the number of partitions generated for the case of the invasive coupling (3.2). It is defined in such a way that $S_i = S_j$ if and only if the sum of the $i$-th row equals that of the $j$-th row for $i, j \in \{1,2,\ldots,N\}$. In other words, it presents the (in)equalities between the row-sums of the adjacency matrix before being reordered. As is seen from Theorem 3.2.3, when the coupling is invasive, a necessary condition for a viable partition is that each block of the (full) reordered adjacency matrix has constant row-sums. This indicates that for any two nodes to be in the same part of a viable partition, it is necessary that their corresponding rows in the original adjacency matrix have the same row-sum. Hence, $v_S$ can be used to limit the options when allocating the nodes into clusters, thus

limiting the number of generated partitions. Given the adjacency matrix, the method will construct this vector automatically, see the procedure in Appendix B. The second vector $v_F = (F_i) \in \mathbb{Z}^N, i = 1,2,\ldots,N$ that represents the types of dynamical systems at all the nodes is also used to restrict the number of partitions. It is defined such that $F_i = F_j$ if and only if $f_i(\cdot) = f_j(\cdot)$, $B_i = B_j$ and $C_i = C_j$ for any $i, j \in \{1,2,\ldots,N\}$. To be precise, this vector is used such that the conditions (3.9) are automatically satisfied when selecting possible partitions. The types of dynamical systems are assumed to be known, *i.e.*, $v_F$ is given.

Now, we demonstrate how the vectors $v_F$ and $v_S$ are used when generating the partitions. To satisfy the conditions on the types of system dynamics, $v_F$ is exploited to reduce the number of partitions. System 1 still gets code 0. However, when adding subsequent systems, the possibilities will be restricted by the (in)equalities of the corresponding entries of $v_S$. When the coupling is invasive, $v_S$ is also used to determine where to put the newly added systems to further reduce the number of generated partitions. See the procedure illustrated in Algorithm 1. In the case of non-invasive coupling, the procedure is similar with the exception that there is no $v_S$, *i.e.*, when adding the sequential system $i$, $D_i = \{d_j \,|F_i = F_j,\ j = 1,\ldots,i-1\} \cup \{\max(d_1,d_2,\ldots,d_{i-1})+1\}$.

**Algorithm 1:** Generate partitions (invasive coupling case), [77]

**Input:** $v_S = [S_1\ \cdots\ S_N]^\top$, $v_F = [F_1\ \cdots\ F_N]^\top$
**Output:** set of partition codes $\{d_1 \cdots d_N\}$ for systems $\{1,\ldots,N\}$
let $D_1 = \{0\}$ be the set of partition codes for system $\{1\}$ ($d_1$=0);
**for** *i=2,...,N* **do**
  set the set of partition codes for systems $\{1,\ldots,i\}$ empty;
  **foreach** *element in the set of partition codes for systems* $\{1,\ldots,i-1\}$ **do**
    retrieve $d_1,\ldots,d_{i-1}$;
    compute the set $D_i =$
    $\underbrace{\{d_j \mid S_i = S_j \,\&\, F_i = F_j,\ j = 1,\ldots,i-1\}}_{\text{System i allowed in the same clusters of the other systems due to the same row-sums and same dynamics}} \cup \underbrace{\{\max(d_1,\ldots,d_{i-1})+1\}}_{\text{System i in a new cluster}}$;
    append every element of $D_i$ to the current code $d_1 \cdots d_{i-1}$ and add all corresponding codes $d_1 \cdots d_i$ in the set of partition codes for systems $\{1,\ldots,i\}$;
  **end**
**end**

With the partitions generated, the viability of these partitions can be checked by using the row-sum criterion from Theorem 3.2.3.

**Remark.** *The number of possible partitions for networks generated by the improved algorithm can be inferred from the structures of $v_F$ and $v_S$ (in case of invasive coupling).*

*For simplicity, we focus on $v_F$ first. Consider a network of systems interacting via invasive coupling with a constant row-sum adjacency matrix ($v_S = \mathbf{1}$) or non-invasive coupling. Suppose $v_F$ has $d$ distinct elements, i.e., the systems have $d$ distinct types of input-ouput dynamics. Denote $k_i, i = 1, \ldots, d$ as the numbers of entries with equal values. The number of partitions $N_{\mathscr{P}}$ generated by the new procedure is*

$$N_{\mathscr{P}} = \prod_{i=1}^{d} B_{k_i}. \tag{3.15}$$

*Here $B_{k_i}$ is the number of partitions of a set of size $k_i$, i.e., the $k_i$-th Bell number.*

*This can also be explained more intuitively. First, we group the nodes according to the types of the systems' input-output dynamics and denote this partition by $\bar{\mathscr{P}} = \{\bar{\mathscr{P}}_1, \ldots, \bar{\mathscr{P}}_d\}$. Then $k_i, i = 1, \ldots, d$ is the number of nodes in $\mathscr{P}_i$. For the nodes to be in the same part in a viable partition, they must be in the same parts of $\bar{\mathscr{P}}$. Therefore, the allowable partitions can be generated by:*

- *Generate all partitions of sub-networks of $\bar{\mathscr{P}}_i$ and denote the partition set as $\bar{\mathscr{D}}_i$;*
- *Obtain the partitions of the whole network by combining these partitions of sub-networks, i.e., $\bar{\mathscr{D}}_1 \times \bar{\mathscr{D}}_2 \times \cdots \times \bar{\mathscr{D}}_d$.*

*Since $|\bar{\mathscr{D}}_i| = B_{k_i}$, the total number of partitions ($|\bar{\mathscr{D}}_1 \times \bar{\mathscr{D}}_2 \times \cdots \times \bar{\mathscr{D}}_d|$) equals $\prod_{i=1}^{d} B_{k_i}$. Obviously, the following inequality always holds for $d \geq 2$*

$$B_N > \prod_{i=1}^{d} B_{k_i}, \tag{3.16}$$

*where $N = \sum_{i=1}^{d} k_i$. This supports the conclusion that in many cases the improved algorithm generates fewer partitions than the original algorithm. For the special case $v_S = \mathbf{1}$ ($d = 1$), i.e., all systems are identical, the numbers of partitions generated by both algorithms are the same.*

*For $v_S$, similar conclusions can be drawn. Consider a network of identical systems ($v_F = \mathbf{1}$) with an invasive coupling. Suppose $v_S$ has $\tilde{d}$ distinct elements, i.e., the adjacency matrix $A$ has $\tilde{d}$ distinct row-sums. Denote $\tilde{k}_i, i = 1, \ldots, d$, as the numbers of entries with equal values. The number of partitions $\tilde{N}_{\mathscr{P}}$ generated by the new procedure is*

$$\tilde{N}_{\mathscr{P}} = \prod_{i=1}^{\tilde{d}} B_{\tilde{k}_i}. \tag{3.17}$$

*Here $B_{\tilde{k}_i}$ is the number of partitions of a set of size $\tilde{k}_i$, i.e., the $\tilde{k}_i$-th Bell number. In this case, we also have the following inequality holds for $\tilde{d} \geq 2$*

$$B_N > \prod_{i=1}^{d} B_{\tilde{k}_i}, \tag{3.18}$$

*which also indicates that fewer partitions are generated in many cases using the algorithm proposed in this chapter.*

In the following example, we show the improvement on computational efficiency of the proposed method in comparison to the original method.

**Example 3.3.1.** *Consider a network of eight identical systems interacting via an invasive coupling. The adjacency matrix of the network is given by*

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

The vector $v_S$ indicating the (in)equalities of row sums is given by

$$v_S = A\mathbf{1}_8 = \begin{bmatrix} 2 & 4 & 6 & 6 & 11 & 1 & 6 & 1 \end{bmatrix}^\top.$$

Since all the systems are identical, the vector $v_F = \mathbf{1}_8$. Table 3.2 shows the process of partition generation using the improved algorithm. As can be seen from it, there are only 10 partitions generated, while the original algorithm generates $B_8 = 4140$ (the 8th Bell number) partitions.

Recall that the number of possible partitions can be calculated from the structure of $v_S$. Re-arrange the entries of $v_S$ by their values, we have

$$\tilde{v}_S = \begin{bmatrix} 1 & 1 & 2 & 4 & 6 & 6 & 6 & 11 \end{bmatrix}^\top,$$

therefore, the total number of possible partitions equals $B_2 \times B_1 \times B_1 \times B_3 \times B_1 = 10$, where $B_i$ represents the $i$-th Bell number. In fact, the partitions generated by the improved algorithm are a subset of those generated by the original algorithm. Thereby, there are fewer partitions available for performing the row-sum test, thus saving computational effort. Both algorithms are implemented as MATLAB programs. For comparisons, these two MATLAB programs are run on the same laptop with a 2.7 GHz Intel® i7-6820HQ processor and 8 GB DDR4 RAM to compute the partial synchronization manifolds of this example. The computation times are 0.219106 seconds (the original algorithm) and 0.065287 seconds (the improved algorithm), which confirms the efficiency of this improved algorithm.

| System 1 | Systems 1,2 | Systems 1,2,3 | Systems 1,2,3,4 | Systems 1,2,3,4,5 | Systems 1,2,4,5,6 | Systems 1,2,3,4,5,6,7 | Systems 1,2,3,4,5,6,7,8 |
|---|---|---|---|---|---|---|---|
| $d_1 \in \{0\}$ | $d_2 \in \{1\}$ | $d_3 \in \{2\}$ | $d_4 \in \{2,3\}$ | $d_5 \in \{3\}$ | $d_6 \in \{4\}$ | $d_7 \in \{2,5\}$ | $d_8 \in \{4,5\}$ |
| 0 | 01 | 012 | 0122 | 01223 | 012234 | 0122342 | 01223424 |
| | | | | | | | 01223425 |
| | | | | | | | $d_8 \in \{4,6\}$ |
| | | | | | | 0122345 | 01223454 |
| | | | | | | | 01223456 |
| | | | | $d_5 \in \{4\}$ | $d_6 \in \{5\}$ | $d_7 \in \{2,3,6\}$ | $d_8 \in \{5,6\}$ |
| | | | 0123 | 01234 | 012345 | 0123452 | 01234525 |
| | | | | | | | 01234526 |
| | | | | | | | $d_8 \in \{5,6\}$ |
| | | | | | | 0123453 | 01234535 |
| | | | | | | | 01234536 |
| | | | | | | | $d_8 \in \{5,7\}$ |
| | | | | | | 0123456 | 01234565 |
| | | | | | | | 01234567 |

Table 3.2: Generating the set of all partitions of eight systems with row-sum restrictions.

## 3.4 Dynamics decomposition of partially synchronized networks

This section presents a systematic approach to split the dynamics of the network into two parts: 1) dynamics of synchronization error; 2) dynamics of solutions lying on the partial synchronization manifolds. The asymptotic stability of the synchronization error dynamics reveals the stability of the partial synchronization manifolds. That is, when the synchronization error dynamics are stable, the partial synchronization manifolds will be able to attract nearby solutions, and partially synchronous solutions will be robust against small perturbations. So the synchronization error dynamics are important for stability analysis of partial synchronization manifolds, while, the dynamics on the manifolds can be used to study the behaviors of the systems after synchronization.

For simplicity, the systems are assumed to be ordered in a preliminary stage according to a viable partition $\mathscr{P}$ associated with $\Pi$ in $\kappa$ clusters. To have the systems ordered in this way, one can use the ordering matrix $R$ introduced in Section 3.2.1. The state variables are also *re-indexed* such that $x_{i,j}$ denotes the $j$-th state in the $i$-th cluster, that is,

$$\begin{array}{ll} x_{1,1},\ x_{1,2},\ldots,x_{1,\kappa_1} & \text{cluster 1,} \\ x_{2,1},\ x_{2,2},\ldots,x_{2,\kappa_2} & \text{cluster 2,} \\ \vdots & \vdots \\ x_{\kappa,1},\ x_{\kappa,2},\ldots,x_{\kappa,\kappa_\kappa} & \text{cluster } \kappa, \end{array}$$

where $\kappa_i$ is the number of nodes in cluster $i$ and $\sum_{i=1}^{\kappa} \kappa_i = N$.

In order to separate the dynamics on the manifold from the error dynamics, a new set of coordinates is defined

$$z = Hx = \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ \vdots \\ x_{\kappa,1} \\ E_1 \\ E_2 \\ \vdots \\ E_\kappa \end{bmatrix}, \text{ with } E_i = \begin{bmatrix} e_{i,2} \\ \vdots \\ e_{i,\kappa_i} \end{bmatrix} = \begin{bmatrix} x_{i,2} - x_{i,1} \\ \vdots \\ x_{i,\kappa_i} - x_{i,1} \end{bmatrix}, i = 1, \ldots, \kappa. \tag{3.19}$$

By construction, $H \in \mathbb{R}^{nN \times nN}$ is an invertible matrix describing the transformation of the coordinates, and $E_i$ denotes the synchronization errors within cluster $i$. The systems corresponding to $x_{1,1}, x_{2,1}, \ldots, x_{\kappa,1}$ are referred to as the reference systems of each cluster.

Since $\Pi$ is a viable partition, the row-sum of all (off-diagonal) blocks in (3.8) are constant for networks with invasive (non-invasive) couplings. Let $\rho_{ij}$ be the value of the row-sums of the $(i, j)$-th block of matrix $A$, with $i, j \in \{1, \ldots, \kappa\}$ and, in case of non-invasive coupling, $i \neq j$. Additionally, for $i = j$, we define $\rho_{ij} = 0$ in case of non-invasive coupling. Also, the dynamical systems at the nodes in each cluster are the same. In this case, let us denote the dynamics of the nodes in cluster $i$ by $g_i, \tilde{B}_i, \tilde{C}_i$, $i = 1, 2, \ldots, \kappa$. That is, $\tilde{f}_1 = f_1 = f_2 = \cdots = f_{\kappa_1}$, $\tilde{f}_2 = f_{\kappa_1+1} = f_{\kappa_1+1} = \cdots = f_{\kappa_1+\kappa_2}$, $\tilde{B}_1 = B_1 = B_2 = \cdots = B_{\kappa_1}$, $\tilde{C}_1 = C_1 = C_2 = \cdots = C_{\kappa_1}$, and so on.

### 3.4.1 Synchronization error dynamics

In what follows, the synchronization error dynamics for networks with invasive and non-invasive couplings are derived.

#### Networks with invasive coupling

Let

$$\rho_i = \sum_{j=1}^{\kappa} \rho_{ij}, \quad i = 1, \ldots, \kappa$$

denote the row-sums of the adjacency matrix, corresponding to each cluster. Then the error dynamics can be expressed as

$$\begin{bmatrix} \dot{e}_{1,2}(t) \\ \vdots \\ \dot{e}_{1,\kappa_1}(t) \\ \vdots \\ \dot{e}_{\kappa,2}(t) \\ \vdots \\ \dot{e}_{\kappa,\kappa_\kappa}(t) \end{bmatrix} = \begin{bmatrix} \tilde{f}_1(x_{1,1}(t)+e_{1,2}(t)) - \tilde{f}_1(x_{1,1}(t)) \\ \vdots \\ \tilde{f}_1(x_{1,1}(t)+e_{1,\kappa_1}(t)) - \tilde{f}_1(x_{1,1}(t)) \\ \vdots \\ \tilde{f}_\kappa(x_{\kappa,1}(t)+e_{\kappa,2}(t)) - \tilde{f}_\kappa(x_{\kappa,1}(t)) \\ \vdots \\ \tilde{f}_\kappa(x_{\kappa,1}(t)+e_{\kappa,\kappa_\kappa}(t)) - \tilde{f}_\kappa(x_{\kappa,1}(t)) \end{bmatrix} - \begin{bmatrix} k\tilde{B}_1\tilde{C}_1\rho_1 e_{1,2}(t) \\ \vdots \\ k\tilde{B}_1\tilde{C}_1\rho_1 e_{1,\kappa_1}(t) \\ \vdots \\ k\tilde{B}_\kappa\tilde{C}_\kappa\rho_\kappa e_{\kappa,2}(t) \\ \vdots \\ k\tilde{B}_\kappa\tilde{C}_\kappa\rho_\kappa e_{\kappa,\kappa_\kappa}(t) \end{bmatrix} \tag{3.20}$$

$$+ k\tilde{\mathscr{B}}(A_{\text{red}} \otimes I_m)\tilde{\mathscr{C}} \begin{bmatrix} e_{1,2}(t-\tau) \\ \vdots \\ e_{1,\kappa_1}(t-\tau) \\ \vdots \\ e_{\kappa,2}(t-\tau) \\ \vdots \\ e_{\kappa,\kappa_\kappa}(t-\tau) \end{bmatrix},$$

with

$$\tilde{\mathscr{B}} = \text{diag}(I_{\kappa_1-1} \otimes \tilde{B}_1, \ldots, I_{\kappa_\kappa-1} \otimes \tilde{B}_\kappa),$$
$$\tilde{\mathscr{C}} = \text{diag}(I_{\kappa_1-1} \otimes \tilde{C}_1, \ldots, I_{\kappa_\kappa-1} \otimes \tilde{C}_\kappa),$$

and

$$A_{\text{red}} = T_1 A T_1^T - T_2 A T_1^T. \tag{3.21}$$

Here $T_1, T_2 \in \mathbb{R}^{(N-\kappa)\times N}$ are defined as

$$T_1 = \text{diag}(T_{11}, \ldots, T_{1\kappa}), T_2 = \text{diag}(T_{21}, \ldots, T_{2\kappa}),$$

with

$$T_{1i} = \begin{bmatrix} 0 & 1 & \cdots & & 0 \\ \vdots & & \ddots & & \vdots \\ & & & 1 & 0 \\ 0 & & \cdots & 0 & 1 \end{bmatrix}, T_{2i} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{(\kappa_i-1)\times\kappa_i}, i = 1, \ldots, \kappa.$$

The first term in (3.21) corresponds to matrix $A$ where the rows and columns, corresponding to the reference systems (described by $x_{1,1}, \ldots, x_{\kappa,1}$) are taken out. The second term has the following form, which corresponds to matrix $A$ where the columns

corresponding to the reference systems are taken out, and the rows corresponding to the reference systems are repeated for $(\kappa_i - 1)$ times in the rows corresponding to cluster $i$.

$$T_2 A T_1^T = \left[\begin{array}{ccc|ccc|c} a_{1,2} & \cdots & a_{1,\kappa_1} & a_{1,\kappa_1+2} & \cdots & a_{1,\kappa_1+\kappa_2} & \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots \\ a_{1,2} & \cdots & a_{1,\kappa_1} & a_{1,\kappa_1+2} & \cdots & a_{1,\kappa_1+\kappa_2} & \\ \hline & \vdots & & & \vdots & & \vdots \\ \hline a_{N-\kappa_\kappa+1,2} & \cdots & a_{N-\kappa_\kappa+1,\kappa_1} & a_{N-\kappa_\kappa+1,\kappa_1+2} & \cdots & a_{N-\kappa_\kappa+1,\kappa_1+\kappa_2} & \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots \\ a_{N-\kappa_\kappa+1,2} & \cdots & a_{N-\kappa_\kappa+1,\kappa_1} & a_{N-\kappa_\kappa+1,\kappa_1+2} & \cdots & a_{N-\kappa_\kappa+1,\kappa_1+\kappa_2} & \end{array}\right].$$

Note that if $B_i = B$ and $C_i = C$ for all $i = 1, \ldots, N$, the term $k\tilde{\mathscr{B}}(A_{\text{red}} \otimes I_m)\tilde{\mathscr{C}}$ reduces to $kA_{\text{red}} \otimes BC$.

The following result concerns the spectrum of $A_{\text{red}}$. Recall that as mentioned in Section 2.1.2, a spectrum is considered as a multiset (a generalization of set by allowing its elements to appear more than once [59]) in this thesis.

**Theorem 3.4.1.** *Let $\sigma_\Pi(A)$ be the part of the spectrum of $A$, corresponding to the invariant subspace $\ker(I-\Pi)$, i.e., all eigenvalues of $A$ whose corresponding eigenspaces belong to $\ker(I-\Pi)$. Then we have*

$$\sigma(A_{\text{red}}) = \sigma(A) \setminus \sigma_\Pi(A). \tag{3.22}$$

*Proof.* When the set $\mathscr{M}(\Pi)$ is a partial synchronization manifold, $\ker(I-\Pi)$ is a right invariant subspace of $A$, see [3] for details. That is,

$$\ker(I-\Pi) = \Big\{ \sum_{i=1}^{\kappa} c_i v_i | c_i \in \mathbb{R} \Big\},$$

where $v_i$ are the (generalized) eigenvectors of $A$, corresponding to $\sigma_\Pi(A)$.

Using $H$ defined in (3.19), we can construct the matrix $\bar{A}$ as

$$\bar{A} = HAH^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ \mathbf{0} & A_{\text{red}} \end{bmatrix}, \tag{3.23}$$

where $A_{11} \in \mathbb{R}^{\kappa\times\kappa}$, $A_{12} \in \mathbb{R}^{\kappa\times(N-\kappa)}$ and $\mathbf{0} \in \mathbb{R}^{(N-\kappa)\times\kappa}$.

Based on the structure of $\bar{A}$, we have

$$\sigma(A) = \sigma(\bar{A}) = \sigma(A_{11}) \uplus \sigma(A_{\text{red}}). \tag{3.24}$$

Here the symbol $\uplus$ represents the sum or (arithmetic) addition of two multisets. For example, suppose that $\sigma(A_{11}) = \{1,2,2\}$ and $\sigma(A_{\text{red}}) = \{1,3\}$, then $\sigma(A) = \sigma(A_{11}) \uplus \sigma(A_{\text{red}}) = \{1,2,2,1,3\}$. Note that the spectra $\sigma(A), \sigma(\bar{A}), \sigma(A_{11}), \sigma(A_{\text{red}})$ and $\sigma_\Pi(A)$ are multisets.

As $v_i \in \ker(I-\Pi)$, *i.e.*, $v_i = \Pi v_i$, the vectors $w_i = Hv_i$ will have the form

$$w_i = \begin{bmatrix} \bar{w}_i \\ \mathbf{0} \end{bmatrix}, \tag{3.25}$$

where $\bar{w}_i \in \mathbb{R}^\kappa$ and $\mathbf{0} \in \mathbb{R}^{N-\kappa}$, $i = 1,2,\ldots,\kappa$.

If for some $i \in \{1,2,\ldots,\kappa\}$, $v_i$ is an eigenvector of $A$, $w_i$ is an eigenvector of $\bar{A}$, which indicates

$$\bar{A}\begin{bmatrix} \bar{w}_i \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{\text{red}} \end{bmatrix}\begin{bmatrix} \bar{w}_i \\ \mathbf{0} \end{bmatrix} = \lambda_i \begin{bmatrix} \bar{w}_i \\ \mathbf{0} \end{bmatrix}.$$

Therefore,

$$A_{11}\bar{w}_i = \lambda_i \bar{w}_i.$$

A similar argument holds for the generalized eigenvectors of $A$. It follows that

$$\sigma(A_{11}) = \sigma_\Pi(A).$$

Hence, we can conclude (3.22), completing the proof. □

### Networks with non-invasive coupling

When defining a series of vectors $\tilde{\rho}_i \in \mathbb{R}^{\kappa_i - 1}$ with their elements being the row-sums of the rows in $A$ corresponding to $x_{i,2},\ldots,x_{i,\kappa_i}$ for $i = 1,\ldots,\kappa$, the error dynamics of networks with non-invasive couplings can be expressed as

$$\begin{bmatrix} \dot{e}_{1,2}(t) \\ \vdots \\ \dot{e}_{1,\kappa_1}(t) \\ \vdots \\ \dot{e}_{\kappa,2}(t) \\ \vdots \\ \dot{e}_{\kappa,\kappa_\kappa}(t) \end{bmatrix} = \begin{bmatrix} \tilde{f}_1(x_{1,2}(t)) - \tilde{f}_1(x_{1,1}(t)) \\ \vdots \\ \tilde{f}_1(x_{1,\kappa_1}(t)) - \tilde{f}_1(x_{1,1}(t)) \\ \vdots \\ \tilde{f}_\kappa(x_{\kappa,2}(t)) - \tilde{f}_\kappa(x_{\kappa,1}(t)) \\ \vdots \\ \tilde{f}_\kappa(x_{\kappa,\kappa_\kappa}(t)) - \tilde{f}_\kappa(x_{\kappa,1}(t)) \end{bmatrix} - k\tilde{\mathscr{B}}(L_{\text{red}} \otimes I_m)\tilde{\mathscr{C}} \begin{bmatrix} e_{1,2}(t-\tau) \\ \vdots \\ e_{1,\kappa_1}(t-\tau) \\ \vdots \\ e_{\kappa,2}(t-\tau) \\ \vdots \\ e_{\kappa,\kappa_\kappa}(t-\tau) \end{bmatrix}, \tag{3.26}$$

with

$$L_{\text{red}} = \text{diag}\,(\tilde{\rho}_1,\ldots,\tilde{\rho}_\kappa) - A_{\text{red}}. \tag{3.27}$$

Again, if $B_i = B$ and $C_i = C$ for all $i = 1, \ldots, N$, the term $k\tilde{\mathscr{B}}(L_{\text{red}} \otimes I_m)\tilde{\mathscr{C}}$ reduces to $kL_{\text{red}} \otimes BC$.

The following result concerns the spectrum of $L_{\text{red}}$.

**Theorem 3.4.2.** *Let $\sigma_\Pi(L)$ be the part of the spectrum of $L$, corresponding to the invariant subspace $\ker(I-\Pi)$, i.e., all eigenvalues of $L$ whose corresponding eigenspaces belong to $\ker(I-\Pi)$. Then we have*

$$\sigma(L_{\text{red}}) = \sigma(L) \setminus \sigma_\Pi(L). \tag{3.28}$$

*Proof.* Replace $A$, $A_{11}$, $A_{12}$ and $A_{\text{red}}$ with $L$, $L_{11}$, and $L_{\text{red}}$ in the proof of Theorem 3.4.1. □

## 3.4.2 Dynamics on the synchronization manifolds

### Networks with invasive coupling

For a network with invasive coupling, the dynamics on the synchronization manifolds can be described by a network of $\kappa$ coupled systems:

$$\dot{x}_{i,1}(t) = \tilde{f}_i(x_{i,1}(t)) + k\tilde{B}_i \sum_{j=1}^{\kappa} \rho_{ij} \left( \tilde{C}_j x_{j,1}(t-\tau) - \tilde{C}_i x_{i,1}(t) \right), \tag{3.29}$$

where $i = 1, \ldots, \kappa$.

This reduced network can be represented by the graph $\tilde{\mathscr{G}} = (\tilde{\mathscr{V}}, \tilde{\mathscr{E}}, A_{\text{syn}})$, where $|\tilde{\mathscr{V}}| = \kappa$ and $A_{\text{syn}} = (\rho_{ij})_{i,j=1}^{\kappa} \in \mathbb{R}^{\kappa \times \kappa}$.

Note that the "reduced" adjacency matrix $A_{\text{syn}}$ has in general diagonal elements different from zero, implying the presence of self-loops in (3.29).

### Networks with non-invasive coupling

For a network with non-invasive coupling, the dynamics on the synchronization manifolds can be described by a network of $\kappa$ coupled systems:

$$\dot{x}_{i,1}(t) = \tilde{f}_i(x_{i,1}(t)) + k\tilde{B}_i \sum_{j=1,\ j\neq i}^{\kappa} \rho_{ij} \left( \tilde{C}_j x_{j,1}(t-\tau) - \tilde{C}_i x_{i,1}(t-\tau) \right), \tag{3.30}$$

where $i = 1, \ldots, \kappa$.

This reduced network can be represented by the graph $\tilde{\mathscr{G}} = (\tilde{\mathscr{V}}, \tilde{\mathscr{E}}, A_{\text{syn}})$, where $|\tilde{\mathscr{V}}| = \kappa$ and $A_{\text{syn}} = (\rho_{ij})_{i,j=1}^{\kappa} \in \mathbb{R}^{\kappa\times\kappa}$. Note that $\rho_{ij} = 0$ if $i = j$.

It is important to note that this type of coupling may affect the dynamics on the synchronization manifold. Hence, strictly speaking, the terminology "non-invasive" only applies to fully synchronous solutions.

**Remark.** *For a stable partial synchronization manifold, to observe the corresponding partial synchrony in practice, it also requires that* $x_{i,1}(t) \neq x_{j,1}(t), \forall t \geq t_0$, *for every* $i \neq j \in \{1,\ldots,\kappa\}$, *which means the reduced network has no stable partial synchronization manifold.*

### 3.4.3 Software

This method presented above is implemented as a publicly available software package which is able to:

- compute partial synchronization manifolds given a network (adjacency matrix $A$ and coupling type, dynamical system types of each node)
- compute the matrices describing the synchronization error dynamics ($A_{\text{red}}$ or $L_{\text{red}}$) and the dynamics on the manifolds ($A_{\text{syn}}$) given a viable partition.

The second part can easily be achieved by using (3.21) or (3.27), and $A_{\text{syn}} = (\rho_{ij})$.

The input of the software includes the adjacency matrix $A$ of a network. If $A$ contains non-integer elements, a pre-defined tolerance is required for performing the row-sum test. If $A$ only contains integer elements, the calculation in the software can be performed in exact arithmetic. If the adjacency matrix $A$ is decomposable in the way defined in (A.1) (*i.e.*, $A = \omega_1 A_1 + \ldots + \omega_r A_r$, where $\omega_i, i = 1,\ldots,r$ are rationally independent and $A_i, i = 1,\ldots,r$ are adjacency matrices with non-negative integer entries, see Appendix A), and if the input $\{A_1, A_2, \ldots, A_r\}$ is provided, the calculations can also be done in exact arithmetic.

The software is implemented as a MATLAB function and can be downloaded from the following link:

http://twr.cs.kuleuven.be/research/software/delay-control/manifolds/

## 3.5 Numerical example

We present a didactic example of a network with four interconnected systems to show the analysis process presented in the preceding sections. It will be shown in the example that the coupling parameters (the coupling strength and time-delay) impact the stability of the partial synchronization manifolds.

The systems are of the following form:

$$\begin{aligned} \dot{x}_i(t) &= A_0 x_i(t) + B u_i(t), \\ y_i(t) &= C x_i(t), \end{aligned} \tag{3.31}$$

with

$$A_0 = \begin{bmatrix} -\varepsilon & 1 \\ -\Omega^2 & -\varepsilon \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \Omega, \varepsilon \in \mathbb{R}_{\geq 0} \text{ and } i = 1,2,3,4.$$

Here $u_i$ describes the couplings between the systems:

$$u_i = k \sum_{j \in \mathcal{N}_i} a_{ij}[y_j(t-\tau) - y_i(t-\tau)], \tag{3.32}$$

where $a_{ij}$ are the elements of the adjacency matrix $A$ defined as below

$$A = \begin{bmatrix} 0 & 0.1 & 2 & 1 \\ 0.1 & 0 & 1 & 2 \\ 4 & 2 & 0 & 0.3 \\ 3 & 3 & 0.3 & 0 \end{bmatrix}, \tag{3.33}$$

and where the coupling strength $k$ and the time delay $\tau$ are considered as parameters. Since the network only contains 4 nodes that host the same linear system, it is possible to characterize and analyze the partial manifolds of the network analytically.
Combining the systems of all the nodes, we have

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = I \otimes A_0 \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} - kL \otimes BC \begin{bmatrix} x_1(t-\tau) \\ x_2(t-\tau) \\ x_3(t-\tau) \\ x_4(t-\tau) \end{bmatrix}. \tag{3.34}$$

Here $L$ is the weighted Laplacian matrix, defined by

$$L := D - A = \begin{bmatrix} 3.1 & -0.1 & -2 & -1 \\ -0.1 & -3.1 & -1 & -2 \\ -4 & -2 & -6.3 & -0.3 \\ -3 & -3 & -0.3 & 6.3 \end{bmatrix}, \tag{3.35}$$

where

$$D = \begin{bmatrix} \sum_{j=1}^{N} a_{1j} & & 0 \\ & \ddots & \\ 0 & & \sum_{j=1}^{N} a_{Nj} \end{bmatrix} = \begin{bmatrix} 3.1 & 0 & 0 & 0 \\ 0 & 3.2 & 0 & 0 \\ 0 & 0 & 6.3 & 0 \\ 0 & 0 & 0 & 6.3 \end{bmatrix}.$$

The eigenvectors and eigenvalues of $L$ are

$$V = \begin{bmatrix} 0.50 & 0.56 & -0.32 & 0.15 \\ 0.50 & -0.79 & -0.32 & -0.22 \\ 0.50 & 0.18 & 0.63 & -0.63 \\ 0.50 & -0.19 & 0.63 & 0.73 \end{bmatrix}, \sigma(L) = \{0, 2.93, 9.00, 6.87\}. \tag{3.36}$$

First, we explain how to compute the stability regions of (3.31)-(3.32) in the $(k,\tau)$ parameter space. The characteristic function of (3.34) is given by

$$\gamma(\lambda; k, \tau) = \det(I \otimes (\lambda I - A_0) + L \otimes kBCe^{-\lambda\tau}). \tag{3.37}$$

As in [29], this function can be factorized as

$$\gamma(\lambda; k, \tau) = \prod_{i=1}^{4} \gamma_i(\lambda; k, \tau), \tag{3.38}$$

with

$$\gamma_i(\lambda; k, \tau) = \det(\lambda I - A_0 + kBC\lambda_i(L)e^{-\lambda\tau}), i = 1, 2, 3, 4. \tag{3.39}$$

Denoting $\tilde{k} = \lambda_i(L)k$, each corresponding characteristic equation is of the following form

$$(\lambda + \varepsilon)^2 + \Omega^2 + \tilde{k}e^{-\lambda\tau} = 0. \tag{3.40}$$

To analyze the stability of (3.34) at the equilibrium, we substitute $\lambda = j\omega, \omega > 0$, into (3.40)

$$-\omega^2 + \varepsilon^2 + 2j\varepsilon\omega + \Omega^2 = -\tilde{k}e^{-j\omega\tau}. \tag{3.41}$$

Equating the module and phase of the left side with the right side of (3.41), we have

$$(-\omega^2 + \varepsilon^2 + \Omega^2)^2 + (2\varepsilon\omega)^2 = \tilde{k}^2, \tag{3.42}$$

$$\arctan\left(\frac{2\varepsilon\omega}{-\omega^2 + \varepsilon^2 + \Omega^2}\right) + 2\pi l = -\omega\tau, l \in \mathbb{Z}. \tag{3.43}$$

From these two equations, we can express $\tilde{k}$ and $\tau$ as functions of $\omega$, *i.e.*, $\tilde{k} := \tilde{k}(\omega)$ and $\tau := \tau(\omega)$. Therefore, by sweeping $\omega$, we can depict the crossing curves of the time-delay system numerically in the parameter space $(\tau, \tilde{k})$. Figure 3.1 shows the stability crossing curves of the system with $\Omega = 1$ and $\varepsilon = 0.02$. These values are used in the remainder of the example. The stability region is marked with the symbol "S".

Note that for $\omega = 0$, we have $\tilde{k} = -(\varepsilon^2 + \Omega^2)$.
Next, the partial manifolds of this system is analyzed by using the procedure and software presented in the preceding section. Using the software developed in this work, we can easily identify a viable partition $\mathscr{P} = \{\{1,2\},\{3,4\}\}$ and derive the separated dynamics.
The synchronization error dynamics are described by

$$\begin{bmatrix} \dot{E}_1(t) \\ \dot{E}_2(t) \end{bmatrix} = \begin{bmatrix} A_0 & 0 \\ 0 & A_0 \end{bmatrix} \begin{bmatrix} E_1(t) \\ E_2(t) \end{bmatrix} - kL_{\text{red}} \otimes BC \begin{bmatrix} E_1(t-\tau) \\ E_2(t-\tau) \end{bmatrix}, \tag{3.44}$$

with

$$L_{\text{red}} = \begin{bmatrix} 3.2 & -1 \\ -1 & 6.6 \end{bmatrix}, \sigma(L_{\text{red}}) = \{2.93, 6.87\}. \tag{3.45}$$

We can see that $\sigma(L_{\text{red}}) \subset \sigma(L)$.
By comparing (3.34) and (3.44), we observe that they are of the same form (but different dimensions). Therefore, the factorized characteristic function is

$$\prod_{i=1}^{2} \gamma_i(\lambda; k, \tau) = \prod_{i=1}^{2} \det(\lambda I - A_0 + kBC\lambda_i(L_{\text{red}})e^{-\lambda\tau}), \tag{3.46}$$

which is part of (3.39).
The dynamics on the manifold are given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} A_0 & 0 \\ 0 & A_0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} kBC\rho_{12}(x_3(t-\tau) - x_1(t-\tau)) \\ kBC\rho_{21}(x_1(t-\tau) - x_3(t-\tau)) \end{bmatrix}, \tag{3.47}$$

*i.e.*,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} A_0 & 0 \\ 0 & A_0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_3(t) \end{bmatrix} - kL_{\text{syn}} \otimes BC \begin{bmatrix} x_1(t-\tau) \\ x_3(t-\tau) \end{bmatrix}, \tag{3.48}$$

where

$$L_{\text{syn}} = \begin{bmatrix} \rho_{12} & -\rho_{12} \\ -\rho_{21} & \rho_{21} \end{bmatrix} = \begin{bmatrix} 3 & -3 \\ -6 & 6 \end{bmatrix}, \sigma(L_{\text{syn}}) = \{0, 9\}. \tag{3.49}$$

We can see that $\sigma(L_{\text{red}}) = \sigma(L) \setminus \sigma(L_{\text{syn}})$, which verifies Theorem 3.4.2. Again, the factorized characteristic function of (3.48) shown below is part of (3.39):

$$\prod_{i=1}^{2} \gamma_i(\lambda; k, \tau) = \prod_{i=1}^{2} \det(\lambda I - A_0 + kBC\lambda_i(L_{\text{syn}})e^{-\lambda\tau}). \tag{3.50}$$

A comparison between (3.36) on one hand, and (3.45), (3.49) on the other hand, yields additional information that the eigenvalues $\lambda_1(L) = 0$ and $\lambda_3(L) = 9$ are related to the dynamics on the manifolds, and that the eigenvalues $\lambda_2(L) = 2.93$ and $\lambda_4(L) = 6.87$ are related to the synchronization errors dynamics. Note that the decomposition can

already be observed from the patterns of the eigenvectors in (3.36), see also Theorem 2 in [47].

However, when the network becomes complicated (*e.g.*, increased size), analysis by hand may become extremely difficult. In this case, the procedure proposed in this chapter will show its advantage, as it is automatic and more systematic.

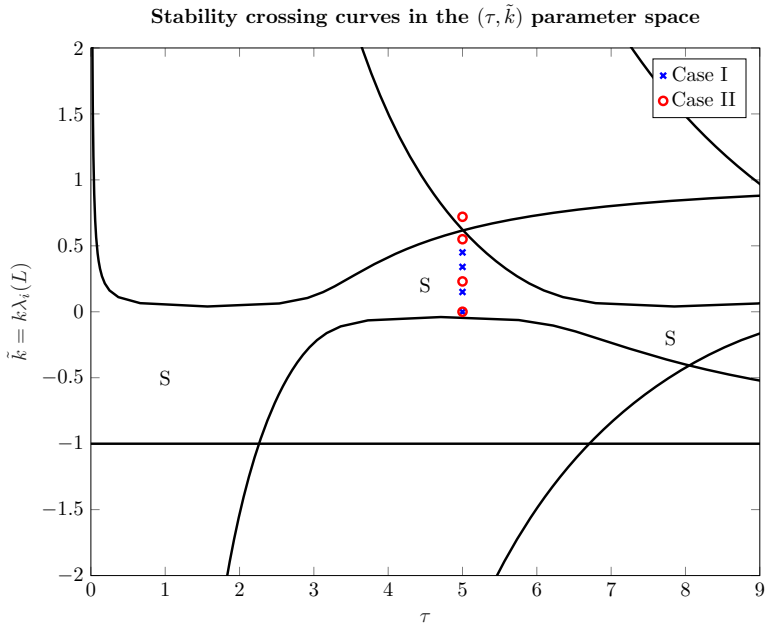



Figure 3.1: Stability regions of (3.31)-(3.32) with $\Omega = 1$ and $\varepsilon = 0.02$.

Now, we check the stability crossing curve of the time-delay system described by (3.41). From Figure 3.1, It can be seen that the parameters $k$ and $\tau$ impact the stability of the partial synchronization manifolds. To illustrate the impact, the two following cases are considered:

- Case I with $\tau = 5.0$ and $k = 0.05$;
- Case II with $\tau = 5.0$ and $k = 0.08$.

For Case I, we have $[\tilde{k}_1\ \tilde{k}_2\ \tilde{k}_3\ \tilde{k}_4] = [0\ 0.15\ 0.45\ 0.34]$. The parameter pairs $(\tau, \tilde{k}_i)$ are shown in Figure 3.1. All the points are inside the stability region. Therefore, all the dynamics are stable.

For Case II, we have $[\tilde{k}_1\ \tilde{k}_2\ \tilde{k}_3\ \tilde{k}_4] = [0\ 0.23\ 0.75\ 0.55]$. The parameter pairs $(\tau, \tilde{k}_i)$ are shown in Figure 3.1. As we can see from it, the pair $(\tau, \tilde{k}_3)$ is outside of the stability region while the other three pairs are inside the stability region. From the analysis above, we know $\tilde{k}_3$ is related to the dynamics on the manifold. Therefore, the partial synchronization manifold is stable, while the dynamics on the manifold are unstable. Figure 3.2 shows the simulation results of the network. Note that the differences between $y_1$ and $y_2$, respectively, between $y_3$ and $y_4$ are invisible after some time.

Finally, we also add a nonlinear term in each system.

$$\begin{aligned} \dot{x}_i(t) &= A_0 x_i(t) + B u_i(t) + h(x_i(t)), \\ y_i(t) &= C x_i(t), \end{aligned} \tag{3.51}$$

where

$$h(x_i(t)) = \begin{bmatrix} -v_i(t)(v_i^2(t) + w_i^2(t)) \\ -w_i(t)(v_i^2(t) + w_i^2(t)) \end{bmatrix}, \tag{3.52}$$

and

$$x_i(t) = \begin{bmatrix} v_i(t) \\ w_i(t) \end{bmatrix}, v_i, w_i \in \mathbb{R}.$$

The effect induced by the nonlinear term is that after the stability of the synchronized equilibrium is lost by increasing $k$, the partially synchronous solutions converge to a limit cycle. The latter is shown in Figure 3.3. Note that when the systems contain nonlinearities, the method using the frequency domain technique as above to analyze the stability of partial synchronization manifolds is not applicable. We present another method to derive conditions for stability of partial synchronization manifolds in networks of general (nonlinear) systems in Chapter 4.

It is worth mentioning that the matrix (3.35) reveals that the coupling of the nodes within clusters is weak and that the coupling of the nodes between clusters is strong. In this sense, it might be surprising that, when $k$ is increased, the dynamics on the manifold associated with $\mathscr{P} = \{\{1,2\},\{3,4\}\}$ become unstable first, instead of breaking the synchrony within the clusters. However, this can be explained by the presence of the time delay and the sensitivity of high gain feedback with respect to it.

## 3.6 Conclusions

In this chapter, a method to identify all partial synchronization manifolds of delay-coupled systems is presented, which is implemented in the form of a MATLAB function. First, the method from [3] is extended to allow the systems in the networks to be different and improve the computational efficiency for the invasive coupling case. Second, the dynamics of the network is split into two parts: 1) synchronization

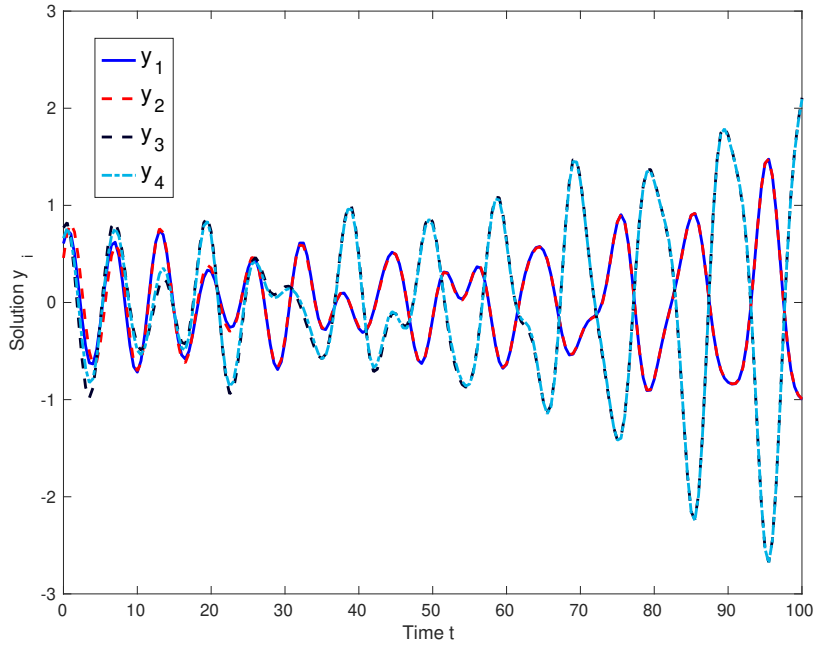



Figure 3.2: Simulation results of the network of coupled linear systems (3.31), (3.32) with $\tau = 5.00$ and $k = 0.08$

error dynamics; 2) dynamics on manifolds. The stability of the synchronization error dynamics is a direct indicator of the stability of partial synchronization manifolds. The relations between the spectrum of the separated dynamics and that of the original networks is also specified. Finally, to demonstrate the proposed procedure for the analysis of partial synchronization manifolds, a didactic example is presented. Using the procedure along with the software, the dynamics of the network can be systematically separated, which can be used for stability analysis of the manifolds.

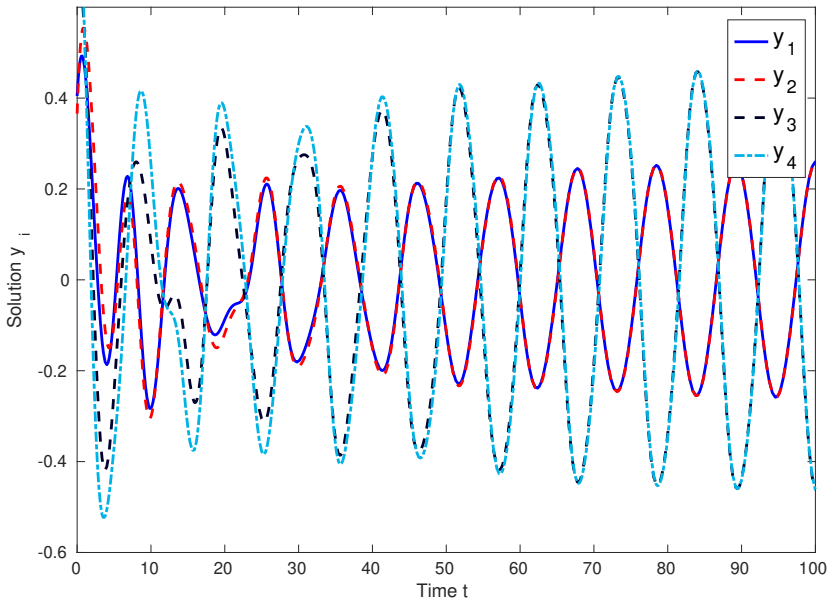



Figure 3.3: Simulation results of the network of coupled nonlinear systms (3.51), (3.32) with $\tau = 5.00$ and $k = 0.08$

# Chapter 4

# Partial synchronization in networks of delay-coupled systems

**Abstract:** In this chapter, sufficient conditions for partial synchronization in networks of delay-coupled systems are presented. The conditions, derived by using a Lyapunov-Krasovskii approach, guarantee the stability of partial synchronization manifolds. First, the synchronization error dynamics are isolated from the network dynamics using the method presented in Chapter 3. Second, we use a parameter-dependent Lyapunov-Krasovskii functional to assess the local stability of partial synchronization manifolds, by employing techniques originally developed for linear parameter-varying (LPV) time-delay systems. The stability conditions are formulated in the form of linear matrix inequalities (LMIs) which can be solved by several existing tools. The results presented in this chapter are published in [84].

## 4.1 Introduction

This chapter focuses on partial synchronization in networks of systems interconnected via linear diffusive time-delay couplings. The synchronization is said to be exact when the states of the synchronized systems converge to each other, *i.e.*, the synchronization errors converge to zero. In the next chapter, synchronization in a practical sense will be considered where the states of the synchronized systems are not completely identical but remain within some distance from each other. Networks may consist of different

types of dynamical systems. More precisely, it is allowed for some (but not all) systems to have different dynamics. The diffusive time-delay couplings can be *invasive* or *non-invasive*. As in Chapter 3, partial synchronization manifolds are used to describe patterns of partial synchronization. The sufficient conditions for partial synchronization will be obtained via the stability analysis of such manifolds. To analyze the stability of such manifolds, firstly, the synchronization error dynamics are isolated from the network dynamics and then linearized around the zero equilibrium solution; secondly, such dynamics are interpreted in terms of a linear parameter-varying (LPV) time-delay system. In this way, by assessing the stability of the LPV time-delay system, we can derive sufficient conditions for the local stability of partial synchronization manifolds. Several papers have been devoted to the stability analysis of LPV time-delay systems, see [85, 86, 87]. The stability of such systems is assessed by using the Lyapunov-Krasovskii theorem with a quadratic Lyapunov-Krasovskii functional candidate. The conditions for positivity of the functional and negative-definiteness of its derivative along the solutions can be expressed in terms of LMIs. By choosing appropriate Lyapunov functionals and techniques for the LMIs, the conservatism can be reduced, see [86, 88]. Although similar LMI conditions deduced from parameter-independent Lyapunov functionals have been used for the stability analysis of full synchronization (see [27, 89, 28]), few works exploit this framework for the partial synchronization case. Therefore, we will use such a method based on a delay- and parameter-dependent Lyapunov functional for the stability analysis of partial synchronization manifolds here.

The structure of this chapter is as follows. Section 4.2 reintroduces the network settings and the relevant concepts, including the systems, couplings, and partial synchronization manifolds, which have been studied in Chapter 3. Section 4.3 addresses the separation of the synchronization error dynamics from the network dynamics by using the method presented in Chapter 3. In Section 4.4, the local stability conditions of the synchronization error dynamics inferred from a Lyapunov-Krasovskii functional are derived. Section 4.5 presents an example where the results of this chapter are applied to a network of Hindmarsh-Rose neuron models. Finally, Section 4.6 provides the conclusions.

## 4.2 Networks

In this section, we introduce the networks of delay-coupled systems under consideration, by adopting the settings in Chapter 3. Besides, the concept of and existence conditions for partial synchronization manifolds are recalled.

The networks are represented by simple, strongly connected directed graphs $\mathscr{G} = (\mathscr{V}, \mathscr{E}, A)$, where

- $\mathscr{V} = \{1, 2, \ldots, N\}$ is a finite set of nodes with cardinality $|\mathscr{V}| = N$;
- $\mathscr{E} \subset \mathscr{V} \times \mathscr{V}$ is the set of *directed edges*, where the edge $(i, j)$ points from node $i$ to node $j$;
- $A = \left(a_{ij}\right) \in \mathbb{R}^{N \times N}$ is the weighted adjacency matrix, where $a_{ij} > 0$ represents the weight of edge $(j, i)$ when $(j, i) \in \mathscr{E}$, and $a_{ij} = 0$ when $(j, i) \notin \mathscr{E}$.

Every node in the network hosts a time-invariant dynamical system of the form

$$\begin{cases} \dot{x}_i(t) = f_i(x_i(t)) + B_i u_i(t), \\ y_i(t) = C_i x_i(t), \end{cases} \tag{4.1}$$

where $x_i(t) \in \mathbb{R}^n$ are the states, $f_i : \mathbb{R}^n \to \mathbb{R}^n$ are sufficiently smooth functions, $u_i(t) \in \mathbb{R}^m$ are the inputs, $y_i(t) \in \mathbb{R}^m$ are the outputs, $B_i \in \mathbb{R}^{n \times m}$ are the input matrices, $C_i \in \mathbb{R}^{m \times n}$ are the output matrices, and $i = 1, \ldots, N$. Here, $f_i$, $B_i$ and $C_i$ can vary for different nodes, that is, in this setting, it is allowed for some systems to have different input-output dynamics. In addition, $C_i B_i$ are assumed to be similar to positive definite matrices to guarantee the left-invertibility of the systems.

The systems (4.1) interact via either one of the following two types of diffusive couplings:

$$\text{invasive coupling } u_i(t) = k \sum_{j \in \mathscr{N}_i} a_{ij} [y_j(t - \tau) - y_i(t)], \tag{4.2}$$

or

$$\text{non-invasive coupling } u_i(t) = k \sum_{j \in \mathscr{N}_i} a_{ij} [y_j(t - \tau) - y_i(t - \tau)], \tag{4.3}$$

where $\mathscr{N}_i$ is the neighboring set of node $i$, defined as $\mathscr{N}_i := \{ j \in \mathscr{V} \,|\, (i, j) \in \mathscr{E} \}$, and $\tau$, $k$ are, respectively, the time-delay and coupling strength.

We have the following assumption on the coupled systems (4.1), (4.2) and (4.1), (4.3).

**Assumption 4.2.1.** *The solutions of the coupled systems* (4.1)*,* (4.2) *and* (4.1)*,* (4.3) *are ultimately bounded.*

The boundedness is then used during the derivation of the stability conditions of the partial synchronization manifolds in Section 4.4.

For the systems (4.1) coupled via invasive coupling (4.2), Assumption 4.2.1 is guaranteed if the systems satisfy the following assumption.

**Assumption 4.2.2.** *Systems* (4.1) *are strictly* $\mathscr{C}^1$*-semipassive with a radially unbounded storage function* $V$*.*

When the systems (4.1) are strictly $\mathscr{C}^1$-semipassive, they have radially unbounded storage functions $V_1(x_1)$, $V_2(x_2)$, …, $V_N(x_N)$, satisfying the semipassivity inequality (2.2). Then, networks of such systems interconnected via the invasive coupling (4.2) have ultimately bounded solutions [63]. There exists a radially unbounded nonnegative function $\mathscr{V}(x_1,x_2,\ldots,x_N)=\sum_{i=1}^N \vartheta_i V_i(x_i)+\frac{k}{2}\sum_{i=1}^N\sum_{j\in\mathscr{N}_i}\vartheta_i a_{ij}\int_\tau^0 \|y_j(t+s)\|^2 \mathrm{d}s$, where $\vartheta_i$ is the $i$-th entry of the vector $\vartheta\in\mathbb{R}^N$ with all entries positive and $\vartheta^\top L=0$, and a constant $\Upsilon^*\geq 0$ such that $\dot{\mathscr{V}}(x_1,x_2,\ldots,x_N)<0$, for any $\Upsilon\geq\Upsilon^*$ and all possible $(x_1,x_2,\ldots,x_N)$ subject to $\mathscr{V}(x_1,x_2,\ldots,x_N)\geq\Upsilon$. See [63] for more details. More precisely, the set $\{x\in\mathscr{C}([-\tau,0],\mathbb{R}^{Nn})|\mathscr{V}(x_1,x_2,\ldots,x_N)\leq\Upsilon\}$ is a forward invariant set for the dynamics (4.1) coupled via (4.2), and all solutions converge to this set in finite time.

For the systems (4.1) coupled via invasive coupling (4.2), similar results hold, but an additional restriction on coupling strength is needed, see [1].

For the coupled systems (4.1), (4.2) or (4.1), (4.3), a solution is a partially synchronous solution if there exist $i,j\in\mathscr{V}$ with $i\neq j$ such that

$$x_i(t)=x_j(t),\quad \forall t\geq t_0, \tag{4.4}$$

whenever $x_i(t)=x_j(t)$ for $t\in[t_0-\tau,\ t_0]$.

According to (4.4), the nodes can be grouped into clusters. To describe the clustering of the nodes, a natural way is to use the concept of partition which is parameterized by an $N\times N$ permutation matrix $\Pi$, as in Section 3.2.

Denoting $x_t\in\mathscr{C}([-\tau,0],\mathbb{R}^{Nn})$ as the state of the network with

$$x_t(\theta):=\begin{bmatrix}x_1(t+\theta)\\ x_2(t+\theta)\\ \vdots\\ x_N(t+\theta)\end{bmatrix},\quad \theta\in[-\tau,0],$$

the conditions of the form (4.4) can be expressed as $x_t\in\mathscr{M}(\Pi),\forall t\geq t_0$, where

$$\mathscr{M}(\Pi):=\{\phi\in\mathscr{C}([-\tau,0],\mathbb{R}^{Nn})\,|\,\phi(\theta)=\mathrm{col}(\phi_1(\theta),\ldots,\phi_N(\theta)),$$
$$\phi_i(\theta)\in\mathbb{R}^n,\ i=1,\ldots,N,\ \phi(\theta)\in\ker(I_{Nn}-\Pi\otimes I_n),\ \forall\theta\in[-\tau,0]\}$$

is the set of partially synchronous states induced by the permutation matrix $\Pi$.

The set $\mathscr{M}(\Pi)$ is a partial synchronization manifold for the coupled systems (4.1), (4.2), or (4.1), (4.3), if and only if it is positively invariant under the dynamics (4.1), (4.2), or (4.1), (4.3), respectively, see Definition 3.2.2. If the set $\mathscr{M}(\Pi)$ is a partial synchronization manifold, the partition $\mathscr{P}$ associated with $\Pi$ is called *viable*. The viability of a partition (the existence of a partial synchronization manifold) can be checked by the conditions in Theorem 3.2.3.

## 4.3 Synchronization error dynamics

In this section, we show how to separate the synchronization error dynamics from the network dynamics, using the procedure presented in Chapter 3. The synchronization error dynamics are linearized and will be used for analyzing the local stability of the partial synchronization manifolds.

As in Chapter 3, the systems are assumed to have been pre-ordered by clusters according to a viable partition $\mathscr{P}$ associated with $\Pi$ as follows

$$\begin{array}{cc} x_{1,1},\, x_{1,2},\ldots,x_{1,\kappa_1} & \text{cluster 1,} \\ x_{2,1},\, x_{2,2},\ldots,x_{2,\kappa_2} & \text{cluster 2,} \\ \vdots & \vdots \\ x_{\kappa,1},\, x_{\kappa,2},\ldots,x_{\kappa,\kappa_\kappa} & \text{cluster } \kappa, \end{array}$$

where $\kappa_i$ is the number of nodes in cluster $i$ with $\sum_{i=1}^{\kappa}\kappa_i = N$. Here, $x_{1,1},\ldots,x_{\kappa,1}$ are referred to as the reference systems of the clusters. Now, we denote the synchronization errors by

$$E_i = \begin{bmatrix} e_{i,2} \\ \vdots \\ e_{i,\kappa_i} \end{bmatrix} = \begin{bmatrix} x_{i,2} - x_{i,1} \\ \vdots \\ x_{i,\kappa_i} - x_{i,1} \end{bmatrix}, i = 1,\ldots,\kappa. \tag{4.5}$$

We also denote $\rho_{ij}$ as the value of the row sums of the $ij^{\text{th}}$ block of the adjacency matrix for $i,j \in \{1,\ldots,\kappa\}$ (in case of non-invasive coupling, $i \neq j$). Note that these blocks have constant row-sums since $\mathscr{P}$ is viable. In the case of non-invasive coupling, we define $\rho_{ij} = 0$ when $i = j$. Recall that the nodes in each cluster host the same dynamical systems. Therefore, we denote the dynamics of the nodes in cluster $i$ by $\tilde{f}_i, \tilde{B}_i, \tilde{C}_i$, $i = 1,2,\ldots,\kappa$, where $\tilde{f}_1 := f_1 = f_2 = \cdots = f_{\kappa_1}$, $\tilde{f}_2 := f_{\kappa_1+1} = f_{\kappa_1+1} = \cdots = f_{\kappa_1+\kappa_2}$, $\tilde{B}_1 := B_1 = B_2 = \cdots = B_{\kappa_1}$, $\tilde{C}_1 := C_1 = C_2 = \cdots = C_{\kappa_1}$, and so on.

### 4.3.1 Networks with invasive coupling

In what follows the linearized synchronization error dynamics of the networks with invasive couplings will be derived. First, we denote the row sums of the adjacency matrix, corresponding to the reference systems by

$$\rho_i = \sum_{j=1}^{\kappa} \rho_{ij}, \;\; i = 1,\ldots,\kappa.$$

According to Section 3.4, the synchronization error dynamics can be presented as a time-delay system (3.20) with a zero equilibrium $E_i = 0$, $i = 1,\ldots,\kappa$.

Letting $\delta E_i$ be the small variation of $E_i$ from zero, the linearized error dynamics of networks with invasive couplings (around the zero equilibrium) can be expressed as

$$\frac{d}{dt}\begin{bmatrix} \delta E_1(t) \\ \vdots \\ \delta E_\kappa(t) \end{bmatrix} = A_{I0}\begin{bmatrix} \delta E_1(t) \\ \vdots \\ \delta E_\kappa(t) \end{bmatrix} + A_{I1}\begin{bmatrix} \delta E_1(t-\tau) \\ \vdots \\ \delta E_\kappa(t-\tau) \end{bmatrix}, \quad (4.6)$$

with

$$A_{I0} = \begin{bmatrix} I_{\kappa_1-1}\otimes(\frac{\partial \tilde{f}_1}{\partial x}(x_{1,1}(t)) - k\rho_1\tilde{B}_1\tilde{C}_1) & 0 \\ & \ddots \\ 0 & I_{\kappa_\kappa-1}\otimes(\frac{\partial \tilde{f}_\kappa}{\partial x}(x_{\kappa,1}(t)) - k\rho_\kappa\tilde{B}_\kappa\tilde{C}_\kappa) \end{bmatrix} \quad (4.7)$$

$$A_{I1} = k\tilde{\mathscr{B}}(A_{\text{red}}\otimes I_m)\tilde{\mathscr{C}}, \quad (4.8)$$

$$\tilde{\mathscr{B}} = \text{diag}(I_{\kappa_1-1}\otimes\tilde{B}_1,\ldots,I_{\kappa_\kappa-1}\otimes\tilde{B}_\kappa),$$

$$\tilde{\mathscr{C}} = \text{diag}(I_{\kappa_1-1}\otimes\tilde{C}_1,\ldots,I_{\kappa_\kappa-1}\otimes\tilde{C}_\kappa),$$

and

$$A_{\text{red}} = T_1AT_1^T - T_2AT_1^T, \quad (4.9)$$

where $T_1, T_2 \in \mathbb{R}^{(N-\kappa)\times N}$ are defined as

$$T_1 = \text{diag}(T_{11},\ldots,T_{1\kappa}), T_2 = \text{diag}(T_{21},\ldots,T_{2\kappa}),$$

with

$$T_{1i} = \begin{bmatrix} 0 & 1 & \cdots & & 0 \\ \vdots & & \ddots & & \vdots \\ & & & 1 & 0 \\ 0 & & \cdots & 0 & 1 \end{bmatrix}, T_{2i} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{(\kappa_i-1)\times\kappa_i}.$$

For more information on the derivation and a detailed interpretation of $T_1AT_1^\top$ and $T_2AT_1^\top$, we refer to Section 3.4.

### 4.3.2 Networks with non-invasive coupling

Here, the linearized synchronization error dynamics of the networks with non-invasive couplings are derived. First, we define a series of vectors $\tilde{\rho}_i \in \mathbb{R}^{\kappa_i-1}$ with their entries being the row sums of the rows in $A$ corresponding to $x_{i,2},\ldots,x_{i,\kappa_i}$ for $i = 1,\ldots,\kappa$. According to Section 3.4, the synchronization error dynamics are given by (3.26). Then,

the linearized error dynamics of networks with non-invasive couplings (around the zero equilibrium) can be expressed as

$$\frac{d}{dt}\begin{bmatrix} \delta E_1(t) \\ \vdots \\ \delta E_\kappa(t) \end{bmatrix} = A_{N0}\begin{bmatrix} \delta E_1(t) \\ \vdots \\ \delta E_\kappa(t) \end{bmatrix} + A_{N1}\begin{bmatrix} \delta E_1(t-\tau) \\ \vdots \\ \delta E_\kappa(t-\tau) \end{bmatrix}, \quad (4.10)$$

with

$$A_{N0} = \begin{bmatrix} I_{\kappa_1-1} \otimes \frac{\partial \tilde{f}_1}{\partial x}(x_{1,1}(t)) & & 0 \\ & \ddots & \\ 0 & & I_{\kappa_\kappa-1} \otimes \frac{\partial \tilde{f}_\kappa}{\partial x}(x_{\kappa,1}(t)) \end{bmatrix} \quad (4.11)$$

$$A_{N1} = -k\tilde{\mathscr{B}}(L_{\text{red}} \otimes I_m)\tilde{\mathscr{C}}, \quad (4.12)$$

$$L_{\text{red}} = \text{diag}\,(\tilde{\rho}_1, \ldots, \tilde{\rho}_\kappa) - A_{\text{red}}. \quad (4.13)$$

## 4.4 Conditions for partial synchronization

In this section, sufficient conditions for partial synchronization, *i.e.*, the stability conditions for the partial synchronization manifolds are presented. These conditions are derived by analyzing the stability of the synchronization error systems (4.6) or (4.10). First, it is shown that the error systems can be interpreted as an LPV time-delay system. Second, the stability conditions for such systems, implying the stability conditions for the partial synchronization manifolds, are derived by using the Lyapunov method and formulated in the form of LMIs.

By comparing the structure of (4.6) with that of (4.10), we notice that these two error systems (4.6) and (4.10) can be re-written in a unified form as

$$\dot{\mathscr{X}}(t) = \mathscr{A}(\eta(t))\mathscr{X}(t) + \mathscr{A}_d\mathscr{X}(t-\tau), \quad (4.14)$$

where

$$\mathscr{X}(t) = \begin{bmatrix} E_1^\top(t) & \cdots & E_\kappa^\top(t) \end{bmatrix}^\top,$$

$$\mathscr{A}(\eta(t)) = \begin{cases} A_{I0} & \text{invasive coupling,} \\ A_{N0} & \text{non-invasive coupling,} \end{cases}$$

$$\eta(t) = [x_{1,1}(t), \ldots, x_{\kappa,1}(t)],$$

$$\mathscr{A}_d = \begin{cases} A_{I1} & \text{invasive coupling,} \\ A_{N1} & \text{non-invasive coupling.} \end{cases}$$

In this way, the error system can be interpreted as an LPV time-delay system, whose dynamics depend on exogenous non-stationary parameters $\eta(t)$. Then the synchronization problem of the delay-coupled networks can be reformulated as the stability problem of LPV time-delay systems. Due to the time-varying nature of the LPV systems, the existing results based on the frequency-domain method or eigenvalue-based analysis techniques cannot be applied for the analysis of this class of systems. Alternatively, we resort to the Lyapunov method to analyze the stability of the underlying systems. In particular, some sufficient delay-dependent conditions to check the partial synchronization problems can be obtained in what follows.

Recall that under Assumption 4.2.1, the solutions are bounded. That is, the parameters $x_{i,1}(t)$, $i = 1,2,\cdots,\kappa$ range between some extremal values $\underline{\eta}_i$ and $\bar{\eta}_i$, i.e., $x_{i,1}(t) \in [\underline{\eta}_i, \bar{\eta}_i]$. The boundedness of $x_i(t)$ also leads to the derivative $\dot{x}_i(t)$ to be confined as $\dot{x}_{i,1}(t) \in [\underline{\nu}_i, \bar{\nu}_i]$, $i = 1,2,\cdots,\kappa$, where $\underline{\nu}_i$ and $\bar{\nu}_i$ are, respectively, the lower and upper bound on $\dot{x}_{i,1}(t)$. The latter can be readily derived from $\dot{\mathscr{X}}(t) = \mathscr{A}(\eta(t))\mathscr{X}(t) + \mathscr{A}_d\mathscr{X}(t-\tau)$. Specifically, if $\|\mathscr{X}(t)\| \le \eta_o$ for all $t \ge 0$, it follows that $\|\dot{\mathscr{X}}(t)\| \le \max_{\{\|\mathscr{X}(t)\|\le\eta_o, \|\mathscr{X}(t-\tau)\|\le\eta_o\}} \|\mathscr{A}(\eta(t))\mathscr{X}(t) + \mathscr{A}_d\mathscr{X}(t-\tau)\|$ holds. However, as $\|\mathscr{A}(\eta(t))\|$ and $\|\mathscr{A}_d\|$ also depend on the coupling parameter $k$, the bounds of $\dot{x}_{i,1}(t)$ relate to $k$. So, it is reasonable to assume that, given a fixed pair of coupling parameters $(k,\tau)$, $(\eta(t),\dot{\eta}(t)) \in \Gamma_\eta \times \Gamma_\nu$, that is, the parameter vector $\eta(t)$ is constrained into a hypercube $\Gamma_\eta$, and the variation rate of $\eta(t)$ also belongs to a hypercube $\Gamma_\nu$. Hence, parameter-dependent conditions for all $(\eta(t),\dot{\eta}(t)) \in \Gamma_\eta \times \Gamma_\nu$, as we now provide, are sufficient for the stability of the partial synchronization manifold under consideration.

**Theorem 4.4.1.** *([84]) The LPV time-delay system (4.14) with bounded $(\eta(t),\dot{\eta}(t)) \in \Gamma_\eta \times \Gamma_\nu$ is asymptotically stable if there exist positive definite symmetric matrices $P(\eta), Q_j, Z_j \in \mathbb{R}^{(Nn-\kappa n)\times(Nn-\kappa n)}$, and matrices $G_j \in \mathbb{R}^{(4Nn-4\kappa n)\times(Nn-\kappa n)}$, $j = 1,2,3$, such that the following LMI holds for all possible $(\eta,\dot{\eta}) \in \Gamma_\eta \times \Gamma_\nu$,*

$$\begin{bmatrix} \Phi & \frac{\tau}{3}G_1 & \frac{\tau}{3}G_2 & \frac{\tau}{3}G_3 \\ * & -\frac{\tau}{3}Z_1 & \mathbf{0} & \mathbf{0} \\ * & * & -\frac{\tau}{3}Z_2 & \mathbf{0} \\ * & * & * & -\frac{\tau}{3}Z_3 \end{bmatrix} < 0, \tag{4.15}$$

*where (letting* $\mathrm{He}(A) := (A + A^\top)$*)*

$$\begin{cases} \Phi := \Phi_1 + \mathrm{He}\Big(\Lambda_0 P(\eta)\bar{\mathscr{A}} + \sum_{j=1}^{3} G_j \Lambda_j\Big) + \bar{\mathscr{A}}^\top \Big(\frac{\tau}{3}\sum_{j=1}^{3} Z_j\Big)\bar{\mathscr{A}}, \\ \Phi_1 := \mathrm{diag}\Big(Q_1 + \frac{\partial P(\eta)}{\partial \eta}\dot{\eta}, -Q_1 + Q_2, \\ \qquad\qquad -Q_2 + Q_3, -Q_3\Big), \\ \bar{\mathscr{A}} := \begin{bmatrix} \mathscr{A}(\eta) & \mathbf{0} & \mathbf{0} & \mathscr{A}_d \end{bmatrix}, \\ \Lambda_0 := \begin{bmatrix} I & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^\top, \\ \Lambda_1 := \begin{bmatrix} I & -I & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\ \Lambda_2 := \begin{bmatrix} \mathbf{0} & I & -I & \mathbf{0} \end{bmatrix}, \\ \Lambda_3 := \begin{bmatrix} \mathbf{0} & \mathbf{0} & I & -I \end{bmatrix}. \end{cases} \tag{4.16}$$

*The symbol* $*$ *denotes the symmetric terms in a symmetric matrix.*

The proof of Theorem 4.4.1 is given in Appendix C. In the proof, the following Lyapunov-Krasovskii functional is used:

$$V(t,\eta) = \mathscr{X}^\top(t)P(\eta)\mathscr{X}(t) + V_1(t) + V_2(t) + V_3(t), \tag{4.17}$$

where

$$V_j(t) := \int_{t-j\frac{\tau}{3}}^{t-(j-1)\frac{\tau}{3}} \mathscr{X}^\top(s)Q_j\mathscr{X}(s)\mathrm{d}s + \int_{-j\frac{\tau}{3}}^{-(j-1)\frac{\tau}{3}}\int_{t+\theta}^{t} \dot{\mathscr{X}}^\top(s)Z_j\dot{\mathscr{X}}(s)\mathrm{d}s\mathrm{d}\theta. \tag{4.18}$$

Here, $V_1 + V_2 + V_3$ is a "discretized" or "fragmented" version of the integral terms $\int_{t-\tau}^{t}\mathscr{X}^\top(s)Q\mathscr{X}(s)\mathrm{d}s + \int_{-\tau}^{0}\int_{t+\theta}^{t}\dot{\mathscr{X}}(s)Z\dot{\mathscr{X}}(s)\mathrm{d}s$ with a fragmentation order of 3. This discretization structure can reduce the conservatism of the results obtained by the Lyapunov-Krosovskii method [68]. However, it comes at a price of higher computational cost. We use a fragmentation order of 3 to balance between the conservatism and computational cost. Another technique, called the free-weighting matrix technique [90], is also used to reduce the conservatism. We refer to Appendix C for the details.

**Remark.** *The delay-dependent conditions in Theorem 4.4.1 are semi-infinite inequalities due to their parametric dependence. To cast the parameter-dependent conditions into a finite-dimensional optimization problem, the Lyapunov matrix* $P(\eta)$ *in* (4.17) *can be approximated by a finite set of basis functions [85]. The basis functions* $\{h_l(\eta)\}_{l=1}^{n_h}$ *for* $P(\eta)$ *should guarantee that*

$$P(\eta) = \sum_{l=1}^{n_h} h_l(\eta)P_l,\ P_l = P_l^\top. \tag{4.19}$$

*To further eliminate the dependence on the parameter vector* $\eta$*, a finite gridding* $\{\eta_r\}_{r=1}^{L}$ *of the parameter space can be introduced to generate finite-dimensional*

*convex optimization conditions [86]. We note that sufficient conditions in terms of finite LMIs can also be obtained using Pólya's relaxation or sum-of-squares techniques [91].*

**Remark.** *By omitting the double integral term associated with $Z_j$ in* (4.18)*, we can also derive delay-independent conditions for the stability of system* (4.14)*. This brings the following corollary.*

**Corollary 4.4.2.** *The LPV time-delay system (4.14) with bounded $(\eta(t), \dot{\eta}(t)) \in \Gamma_\eta \times \Gamma_v$ is asymptotically stable if there exist positive definite symmetric matrices $P(\eta)$, $Q_j \in \mathbb{R}^{(Nn-\kappa n)\times(Nn-\kappa n)}$, $j = 1,2,3$, such that the following LMI holds for all possible $(\eta, \dot{\eta}) \in \Gamma_\eta \times \Gamma_v$,*

$$\tilde{\Phi} := \Phi_1 + \mathrm{He}\big(\Lambda_0 P(\eta) \bar{\mathscr{A}}\big) < 0, \tag{4.20}$$

*where $\Phi_1$, $\Lambda_0$ and $\bar{\mathscr{A}}$ are defined in* (4.16)*.*

## 4.5 Numerical example

In this section, the results from Section 4.4 are applied to a network consisting of four Hindmarsh-Rose neurons interconnected via invasive coupling. It is explored under which conditions, in particular with which coupling strengths $k$ and time-delays $\tau$, partial synchronization can occur in such network.

Consider a network of four Hindmarsh-Rose neurons described by the following equations:

$$\begin{aligned}
\dot{x}_{i,1}(t) &= c - d x_{i,3}^2(t) - x_{i,1}(t), \\
\dot{x}_{i,2}(t) &= r_i(s(x_{i,3}(t) + v_0) - x_{i,2}(t)), \\
\dot{x}_{i,3}(t) &= -a x_{i,3}^3(t) + b x_{i,3}^2(t) + x_{i,1}(t) - x_{i,2}(t) + E_m + u(t), \\
y_i(t) &= x_{i,3}(t), \qquad\qquad i = 1,2,3,4,
\end{aligned} \tag{4.21}$$

where the parameters $a$, $b$, $c$, $d$, $r_i$, $s$, $v_0$, $E_m$ are constants, and $x_{i,1}(\cdot)$, $x_{i,2}(\cdot)$, $x_{i,3}(\cdot)$ and $u_i(\cdot)$ are the recovery variable, the adaptation variable, the membrane potential, and the external current of the $i$-th neuron [92]. The values of these parameters are $a = 1$, $b = 3$, $c = 1$, $d = 5$, $r_1 = r_3 = 0.004$, $r_2 = r_4 = 0.005$, $s = 4$, $v_0 = 1.618$ and $E_m = 3.25$. It has been proved in [93] that this Hindmarsh-Rose model is strictly semipassive. The structure of the network is described by the adjacency matrix

$$A = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{bmatrix}. \tag{4.22}$$

Here, we assume that the systems are coupled via (4.2). Using the tool developed in Chapter 3, only one viable partition $\{\{1,3\},\{2,4\}\}$ is identified. The full synchronization manifold does not exist, which indicates that full synchronization cannot be observed. The corresponding linearized synchronization error dynamics

$$\dot{\mathscr{X}}(t) = \mathscr{A}\mathscr{X}(t) + \mathscr{A}_d\mathscr{X}(t-\tau), \tag{4.23}$$

where

$$\begin{aligned}
&\mathscr{A} := \operatorname{diag}(A_1, A_2),\\
&A_1 := \begin{bmatrix} -1 & 0 & -2dx_{1,3}(t) \\ 0 & -r_1 & r_1 s \\ 1 & -1 & -3ax_{1,3}^2(t) + 2bx_{1,3}(t) - 3k \end{bmatrix},\\
&A_2 := \begin{bmatrix} -1 & 0 & -2dx_{2,3}(t) \\ 0 & -r_2 & r_2 s \\ 1 & -1 & -3ax_{2,3}^2(t) + 2bx_{2,3}(t) - 3k \end{bmatrix},\\
&\mathscr{A}_d := kA_{\text{red}} \otimes BC,\ B = C^\top = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top,\\
&A_{\text{red}} := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.
\end{aligned}$$

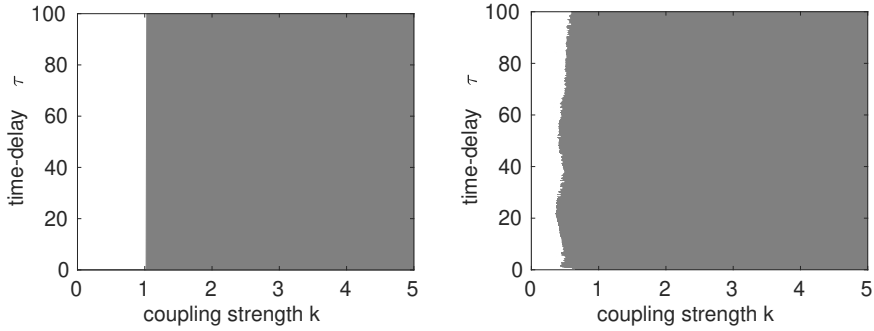
For this network of Hindmarsh-Rose neurons, given a pair of $(k,\tau)$, the stability analysis of the partial synchronization manifolds is performed in two steps: 1) estimate the bounds of $x_{1,3}(t)$, $x_{2,3}(t)$, $\dot{x}_{1,3}(t)$ and $\dot{x}_{2,3}(t)$ by numerical simulation of the network; 2) check the feasibility of the LMIs in Theorem 4.4.1. Specifically, to solve the stability problem, we pick five basis functions in expansion (4.19) as follows:

$$\begin{aligned}
&h_1(\eta) = 1,\ h_2(\eta) = x_{1,3}(t),\ h_3(\eta) = x_{1,3}^2(t),\\
&h_4(\eta) = x_{2,3}(t),\ h_5(\eta) = x_{2,3}^2(t).
\end{aligned} \tag{4.24}$$

To this end, by solving the condition (4.15), it has been checked that the Hindmarsh-Rose neurons with networks described by (4.22) are partially synchronous with $x_{1,3}(t) = x_{3,3}(t)$ and $x_{2,3}(t) = x_{4,3}(t)$ when $k = 1.05$ and $\tau = 10$. The simulation of the coupled systems is performed with the initial conditions $x_i(\theta) = [-3.6070\ 3.5837\ -0.7221]^\top, i = 1,2,3,4$ using the *dde23* solver from MATLAB, and the simulation runs over 500 time units. Based on the simulation, it is found that the states $x_{1,3}$ and $x_{2,3}$ are in the range of $[-2,2]$, and the state derivatives $\dot{x}_{1,3}$ and $\dot{x}_{2,3}$ are in the range of $[-6,6]$. With these bounds, the LMIs (4.15) is solved by using the MATLAB LMI toolbox [74].

Furthermore, by using Theorem 4.4.1 with different values of $k$ and $\tau$, we have calculated the synchronization region for $k \in [0,5]$ and $\tau \in [0,100]$, see the grey area in Figure 4.1(a). It is shown that when $k \geq 1.05$, the partial synchronization manifold is locally stable even for a quite large value of time-delay $\tau$. For comparison, the synchronization region detected by comparing the state trajectories of the same network is also depicted in Figure 4.1(b). The trajectories are calculated by simulating

the network in MATLAB using the same solver and initial conditions indicated in the previous paragraph. It can be seen from the two figures that the partial synchronization region obtained by Theorem 4.4.1 is smaller than the one obtained by simulation. This is not unexpected since the Lyapunov method generally comes with conservatism. Besides, the LPV time-delay system is an over-approximation of the synchronization error system. For example, the exogenous variables (parameters) $\eta$ in the LPV time-delay system (4.14) are actually state components of the coupled systems, and therefore these parameters are not independent from each other. It is possible that the exogenous variables $\eta$ and their derivatives $(\eta, \dot{\eta})$ cannot traverse every point in the space defined by $\Gamma_\eta \times \Gamma_\nu$.




(a) LMIs (b) Simulations

Figure 4.1: Partial synchronization regions (grey-colored area) in the $(k, \tau)$ parameter space computed via the LMIs in Theorem 4.4.1 (a) and via simulations (b)

**Remark.** *Using Theorem 4.4.1, it has been validated that when k is large enough ($k \geq 1.05$), the Hindmarsh-Rose neurons are partially synchronized ($x_{1,3}(t) = x_{3,3}(t)$ and $x_{2,3}(t) = x_{4,3}(t)$) within a large range of $\tau$. That is, the partial synchronization manifold tends to be locally stable independent of time-delay $\tau$. This can be explained via the concept of convergent dynamics ([63]). In particular, bearing in mind that the input matrix $B = [\ 0 \quad 0 \quad 1\ ]^\top$ and output matrix $C = B^\top$ in this example, it can be easily checked that $CB = 1 > 0$. Then, the system can be equivalently transformed into the following form:*

$$\begin{cases} \dot{\bar{\mathscr{X}}}(t) = g_1(\bar{\mathscr{X}}(t), \bar{y}(t)), \\ \dot{\bar{y}}(t) = g_2(\bar{y}(t), \bar{\mathscr{X}}(t)) - kb\bar{D}\bar{y}(t) + kb\bar{A}\bar{y}(t-\tau), \end{cases} \tag{4.25}$$

*where $\bar{A} := T_1^\top A T_1$, $\bar{D} := \mathrm{diag}(\tilde{\rho}_1, \tilde{\rho}_2)$, $b = CB = 1$, in addition, the $\bar{\mathscr{X}}(t)$ subsystem with $\bar{y}(t)$ as input is convergent. For this family of delay-coupled systems, it has been shown in [40] that the coupled networks can realize the delay-independent partial synchronization when k is larger than some positive number $k_0$. However, it is difficult*

*to compute the exact value of this threshold $k_0$ by the synchronization conditions in [40]. Resorting to Theorem 4.4.1 in this chapter, we can estimate the value of $k_0$.*

**Remark.** *There is another method to estimate the bounds of $(\eta, \dot{\eta})$, which does not rely on simulations. As mentioned at the beginning of this chapter, the solutions of interconnected semipassive systems coupled via invasive coupling are ultimately bounded. It has been proved that Hindmarsh-Rose neurons are semipassive [93]. Therefore, the coupled Hindmarsh-Rose neurons in this example have ultimately bounded solutions. In fact, an invariant set can be found for these coupled neurons, see Appendix D for more details. It is also shown in Appendix D that the solutions of each coupled neuron can stay inside a box $\mathbb{B} = [-73.0705, 1] \times [-3.6994, 3.8489] \times [-8.3256, 21.8676] \in \mathbb{R}^3$ once they enter this box. This box is also an invariant set for an uncoupled neuron. However, the solutions of coupled neurons are confined to a smaller space than the box, see Figure 4.2. It shows the invariant box and the trajectories of coupled neurons with $k = 1.05$ and $\tau = 10$ that are initialized at four different vertices of the invariant box. The trajectories of the neurons converge to a much smaller space after the transient period. Therefore, the bounds of $(\eta, \dot{\eta})$ inferred from this invariant box are much larger than the actual bounds, which leads to a quite conservative result when using the LMI conditions to identify the synchronization regions. For example, for all solutions inside this box, the states $x_{1,3}$ and $x_{2,3}$ are in the range of $[-3.6994, 3.8489]$, and the state derivatives $\dot{x}_{1,3}$ and $\dot{x}_{2,3}$ are in the range of $[-129.8799, 123.8744]$. Based on the bounds of these intervals, the LMIs are only feasible for $\tau = 10$ when $k \geq 9$.*

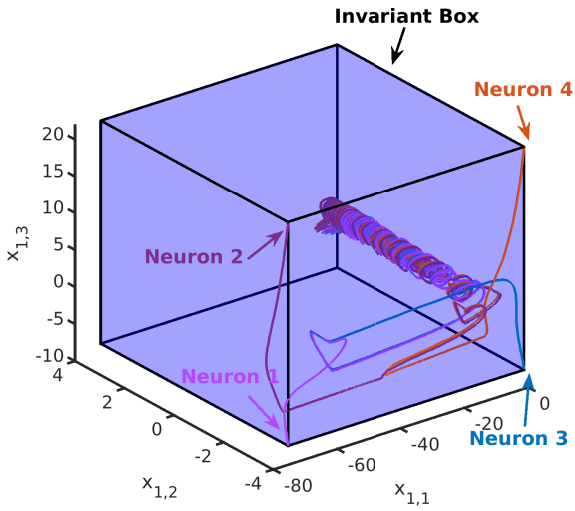



Figure 4.2: The invariant box and the trajectories of coupled neurons

## 4.6 Conclusions

In this chapter, the local stability of (exact) partial synchronization manifolds is addressed by utilizing the Lyapunov method. First, the synchronization error dynamics are isolated from the network dynamics, whose stability determines the stability of the partial synchronization manifold under consideration. Second, it is shown that the linearized synchronization error dynamics can be (over)approximated by an LPV time-delay system. By choosing a parameter- and delay-dependent Lyapunov-Krasovskii functional, the stability conditions are formulated in terms of LMIs, which can be solved efficiently. Finally, a numerical example is presented, where this method is applied to a network consisting of four Hindmarsh-Rose neurons.

# Chapter 5

# Practical partial synchronization in networks of delay-coupled systems

**Abstract:** In this chapter, sufficient conditions for practical partial synchronization in networks of delay-coupled systems are presented. Due to perturbations or uncertainties in the network or system dynamics, practical partial synchronization may be observed, which refers to the situation where the states of the systems within each cluster converge to each other up to some bound, and this bound tends to zero if (the size of) the perturbations tend to zero. In this chapter, the case where perturbations exist in the interactions of systems is considered. To derive sufficient conditions for practical partial synchronization, first, the synchronization error dynamics are separated from the network dynamics, and we interpret them in terms of a nonautonomous system of delay differential equations with a bounded additive perturbation. Second, by assessing the practical stability of this error system, conditions for practical partial synchronization are derived and formulated in terms of linear matrix inequalities. In addition, an explicit relation between the size of perturbation and the bound of the synchronization error is provided. The results presented in this chapter are published in [94].

## 5.1 Introduction

As mentioned in the previous chapters, the existence and (asymptotic) stability of partial synchronization manifolds are required to observe partial synchronization in

a network of coupled systems. Such manifolds are actually hyperplanes in the state space of the networked systems. When such a manifold exists, the solutions of the systems that start on the corresponding hyperplane will remain on it. For solutions that are initialized outside that hyperplane, the stability of the partial synchronization manifold will ensure the solutions to be attracted to this hyperplane. See Figure 5.1 for an illustration of these concepts using a network of three coupled systems as an example. The existence of partial synchronization manifold $\mathscr{M} = \{(x_1, x_2, x_3) | x_1 = x_2\}$ guarantees that the trajectory A stays on the manifold (middle plane, in red) once it reaches the manifold. The stability of the partial synchronization manifold guarantees that trajectory A is attracted to the manifold.

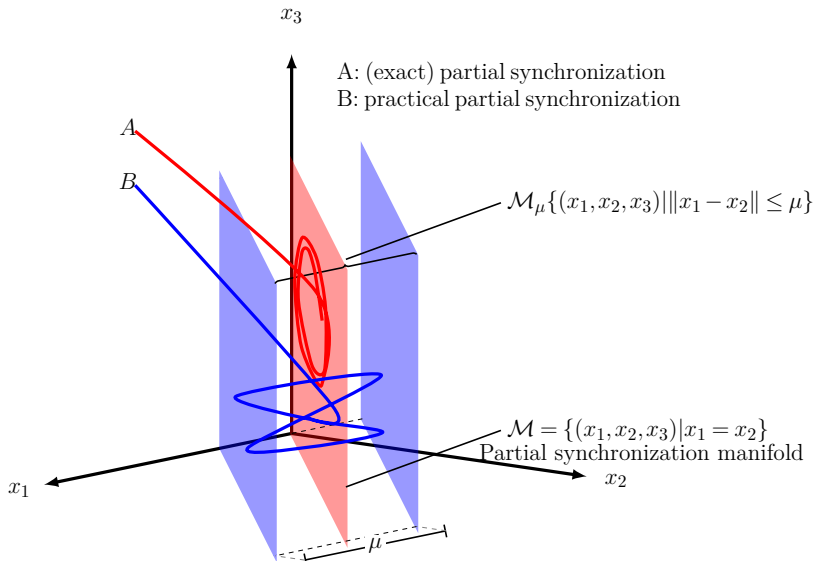



Figure 5.1: Partial synchronization and practical partial synchronization in a network of three coupled first-order systems with systems 1 and 2 being synchronized.

The necessary and sufficient conditions for the existence of partial synchronization manifolds have been presented in Chapter 3. However, the existence conditions are fragile, in the sense that they may be violated by infinitesimal perturbations on coupling weights and delays, as well as perturbations to the dynamical systems at the nodes that create more heterogeneity. As a result, synchronization in the sense that the states of the synchronized systems converge to each other cannot be achieved. Instead, an approximate form of synchronization may be observed, in the sense that the states (or outputs) of the synchronized systems are not completely identical, but remain close to each other within some bound that depends on the size of the perturbation. Inspired by the notion of practical stability, this robust version of synchronization is referred

to as *practical synchronization*. Figure 5.1 also illustrates an example of such a form of synchronization. As the figure shows, the solution may be unable to stay on the manifold because of perturbations. In such a case, the trajectory B is unable to stay on the middle plane (in red), but it remains inside the space between the two outer planes (in blue). This space is represented by the set $\mathscr{M}_\mu = \{(x_1, x_2, x_3) | \|x_1 - x_2\| \leq \mu\}$. The example in this figure is a finite-dimensional case, however, the idea holds true for the infinite-dimensional case as well. The actual reason inducing practical synchronization is that the linear subspace characterized by the zero synchronization errors $E \equiv 0$ is no longer (positively) invariant in the system of synchronization errors due to the perturbations. In other words, the synchronization error system has no zero equilibrium in this case. Without the zero equilibrium, an intuitive expectation is that the errors will stay close to zero if the perturbation is small under suitable conditions. However, this is not always the case, for example, when the system is in the vicinity of a bifurcation point. Hence, it is important to develop methods to guarantee practical partial synchronization. In addition, due to network imperfections, modeling error, *etc.*, perturbations are often present in the network of coupled systems, thus making it important to study this type of synchronization.

Several studies have been performed to investigate practical synchronization in delay-free coupled systems, see [44, 3, 43, 42]. In [44] and its supplementary material [3], theoretical results on practical synchronization of delay-free, diffusively coupled non-identical systems with application to Hindmarsh-Rose neurons are presented. It is shown that when the coupling strength is sufficiently large, the systems achieve practical synchronization by using the property of semi-passivity and convergent system theory. In [43], some preliminary results on practical synchronization of delay-free interconnected heterogeneous oscillators are presented, where the product between the input matrix and output matrix is assumed to be positive definite and the zero dynamics subsystem is convergent. In [42], by assuming the QUAD property (see [95]) for each system, conditions on the interconnection structure of networks of perturbed nonlinear systems with non-delayed diffusive couplings are derived to achieve practical synchronization. The benefit of the results in [43], [96], [44] and [42] is that they provide conceptual insights into the collective behavior of network-interconnected systems with model heterogeneity. However, no work has been done on the quantitative analysis of practical partial synchronization of interconnected systems, in particular in the presence of delayed couplings.

In this chapter, the practical partial synchronization in networks of delay-coupled systems is studied. We start with partial synchronization investigated in Chapters 3-4 as a nominal case. Subsequently, we focus on the case where there are perturbations on the coupling. It is shown that when the perturbations are present, exact partial synchronization becomes in general impossible. More precisely, the synchronization error dynamics have no zero equilibrium due to perturbations. Without the zero equilibrium, the methods used in most works on synchronization (including the one

presented in Chapter 4) which require a linearization around the equilibrium are not applicable. Here, with (exact) partial synchronization as a nominal case, we study its persistence under perturbations. In particular, by connecting the practical partial synchronization with the practical stability of the synchronization error dynamics, we derive conditions for practical partial synchronization. In this practical stability analysis, an equilibrium of the synchronization error dynamics is not required. To this end, firstly, the dynamics of synchronization errors are isolated from the network dynamics with the inclusion of the perturbation; secondly, by viewing the synchronization error dynamics as a time-delay system affected by additive perturbations, conditions for the practical stability of such an error system, corresponding to the conditions for practical partial synchronization of the network are derived. The conditions are formulated in the form of LMIs, which can be efficiently solved by several existing numerical tools.

The remainder of this chapter is organized as follows. In Section 5.2, by reusing the setting of Chapter 3, the networks of delay-coupled systems under consideration are introduced, including the systems and couplings. In addition, partial synchronization is re-introduced as a nominal case for practical partial synchronization. It is shown that small perturbations on the coupling can render exact synchronization impossible and lead to practical partial synchronization. In Section 5.3, the main problem addressed in this chapter is formulated by formally defining practical partial synchronization. In Section 5.4, the main results of this chapter, *i.e.*, the sufficient conditions for practical partial synchronization, are presented. In Section 5.5, a numerical example is presented. Finally, the conclusions are given in Section 5.6.

## 5.2 Networks

This section introduces the setting of the networks, including the systems in the networks and the coupling between the systems. In addition, the (exact) partial synchronization manifold is re-introduced and one example is used to show that small perturbations can make exact synchronization impossible and lead to practical partial synchronization.

In this chapter, we consider the same networks of systems interacting via linear diffusive time-delay couplings as in the previous chapters, therefore the same setting is adopted. These networks are represented by simple, strongly connected directed graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where

- $\mathcal{V} = \{1, 2, \ldots, N\}$ is a finite set of nodes with cardinality $|\mathcal{V}| = N$;
- $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the *ordered* set of edges, where the edge $(i, j)$ points from node $i$ to node $j$;

- $A = \left(a_{ij}\right) \in \mathbb{R}^{N\times N}$ is the weighted adjacency matrix, where $a_{ij} > 0$ represents the weight of edge $(j,i)$ when $(j,i) \in \mathscr{E}$, and $a_{ij} = 0$ when $(j,i) \notin \mathscr{E}$.

Every node in the networks hosts a dynamical system of the form

$$\begin{cases} \dot{x}_i(t) = f_i(x_i(t)) + B_i u_i(t) \\ y_i(t) = C_i x_i(t), \end{cases} \tag{5.1}$$

where $i \in \mathscr{V}$, states $x_i(t) \in \mathbb{R}^n$, sufficiently smooth functions $f_i : \mathbb{R}^n \to \mathbb{R}^n$, inputs $u_i(t) \in \mathbb{R}^m$, outputs $y_i(t) \in \mathbb{R}^m$, input matrices $B_i \in \mathbb{R}^{n\times m}$ and output matrices $C_i \in \mathbb{R}^{m\times n}$, $i = 1,\ldots,N$. Here, we assume all $C_iB_i$ are similar to positive definite matrices, which guarantees that all the systems are left-invertible (the system input-out maps are injective).

The systems (5.1) interact via invasive coupling

$$u_i(t) = k \sum_{j\in\mathscr{N}_i} a_{ij}[y_j(t-\tau) - y_i(t)], \tag{5.2}$$

where, $\mathscr{N}_i$ is the neighboring set of node $i$, *i.e.*, $\mathscr{N}_i := \{j \in \mathscr{V} \mid (i,j) \in \mathscr{E}\}$, and $\tau$, $k$ are, respectively, the time-delay and coupling strength.

As in Chapter 4, we have the following assumption on systems (5.1).

**Assumption 5.2.1.** *Systems* (5.1) *are strictly semipassive with a radially unbounded storage function* $V$.

This assumption on semipassivity is to be used to guarantee the boundedness of solutions of the networked systems. The boundedness is later on used in the derivation of conditions for practical partial synchronization. It has been shown in [63] that Assumption 5.2.1 guarantees all solutions of the systems (5.1) coupled via (5.2) to be ultimately bounded independently of $k$ and $\tau$. More precisely, it has been pointed out in [63] that under this assumption, there exists a bounded set $\Omega$, positively invariant under the dynamics (5.1), (5.2), which attracts all solutions in finite time.

Networks of systems interconnected via non-invasive coupling are not considered in this chapter because for such networks, Assumption 5.2.1 does not guarantee the solutions of the systems to be ultimately bounded. An additional restriction on the coupling strength is required (an upper bound should be imposed on the coupling strength), see [1] for more details. If the solutions of the systems in networks with non-invasive coupling are ultimately bounded, the results presented in this chapter can be extended to such networks.

The state space of the coupled systems (5.1) and (5.2) is denoted by $\mathscr{C}([-\tau,0],\mathbb{R}^{Nn})$. The network state is defined as below

$$x_t \in \mathscr{C}([-\tau,0],\mathbb{R}^{Nn}), x_t = x(t+\theta), \theta \in [-\tau,0]. \tag{5.3}$$

In what follows we review the results on exact practical partial synchronization from the previous chapters, which serve as a nominal case in the subsequent sections. In addition, an example is presented which shows that these conditions can easily become invalid due to a small perturbation. In such a situation, exact synchronization is no longer possible, and only practical synchronization may occur.

As in Chapter 3, the set of partially synchronous states induced by the permutation matrix $\Pi$ (representing a partition $\mathscr{P}$) is given by

$$\mathscr{M}(\Pi) := \{\phi \in \mathscr{C}([-\tau,0],\mathbb{R}^{Nn}) \,|\, \phi(\theta) = \mathrm{col}(\phi_1(\theta),\dots,\phi_N(\theta)),$$
$$\phi_i(\theta) \in \mathbb{R}^n,\ i=1,\dots,N,\ \phi(\theta) = (\Pi \otimes I_n)\phi(\theta),\ \forall \theta \in [-\tau,0]\}.$$

Then, the set $\mathscr{M}$ is an (exact) partial synchronization manifold when it is positively invariant under the dynamics (5.1), (5.2), see Definition 3.2.2.

A partition $\mathscr{P}$ associated with $\Pi$ is called *viable* if the set $\mathscr{M}(\Pi)$ is a partial synchronization manifold. The viability of a partition can be verified using the row-sum conditions and conditions (3.9) in Theorem 3.2.3. However, these conditions are very fragile in the sense that they may be violated by infinitesimal perturbations on coupling weights and delays, as well as perturbations to the dynamical systems at the nodes which create more heterogeneity. In the example below, we demonstrate that small perturbations on the coupling cause such conditions to be violated, and as a result, exact partial synchronization can not be observed.

**Example 5.2.2.** *Consider a network of four nodes shown in Figure 5.2. The adjacency matrix of this network is*

$$A_n = \begin{bmatrix} 0 & 2 & 0.1 & 1 \\ 4 & 0 & 2 & 0.3 \\ 0.1 & 1 & 0 & 2 \\ 3 & 0.3 & 3 & 0 \end{bmatrix}. \tag{5.4}$$

*We assume that every node hosts a nonlinear system of the form*

$$\begin{cases} \dot{x}_{i,1}(t) = 0.2x_{i,1}(t) + x_{i,2}(t) - 10x_{i,1}(t)(x_{i,1}^2(t) + x_{i,2}^2(t)) + u_{i,2}(t) \\ \dot{x}_{i,2}(t) = -x_{i,1}(t) + 0.2x_{i,2}(t) - 10x_{i,2}(t)(x_{i,1}^2(t) + x_{i,2}^2(t)) + u_{i,1}(t) \\ y_{i,1}(t) = x_{i,2}(t) \\ y_{i,2}(t) = x_{i,1}(t), \quad i = 1,2,3,4, \end{cases} \tag{5.5}$$

*and that the systems are coupled via* (5.2)*. Note that the system* (5.5) *is semipassive, see Appendix G.*

*By applying Theorem 3.2.3 to all possible partitions of this network, we find the only viable partition $\mathscr{P} = \{\{1,3\},\{2,4\}\}$. This can be done automatically by using the*

*software program presented in Chapter 3. As the network here is relatively simple, the viability of this partition can be easily done by hand. First, since all the systems are identical, conditions* (3.9) *are satisfied. Second, the constant row sum requirement is also met. The corresponding permutation matrix* $\Pi$ *and the reordering matrix R are given by*

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

*The reordered adjacency matrix is*

$$R^\top A_n R = \left[\begin{array}{cc|cc} 0 & 0.1 & 2 & 1 \\ 0.1 & 0 & 1 & 2 \\ \hline 4 & 2 & 0 & 0.3 \\ 3 & 3 & 0.3 & 0 \end{array}\right].$$

*Each block in this matrix has a constant row sum.*

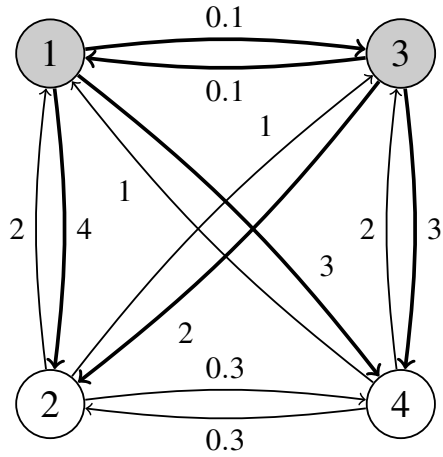



Figure 5.2: The network of Example 5.2.2 with adjacency matrix $A_n$

*To validate the result, a simulation of this network with* $k = 0.3$ *and* $\tau = 2$ *has been performed. The results are plotted in Figure 5.3. The network is initialized on the partial synchronization manifold, i.e.,* $x_1(\theta) = x_3(\theta), x_2(\theta) = x_4(\theta), -\tau \le \theta \le 0$*. From Figure 5.3, we can see that systems 1 and 3, 2 and 4 remain perfectly synchronized. The states remain on the partial synchronization manifold since they start on the manifold. However, as mentioned before, the existence conditions of such manifold are fragile. For instance, a small perturbation on the coupling can cause a violation of these*

*conditions. Assume there exists a perturbation $\Delta A$ on the adjacency matrix:*

$$A = A_n + \Delta A = \begin{bmatrix} 0 & 2 & 0.1 & 1 \\ 4 & 0 & 2 & 0.3 \\ 0.1 & 1 & 0 & 2 \\ 3 & 0.3 & 3 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0.2 & 0 & 0 \\ 0.2 & 0 & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0.2 & 0 \end{bmatrix}. \tag{5.6}$$

*With this adjacency matrix, the row-sum conditions in Theorem 3.2.3 are not satisfied. The simulation of this perturbed network is shown in Figure 5.4. As can be seen from Figure 5.4, the states of systems 1 and 3, 2 and 4 are not identical, but remain close to each other, that is, an approximate form of synchronization occurs. This is also confirmed in Figure 5.5 which shows the synchronization errors defined as $e_1 = x_{3,1} - x_{1,1}$, $e_2 = x_{3,2} - x_{1,2}$, $e_3 = x_{4,1} - x_{2,1}$ and $e_4 = x_{4,2} - x_{2,2}$. In this figure, to better indicate the sizes of these errors, they are expressed as the percentages of maximum magnitudes of the relevant state variables ($\frac{e_1}{\max\limits_t |x_{1,1}(t)|} \times 100\%$, $\frac{e_2}{\max\limits_t |x_{1,2}(t)|} \times 100\%$, $\frac{e_3}{\max\limits_t |x_{2,1}(t)|} \times 100\%$, $\frac{e_4}{\max\limits_t |x_{2,2}(t)|} \times 100\%$). Besides, we can observe that $e_i = 0$ is not an equilibrium of the synchronization error dynamics when the perturbation is present. The results above may not be very surprising. For small perturbations to the nominal adjacency matrix corresponding to the coupling, one would intuitively expect the solutions to remain almost synchronized within the clusters. However, this intuition may also fail, for instance, if the system is close to a bifurcation point. Therefore, it is important to develop tools that guarantee the required robustness of the synchronization with respect to the perturbation.*

*In the next section, this situation with perturbation on coupling impacting the partial synchronization of delay-coupled systems is generalized and forms the main problem addressed in this chapter.*

**Remark.** *First, we would like to emphasize that in this example with* (5.4)*, no full synchronization manifold exists. By definition, a full synchronization manifold is always a subset of any partial synchronization manifold. Therefore, to have partial synchronization, the stability conditions of the partial synchronization manifold should not coincide with the stability conditions of the full synchronization manifold. If no full synchronization manifold exists, this requirement is readily met. Second, from the reordered adjacency matrix, we can see that the coupling of the nodes within clusters is weak, and that the coupling of the nodes between clusters is strong (see the ordered adjacency matrix). In fact, this property favors the partial synchronization manifold to be stable in the network with $A_n$ for some $(k, \tau)$. It might sound counter-intuitive but can be explained by the presence of the delay and the sensitivity of high gain feedback with respect to it. Thus, while in the delay-free case high gain coupling favors synchronization, the situation is different in the presence of time-delay, see [78]. More precisely, an upper bound exists for the product of the coupling gain and time-delay such that the delay-coupled network can be synchronized [1]. In other words, for a*

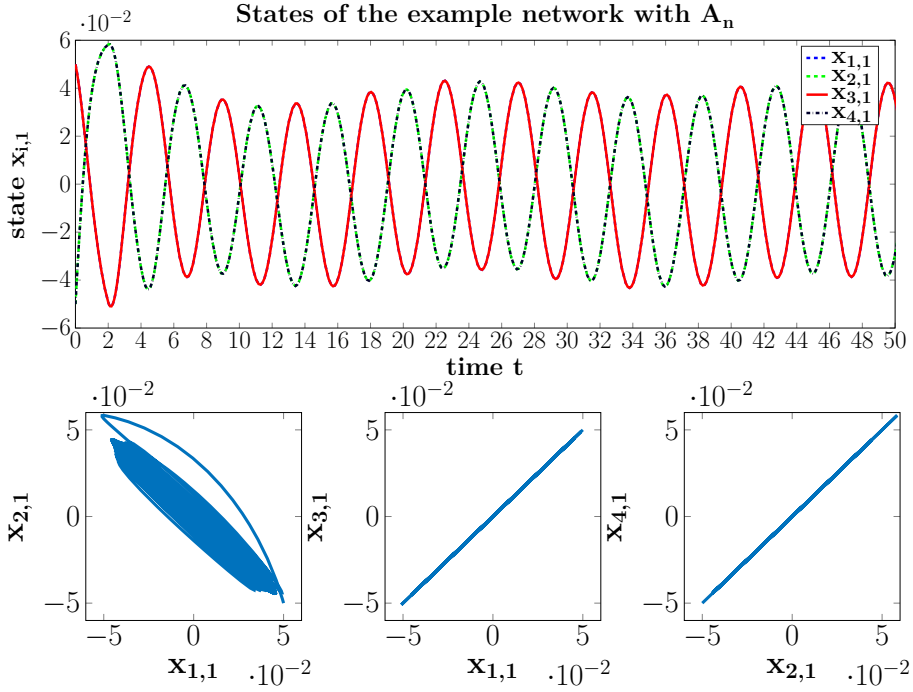



Figure 5.3: Simulation results of the network corresponding to Example 5.2.2, with adjacency matrix $A_n$

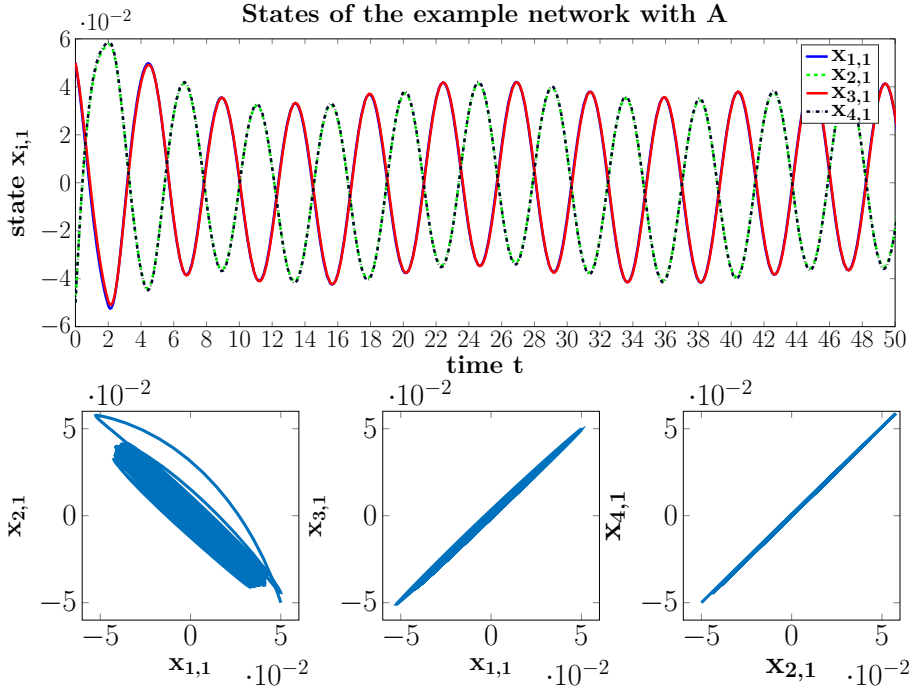



Figure 5.4: Simulation results of the network corresponding to Example 5.2.2, with adjacency matrix $A$

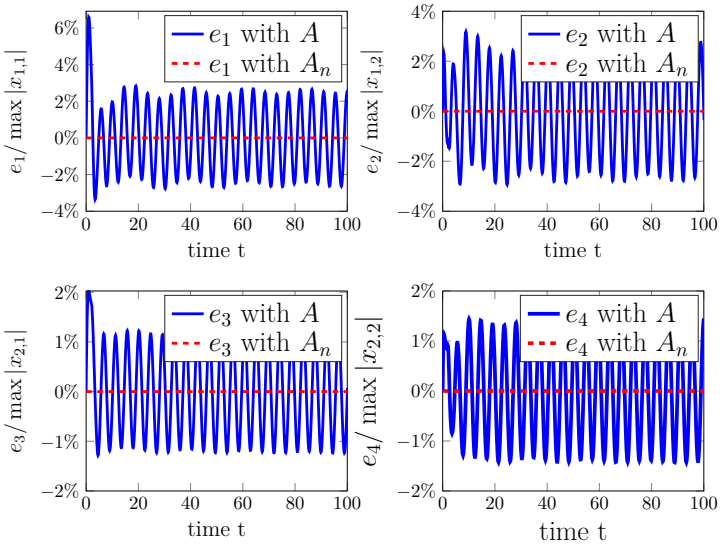



Figure 5.5: Synchronization errors of the network corresponding to Example 5.2.2, with $A$ and $A_n$

*given time-delay, the coupling gain should be bounded such that the delay-coupled network can be synchronized.*

## 5.3 Practical partial synchronization

This section introduces the concept of practical partial synchronization, addressing the case where there are perturbations on the coupling. In addition, the synchronization error dynamics are also derived, which lays the basis for the stability analysis in the next section.

### 5.3.1 Practical partial synchronization definition

Suppose that there exists a viable partition $\mathscr{P}$ for the network of systems represented by the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, A_n)$ and assume that the adjacency matrix of the network is perturbed to $A = A_n + \Delta A$, with a perturbation bound

$$\|\Delta A\|_\infty \leq \delta. \tag{5.7}$$

Note that the perturbed adjacency matrix $A$ should preserve the basic properties of the adjacency matrix: non-negative entries and zero diagonal entries. That is, $\Delta A$ is constrained such that $a_{ij}^{\Delta} = 0$ if $i = j$ and $a_{ij}^{\Delta} \geq -a_{ij}^{n}$ if $i \neq j$ with $a_{ij}^{\Delta}$ being the entries of $\Delta A$ and $a_{ij}^{n}$ the entries of $A_n$.

In this case, the row-sum conditions in Theorem 3.2.3 may be violated even for arbitrarily small perturbations, and exact partial synchronization may become impossible. However, using the following relaxation of the set $\mathscr{M}(\Pi)$ for $\mu \geq 0$

$$\mathscr{M}_{\mu}(\Pi) := \{\phi \in \mathscr{C}([-\tau,0],\mathbb{R}^{Nn}) \,|\, \phi(\theta) = \mathrm{col}(\phi_1(\theta),\ldots,\phi_N(\theta)),$$
$$\phi_i(\theta) \in \mathbb{R}^n, i = 1,\ldots,N,\ \|\phi(\theta) - (\Pi \otimes I_n)\phi(\theta)\| \leq \mu, \forall \theta \in [-\tau,0]\},$$

we can analyze an approximate form of partial synchronization, formally defined as follows.

**Definition 5.3.1.** *For a network of systems* (5.1)*,* (5.2) *represented by the graph* $\mathscr{G} = (\mathscr{V},\mathscr{E},A_n)$*, given a partition* $\mathscr{P}$*, permutation matrix* $\Pi$ *and perturbation bound* $\delta$ *in (5.7), the network is practically partially synchronized with respect to the clustering represented by* $\mathscr{P}$ *if there exist a class* $\mathscr{K}$ *function* $\beta(\delta)$ *and a function* $T = T(\phi,\delta)$ *such that whenever the adjacency matrix is perturbed to* $A_n + \Delta A$ *with* $||\Delta A||_{\infty} \leq \delta$ *the solutions satisfy*

$$x_t(\phi) \in \mathscr{M}_{\mu}(\Pi), \quad \forall \phi \in \mathscr{C}([-\tau,0],\mathbb{R}^{Nn}),\ \forall t > T,$$

*where* $\mu = \beta(\delta)$.

Note that since $\beta \in \mathscr{K}$, $\mathscr{M}_{\mu=0}(\Pi) = \mathscr{M}(\Pi)$, hence, practical partial synchronization implies partial synchronization of the nominal network.

### 5.3.2 Perturbed synchronization error dynamics

Here, the perturbed synchronization error dynamics (the synchronization error dynamics of the perturbed networks) are derived. First, the synchronization error dynamics of the nominal networks are presented by recalling the results of Chapter 3. Then, by adding the perturbations to the error system, the synchronization error dynamics of the perturbed networks are derived. By comparing the synchronization error dynamics of these two cases, the impacts of the perturbations are shown, which explains why exact partial synchronization becomes impossible in the presence of perturbations. In addition, the perturbed synchronization error dynamics will be used to derive the conditions for practical partial synchronization in the next section.

**Nominal network** We start with the nominal network using the procedure in Chapter 3. Suppose that there exists a viable partition $\mathscr{P}$ for the networked systems with the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, A_n)$. Without loss of generality, we assume that the systems have been pre-ordered into clusters according to the partition $\mathscr{P}$ associated with $\Pi$ as follows

$$\begin{array}{ll} x_{1,1},\ x_{1,2},\ldots,x_{1,\kappa_1} & \text{cluster } 1,\\ x_{2,1},\ x_{2,2},\ldots,x_{2,\kappa_2} & \text{cluster } 2,\\ \vdots & \vdots\\ x_{\kappa,1},\ x_{\kappa,2},\ldots,x_{\kappa,\kappa_\kappa} & \text{cluster } \kappa. \end{array}$$

We adopt the same notations as in Section 3.4. The synchronization error dynamics of the nominal network are then given by the time-delay system (3.20). To rewrite it in a more compact form, we define $\mathscr{X}(t), \mathscr{X}_r(t), A_0, A_1, F(\mathscr{X}(t), \mathscr{X}_r(t))$ as below.

$$\mathscr{X}(t) = \begin{bmatrix} E_1(t)\\ \vdots\\ E_\kappa(t) \end{bmatrix} \in \mathbb{R}^{Nn-\kappa n}, \quad \mathscr{X}_r(t) = \begin{bmatrix} \mathbf{1}_{\kappa_1-1}\otimes x_{1,1}(t)\\ \vdots\\ \mathbf{1}_{\kappa_\kappa-1}\otimes x_{\kappa,1}(t) \end{bmatrix} \in \mathbb{R}^{Nn-\kappa n},$$

$$A_0 = \begin{bmatrix} I_{\kappa_1-1}\otimes(\rho_1\tilde{B}_1\tilde{C}_1) & & 0\\ & \ddots & \\ 0 & & I_{\kappa_\kappa-1}\otimes(\rho_\kappa\tilde{B}_\kappa\tilde{C}_\kappa) \end{bmatrix},$$

$$A_1 = \tilde{\mathscr{B}}(A_{\text{red}}\otimes I_m)\tilde{\mathscr{C}},$$

$$F(\mathscr{X}(t), \mathscr{X}_r(t)) = \begin{bmatrix} \tilde{f}_1(x_{1,1}(t)+e_{1,2}(t)) - \tilde{f}_1(x_{1,1}(t))\\ \vdots\\ \tilde{f}_1(x_{1,1}(t)+e_{1,\kappa_1}(t)) - \tilde{f}_1(x_{1,1}(t))\\ \vdots\\ \tilde{f}_\kappa(x_{\kappa,1}(t)+e_{\kappa,2}(t)) - \tilde{f}_\kappa(x_{\kappa,1}(t))\\ \vdots\\ \tilde{f}_\kappa(x_{\kappa,1}(t)+e_{\kappa,\kappa_\kappa}(t)) - \tilde{f}_\kappa(x_{\kappa,1}(t)) \end{bmatrix}. \tag{5.8}$$

See Section 3.4 for the definitions of $E_i, \rho_i, \tilde{B}_i, \tilde{C}_i, \tilde{f}_i, i = 1,\ldots,\kappa$ and $A_{\text{red}}$. The vectors $\mathscr{X}$ and $\mathscr{X}_r$ are the collection of the synchronization errors and states of the reference systems, respectively. Roughly speaking, $F$ is related to the differences of the nonlinear terms $f_i$ between the systems in each cluster, $A_0$ is related to the dynamics of the reference systems, and $A_1$ is related to the network structure. Note that $F(\mathbf{0}, \mathscr{X}_r(t)) \equiv \mathbf{0}$.

In this way, the error dynamics of the nominal network can be rewritten as

$$\dot{\mathscr{X}}(t) = F(\mathscr{X}(t), \mathscr{X}_r(t)) - kA_0\mathscr{X}(t) + kA_1\mathscr{X}(t-\tau). \tag{5.9}$$

As a consequence of the viability of the partition, $\mathscr{X}(t) = 0$ is an equilibrium point of the synchronization error dynamics (5.9). More specifically, because of the row sum conditions and condition (3.9) in Theorem 3.2.3, the other two terms containing $\mathscr{X}_r(t)$ and $\mathscr{X}_r(t-\tau)$ respectively, are canceled during the derivation of (5.9). These two terms are not zero when $\mathscr{X}(t) = 0$. The stability of partial synchronization manifolds can be inferred from the stability of this equilibrium, which is necessary for networks to exhibit exact partial synchronization in practice.

**Perturbed network** We now look into the effect of the perturbation in (5.7) on the synchronization error dynamics (5.9). In the presence of the perturbation, the row sums of the blocks in the reordered adjacency matrix (3.8) are not anymore constant. We keep associating $x_{1,1}, \ldots, x_{\kappa,1}$ with the role of reference systems. Hence, we continue to denote $\rho_{ij}$ as the row sum of the first row of $ij^{\text{th}}$ block of the adjacency matrix for $i, j \in \{1, \ldots, \kappa\}$, while the sums of the other rows become $\rho_{ij} + \Delta_{ij,l}$ with $l = 2, \ldots, \kappa_i$. That is, for the block $A_{ij}$ in (3.8), its row sums are denoted by

$$\begin{bmatrix} \rho_{ij} \\ \rho_{ij} + \Delta_{ij,2} \\ \vdots \\ \rho_{ij} + \Delta_{ij,\kappa_i} \end{bmatrix}, \tag{5.10}$$

where, by the assumption that $\|\Delta A\|_\infty \leq \delta$, $|\Delta_{ij,l}| \leq \delta$ for $i, j = 1, \cdots, \kappa, l = 1, \cdots, \kappa_i$. The collection of $\Delta_{ij,l}$ is denoted by $\Delta \in \mathbb{R}^{(N-\kappa)\times\kappa}$ as below

$$\Delta = \begin{bmatrix} \Delta_{11,2} & \Delta_{12,2} & \cdots & \Delta_{1\kappa,2} \\ \vdots & \vdots & \vdots & \vdots \\ \Delta_{11,\kappa_1} & \Delta_{12,\kappa_1} & \cdots & \Delta_{1\kappa,\kappa_1} \\ \vdots & \ddots & \ddots & \vdots \\ \Delta_{\kappa 1,2} & \Delta_{\kappa 2,2} & \cdots & \Delta_{\kappa\kappa,2} \\ \vdots & \vdots & \vdots & \vdots \\ \Delta_{\kappa 1,\kappa_1} & \Delta_{\kappa 2,\kappa_1} & \cdots & \Delta_{\kappa\kappa,\kappa_1} \end{bmatrix}. \tag{5.11}$$

Then the synchronization error dynamics can be expressed as

$$\dot{\mathscr{X}}(t) = F(\mathscr{X}(t), \mathscr{X}_r(t)) - kA_0\mathscr{X}(t) + kA_1\mathscr{X}(t-\tau) + \Xi(\mathscr{X}_r(t), \mathscr{X}_r(t-\tau), \Delta), \tag{5.12}$$

where the additional term compared to (5.9) is given by

$$\Xi(\mathscr{X}_r(t), \mathscr{X}_r(t-\tau), \Delta) = A_2(\Delta)\mathscr{X}_r(t) + A_3(\Delta)\mathscr{X}_r(t-\tau), \tag{5.13}$$

satisfying the property that

$$\Xi(\mathscr{X}_r(t), \mathscr{X}_r(t-\tau), \mathbf{0}) = \mathbf{0}. \tag{5.14}$$

The matrices $A_2(\Delta)$ and $A_3(\Delta)$ are defined as follows:

$$A_2(\Delta) = k \begin{bmatrix} \operatorname{diag}\left(\sum_{l=1}^{\kappa} \Delta_{1l,2}, \ldots, \sum_{l=1}^{\kappa} \Delta_{1l,\kappa_1}\right) \otimes \tilde{B}_1 \tilde{C}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \operatorname{diag}\left(\sum_{l=1}^{\kappa} \Delta_{1l,2}, \ldots, \sum_{l=1}^{\kappa} \Delta_{1l,\kappa_\kappa}\right) \otimes \tilde{B}_\kappa \tilde{C}_\kappa \end{bmatrix}, \tag{5.15}$$

$$A_3(\Delta) = k\tilde{\mathscr{B}}(A_\Delta \otimes I_m)\tilde{\mathscr{C}}, \tag{5.16}$$

where

$$A_\Delta = \begin{bmatrix} \operatorname{diag}(\Delta_{11,2}, \ldots, \Delta_{11,\kappa_1}) & \cdots & \operatorname{diag}(\Delta_{1\kappa,2}, \ldots, \Delta_{1\kappa,\kappa_1}) \\ \vdots & \ddots & \vdots \\ \operatorname{diag}(\Delta_{\kappa 1,2}, \ldots, \Delta_{\kappa 1,\kappa_\kappa}) & \cdots & \operatorname{diag}(\Delta_{\kappa\kappa,2}, \ldots, \Delta_{\kappa\kappa,\kappa_\kappa}) \end{bmatrix}.$$

As a consequence of Assumption 5.2.1, all solutions of systems (5.1), coupled via (5.2), converge to a positively invariant set $\Omega$. Hence, without losing generality, we can assume that the initial conditions satisfy $x_{t_0} \in \Omega$, implying $x_t \in \Omega$ for all $t \geq t_0$. Meanwhile, $\|A_2\|$ and $\|A_3\|$ depend on $\Delta$, in such a way that they can be uniformly bounded for all $x_t \in \Omega$. Therefore, we can assume that the perturbation term is also bounded,

$$\|\Xi(\mathscr{X}_r(t), \mathscr{X}_r(t-\tau), \Delta)\| \leq \gamma, \tag{5.17}$$

where $\gamma = \zeta(\delta)$ and function $\zeta$ belongs to class $\mathscr{K}$. Furthermore, there exists a bounded set $\Omega_r$, independent of $\delta$, such that $x_t \in \Omega$ implies $\mathscr{X}_r \in \Omega_r$.

In summary, the perturbation on the coupling introduces an additional bounded term in the equation for the synchronization error dynamics. The term becomes zero if and in general only if the perturbation becomes zero. Moreover, due to this term, the zero point is generally not an equilibrium of the synchronization error dynamics. All the above allows to connect the problem of practical partial synchronization with practical stability of the error system, in the sense of Definition 1 in [97].

**Definition 5.3.2.** *Given a partition $\mathscr{P}$, permutation matrix $\Pi$ and perturbation bound $\gamma$, the network is practically partially synchronized with the clustering represented by $\mathscr{P}$ if system* (5.12) *with $\mathscr{X}_r \in \Omega_r$ and the initial data $\mathscr{X}_0 = \Psi$, is practically stable, i.e., there exist functions $\beta(\gamma) \in \mathscr{K}$ and $T = T(\gamma, \Psi) \geq 0$ such that, whenever $\|\Xi(t)\| \leq \gamma$ and $\mathscr{X}_r(t) \in \Omega_r$ for all $t \geq 0$, we have*

$$\|\mathscr{X}(\Psi)(t)\| \leq \beta(\gamma), \ \ \forall \Psi \in \mathscr{C}([-\tau, 0], \mathbb{R}^{(N-\kappa)n}), \ \ \forall t \geq T(\gamma, \Psi). \tag{5.18}$$

In the following section, tractable sufficient conditions for practical stability of (5.12) are derived, accompanied by the explicit construction of a function $\beta$. It is important to note that due to the "additive perturbation" $\Xi$ in (5.12), the synchronization error

$\mathscr{X}(t) \equiv 0$ is not invariant under the dynamics (5.12). As a distinctive feature of this work, this implies that in the stability analysis, linearization around the zero equilibrium, which is a cornerstone in most works on exact (partial) synchronization, is no longer possible.

## 5.4 Conditions for practical partial synchronization

In this section, sufficient conditions for practical partial synchronization are derived through the stability analysis of the synchronization error dynamics. The Lyapunov-Krasovskii theorem is used for the stability analysis and the obtained conditions are expressed in the form of LMIs.

As mentioned earlier, the solutions of the coupled systems (5.1), (5.2) are bounded, that is, $x \in \Omega$ with $\Omega = \{x | \|x\| \leq r, r > 0\}$. Hence, for every cluster, we have $x_{i,j} \in \Omega_x, i = 1, \ldots, \kappa, j = 1, \ldots, \kappa_i$ with $\Omega_x = \{x_{i,j} | \|x_{i,j}\| \leq r_x, r_x > 0\}$ being a bounded set induced by $\Omega$. Additionally, the functions $f_i$ are sufficiently smooth. Therefore, a Lipschitz condition[1] is satisfied for each component of $F(\mathscr{X}(t), \mathscr{X}_r(t))$,

$$\|\tilde{f}_i(x_{i,j} + e_{i,j}) - \tilde{f}_i(x_{i,j})\| \leq L_{i,j}\|e_{i,j}\|, \forall x_{i,j} + e_{i,j}, x_{i,j} \in \Omega_x, \tag{5.19}$$

where $L_{i,j} \in \mathbb{R}_+$ and $\Omega_x = \{x_{i,j} | \|x_{i,j}\| \leq r_x, r_x > 0\}$.

Combining (5.19) with the structure of (5.8), we can also conclude that there exists an $L \in \mathbb{R}_+$ such that for all possible $\mathscr{X}_r(t) \in \Omega_r$ and $\mathscr{X}(t) \in \Omega$,

$$\|F(\mathscr{X}, \mathscr{X}_r)\| \leq L\|\mathscr{X}\|. \tag{5.20}$$

Using the above Lipschitz condition, we extend the method in [97] to analyze the practical stability of the time-delay system (5.12) which includes a nonlinear component $F(\mathscr{X}(t), \mathscr{X}_r(t))$. In [97], only linear time-delay systems are considered. Through this stability analysis, the conditions for practical partial synchronization are derived, which also yield an explicit expression of $\mu$ as a $\mathscr{K}$ function of $\gamma$, as stated in the theorem below.

**Theorem 5.4.1.** *[94] Consider a network of systems* (5.1)*,* (5.2) *represented by the graph* $\mathscr{G} = (\mathscr{V}, \mathscr{E}, A)$*. For a partition* $\mathscr{P}$*, if there exist positive definite matrices* $P, Q \in \mathbb{R}^{(N-\kappa)n \times (N-\kappa)n}$ *and a positive constant* $\sigma$ *such that the inequalities*

$$\mathscr{M}(P, Q) + 2\sigma\mathscr{N}(P) < 0, \tag{5.21}$$

[1] A function $f : \Omega \to \mathbb{R}^n$ satisfies a Lipschitz condition if there exists an $L \in \mathbb{R}_+$ such that $\|f(x) - f(y)\| \leq L\|x - y\|, \forall x, y \in \Omega$. A Lipschitz condition is satisfied for any smooth function $f(x), x \in \Omega$ with $\Omega$ being a compact set.

$$L < \frac{\lambda_{\min}(P)}{\lambda_{\max}(P)}\sigma, \tag{5.22}$$

*hold, where*

$$\mathscr{M}(P,Q) = \begin{bmatrix} -kPA_0 - kA_0^\top P + Q & kPA_1 \\ kA_1^\top P & -e^{-2\sigma\tau}Q \end{bmatrix}, \quad \mathscr{N}(P) = \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix}, \tag{5.23}$$

*and L is the Lipschitz gain defined in* (5.20)*, then the error system* (5.12)*, corresponding to this partition, is practically stable, i.e., the network is practically partially synchronized with respect to* $\mathscr{P}$*.*

Furthermore, the functions $\beta$ and $T$ in (5.18) can be constructed from the solutions of (5.21) - (5.23), see Appendix E.

The proof of Theorem 5.4.1 is given in Appendix E. In the proof, the Lyapunov-Krasovskii method is used with the following Lyapunov-Krasovskii functional

$$v(\mathscr{X}_t) = \mathscr{X}^\top(t)P\mathscr{X}(t) + \int_{-\tau}^{0} \mathscr{X}^\top(t+\theta)e^{2\sigma\theta}Q\mathscr{X}(t+\theta)\mathrm{d}\theta. \tag{5.24}$$

We refer to Appendix E for the details.

Although the LMIs are not very transparent, in the sense that they do not explicitly reveal a relation between practical partial synchronization and the network parameters ($k$ and $\tau$), some conclusions can still be drawn by looking into the structure of the LMIs. For exact partial synchronization, it has been shown in [1] that there exist positive constants $\bar{k}$ and $\bar{\rho}$ such that if $k > \bar{k}$ and $k\tau < \bar{\rho}$, the exact partial synchronization manifold is stable, as illustrated in Figure 5.6. Similar conclusions are also expected for practical partial synchronization. First, $k$ should be large enough to achieve practical partial synchronization. From the term $-kPA_0 - kA_0^\top P + Q$ at the upper-left corner of $\mathscr{M}(P,Q)$, we can see that a larger $k$ can contribute to the negative definiteness of $\mathscr{M}(P,Q)$. Second, $k$ should also be bounded. For $\mathscr{M}(P,Q)$ to be negative definite, the Schur complement of $-e^{-2\sigma\tau}Q$ should be also negative definite, *i.e.*,

$$-kPA_0 - kA_0^\top P + Q + k^2 PA_1 e^{2\sigma\tau}Q^{-1}A_1^\top P < 0. \tag{5.25}$$

Note that the last term contains $k^2$. Therefore, when $k$ is too large, the last term becomes dominant, thus making the above inequality difficult to fulfill. Third, $\tau$ can be small. A smaller $\tau$ helps the term $-e^{-2\sigma\tau}Q$ in $\mathscr{M}(P,Q)$ to remain negative definite. Finally, large $\tau$ is not desired. Clearly, for large $\tau$, $-e^{-2\sigma\tau}Q$ goes to zero, thus making the LMIs (5.21) difficult to be satisfied.

In the derivation of the conditions in Theorem 5.4.1, we have included information about the nonlinearity of the functions $f_i$ into a single Lipschitz gain which needs

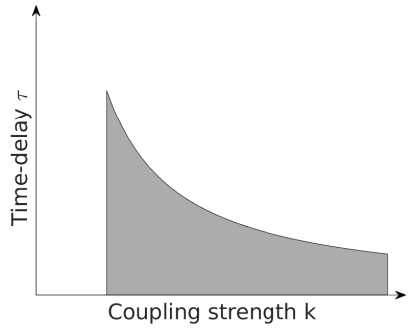



Figure 5.6: Partial synchronization region (gray area) in the $(k,\tau)$ space [1].

to be compensated by the coupling. This approach facilitates the formulation of conditions, however, it also leads to the additional restriction on the upper bound of $L$ in (5.22), which indicates that Theorem 5.4.1 is only applicable for systems with mild nonlinearities. To reduce the restriction, another theorem is proposed where more information on (the boundedness of) the nonlinearity is exploited. To this end, we introduce the bounds of the nonlinearity componentwise such that the boundedness of the nonlinearity is captured by a series of bounds instead of a single constant $L$.

The vectors $\mathscr{X}(t), \mathscr{X}_r(t)$ and $F(\mathscr{X}(t), \mathscr{X}_r(t))$ are reformulated as follows:

$$\mathscr{X}(t) = \begin{bmatrix} \mathscr{X}_1(t) \\ \vdots \\ \mathscr{X}_{Nn-\kappa n}(t) \end{bmatrix}, \mathscr{X}_r(t) = \begin{bmatrix} \mathscr{X}_{r,1}(t) \\ \vdots \\ \mathscr{X}_{r,Nn-\kappa n}(t) \end{bmatrix},$$

$$F(\mathscr{X}(t), \mathscr{X}_r(t)) = \begin{bmatrix} \psi_1(\mathscr{X}_1(t), \mathscr{X}_r(t)) \\ \vdots \\ \psi_{Nn-\kappa n}(\mathscr{X}_{Nn-\kappa n}(t), \mathscr{X}_{r,Nn-\kappa n}(t)) \end{bmatrix},$$

where $\mathscr{X}_\ell(t), \mathscr{X}_{r,\ell}(t) \in \mathbb{R}$ and $\psi_\ell : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$, $\ell = 1,\ldots,Nn-\kappa n$.

Since the functions $f_i$ are sufficiently smooth and the solutions $x_i(t)$ are bounded, there exist $F_\ell^-$ and $F_\ell^+$ such that

$$F_\ell^- \leq \frac{\psi_\ell(\mathscr{X}_\ell(t), \mathscr{X}_{r,\ell}(t))}{\mathscr{X}_\ell(t)} \leq F_\ell^+, \text{ for } \ell = 1,\ldots,Nn-\kappa n, \tag{5.26}$$

or equivalently,

$$(\psi_\ell(\mathscr{X}_\ell(t), \mathscr{X}_{r,\ell}(t)) - F_\ell^- \mathscr{X}_\ell(t))(\psi_\ell(\mathscr{X}_\ell(t), \mathscr{X}_{r,\ell}(t)) - F_\ell^+ \mathscr{X}_\ell(t)) \leq 0. \tag{5.27}$$

With $F_\ell^+$ and $F_\ell^-$ specified, another set of conditions for practical partial synchronization is provided in the theorem below.

**Theorem 5.4.2.** *[94] Consider a network of systems* (5.1), (5.2) *represented by the graph* $\mathscr{G} = (\mathscr{V}, \mathscr{E}, A)$. *For a partition* $\mathscr{P}$, *if there exist symmetric positive definite matrices* $P$, $Q$, $Z$, $W \in \mathbb{R}^{(N-\kappa)n\times(N-\kappa)n}$, *a positive definite diagonal matrix* $V \in \mathbb{R}^{(N-\kappa)n\times(N-\kappa)n}$, *and a positive constant* $\sigma$ *such that the inequality*

$$\Theta + H_1 P H_2 + H_2^\top P H_1^\top + \tau^2 H_2^\top Z H_2 < 0 \tag{5.28}$$

*holds, where*

$$\Theta = \begin{bmatrix} 2\sigma P + Q - e^{-2\sigma\tau}Z - F_1 V & e^{-2\sigma\tau}Z & F_2 V & 0 \\ * & -e^{-2\sigma\tau}Q - e^{-2\sigma\tau}Z & 0 & 0 \\ * & * & -V & 0 \\ * & * & * & -W \end{bmatrix},$$

$$\begin{aligned} H_1 &= \begin{bmatrix} I & 0 & 0 & 0 \end{bmatrix}^\top, \\ H_2 &= \begin{bmatrix} -kA_0 & kA_1 & I & I \end{bmatrix}, \\ F_1 &= \operatorname{diag}\{F_1^- F_1^+, F_2^- F_2^+, \cdots, F_{Nn-\kappa n}^- F_{Nn-\kappa n}^+\}, \\ F_2 &= \operatorname{diag}\left\{\frac{F_1^- + F_1^+}{2}, \frac{F_2^- + F_2^+}{2}, \cdots, \frac{F_{Nn-\kappa n}^- + F_{Nn-\kappa n}^+}{2}\right\}, \end{aligned} \tag{5.29}$$

*then the corresponding system* (5.12) *is practically stable. That is, the network is practically partially synchronized with respect to* $\mathscr{P}$. *Furthermore, the functions* $\beta$ *and* $T$ *in (5.18) can be chosen as:*
*1) for* $\|\Psi\|_W \le \frac{\gamma}{\sqrt{2\sigma\alpha_2\lambda_{\min}(W)}}$,

$$\mu = \beta(\gamma) = \frac{\gamma}{\sqrt{2\sigma\alpha_1\lambda_{\min}(W)}}, \quad T = 0; \tag{5.30}$$

*2) for* $\|\Psi\|_W > \frac{\gamma}{\sqrt{2\sigma\alpha_2\lambda_{\min}(W)}}$,

$$\mu = \beta(\gamma) = \frac{k_\mu \gamma}{\sqrt{2\sigma\alpha_1\lambda_{\min}(W)}}, \quad T = \frac{1}{2\sigma}\ln\left(\frac{2\sigma\alpha_2\|\Psi\|_W^2 - \gamma^2\lambda_{\max}(W)}{2\sigma\alpha_1\mu^2 - \gamma^2\lambda_{\max}(W)}\right), \tag{5.31}$$

*with* $1 < k_\mu < \frac{1}{\gamma}\sqrt{\frac{2\sigma\alpha_2}{\lambda_{\max}(W)}}\|\Psi\|_s$.
*Here,* $\|\Psi\|_W$ *is the norm of the initial data* $\|\Psi(\theta)\|_W := \max\{\|\Psi(\theta)\|_s, \|\dot{\Psi}(\theta)\|_s\}$ *as defined in* (2.7), *and* $\alpha_1 = \lambda_{\min}(P)$, $\alpha_2 = \lambda_{\max}(P) + \tau\lambda_{\max}(Q) + \frac{\tau^3}{2}\lambda_{\max}(Z)$. *The symbol* $*$ *denotes the symmetric terms in symmetric matrices.*

The proof of Theorem 5.4.2 is given in Appendix F. As with the proof of Theorem 5.4.1, the Lyapunov-Krasovskii method is used. A Lyapunov Krassovskii functional of

the following form is used:

$$\begin{aligned} v(\mathscr{X}_t) = &\mathscr{X}^\top(t)P\mathscr{X}(t) + \int_{t-\tau}^{t} \mathscr{X}^\top(\theta)e^{2\sigma(\theta-t)}Q\mathscr{X}(\theta)\mathrm{d}\theta \\ &+ \tau\int_{-\tau}^{0}\int_{t+s}^{t} \dot{\mathscr{X}}^\top(\theta)e^{2\sigma(\theta-t)}Z\dot{\mathscr{X}}(\theta)\mathrm{d}\theta\mathrm{d}s. \end{aligned} \tag{5.32}$$

We refer to Appendix F for the details.

As with Theorem 5.4.1, similar conclusions on the parametric dependence of the practical partial synchronization can also be drawn from Theorem 5.4.2. We focus on the block at the upper-left corner of the matrix $\Theta + H_1PH_2 + H_2^\top PH_1^\top + \tau^2 H_2^\top ZH_2$, given by

$$-kPA_0 - kA_0^\top P + \tau^2 k^2 A_0^T ZA_0 + 2\sigma P + Q - e^{-2\sigma\tau}Z - F_1V. \tag{5.33}$$

For the LMIs to be feasible, it is desired for this matrix to be negative definite. Apparently, we can see that a sufficiently large $k$ helps the matrix (5.33) to be negative definite due to the part $-kPA_0 - kA_0^\top P$. However, due to the part $\tau^2k^2A_0^TZA_0$, $k\tau$ should be bounded such that (5.33) can be negative definite.

**Remark.** *Similar as in [97], an "exponential estimate" of the synchronization errors $\mathscr{X}$ is provided when deriving these two theorems, which indicates how fast solutions of the coupled systems converge to the set $\mathscr{M}_\mu$. The constant $\sigma$ is related to the exponential decay rate estimate. A larger value of $\sigma$ indicates a faster decay of the synchronization errors. In addition, as can be seen from the expressions of $\beta(\gamma)$, $\sigma$ is also inversely related to the amplification of the perturbation. Thus, a larger value of $\sigma$ results in a smaller bound on the synchronization errors $\mu$. However, the LMIs* (5.21) *and* (5.28) *may become infeasible for a large $\sigma$, due to the presence of such terms as $2\sigma\mathscr{N}(P)$, $-e^{-2\sigma\tau}Q$ in* (5.21) *and $2\sigma P$, $-e^{-2\sigma\tau}Z$ in* (5.28)*, respectively. Furthermore, because of the presence of the terms $-e^{-2\sigma\tau}Q$ and $-e^{-2\sigma\tau}Z$, a smaller value of time-delay $\tau$ is required for a larger value of $\sigma$. Therefore, there exists a trade-off when choosing the value of $\sigma$. A practical way to handle this trade-off, is to start with a small $\sigma$ for the sake of the LMIs feasibility, and then increase it incrementally to a desired value (e.g., for a small enough bound of the synchronization errors $\mu$) for which the LMIs are still feasible with an acceptable time-delay $\tau$.*

## 5.5 Numerical example

This section revisits Example 5.2.2 presented in Section 5.2 with the focus on the perturbed network, which is analyzed by using the results from the previous section.

The network is shown in Figure 5.7. Recall that the adjacency matrix is perturbed with $\Delta A$ such that

$$A = A_n + \Delta A = \begin{bmatrix} 0 & 2 & 0.1 & 1 \\ 4 & 0 & 2 & 0.3 \\ 0.1 & 1 & 0 & 2 \\ 3 & 0.3 & 3 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0.2 & 0 & 0 \\ 0.2 & 0 & 0 & 0.05 \\ 0 & 0 & 0 & 0 \\ 0.1 & 0 & 0.2 & 0 \end{bmatrix}. \tag{5.34}$$

It can be proved that the systems (5.5) of the network are semipassive, see Appendix G.

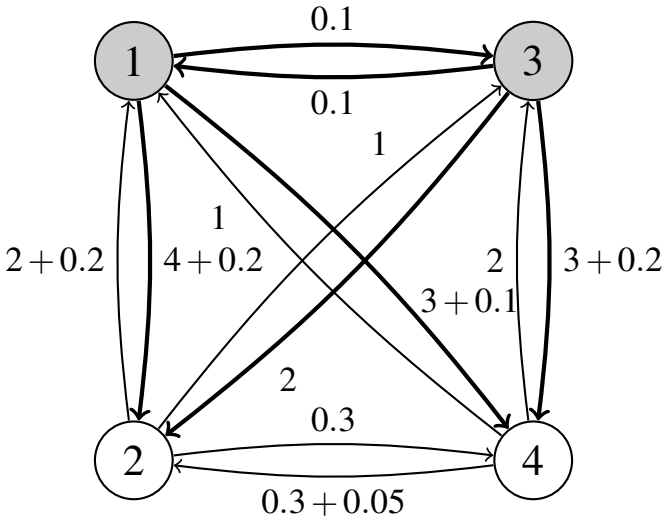



Figure 5.7: The network of Example 5.2.2 with adjacency matrix $A$

Recall that for the nominal adjacency matrix $A_n$, partition $\mathscr{P} = \{\{1,3\},\{2,4\}\}$ is a viable partition. However, due to the perturbation $\Delta A$, no viable partition exists for this network with $A$. The corresponding synchronization error dynamics of form (5.12) are derived accordingly by considering the practical partial synchronization of nodes $1,3$ in one cluster and nodes $2,4$ in another cluster. The key matrices $A_0$ and $A_1$, which are required to construct the LMIs (5.28) in Theorem 5.4.2, are given by

$$\begin{aligned} A_0 &:= \begin{bmatrix} \rho_1 BC & 0 \\ 0 & \rho_2 BC \end{bmatrix} - \tfrac{1}{k} \begin{bmatrix} A_s & 0 \\ 0 & A_s \end{bmatrix}, \\ A_1 &:= A_{\text{red}} \otimes BC, \end{aligned}$$

where

$$\rho_1 = 3.3,\ \ \rho_2 = 6.55,$$
$$A_s = \begin{bmatrix} 0.2 & 1 \\ -1 & 0.2 \end{bmatrix},\ B = C^\top = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ A_{\text{red}} = \begin{bmatrix} -0.1 & 1 \\ 1.2 & -0.35 \end{bmatrix}.$$

Note that the matrix $A_s$ represents the linear components of the systems (5.5). The matrix $A_0$ is extended to include these linear components.

For a given pair $(k,\tau)$ Theorem 5.4.2 can be used to check if the network is practically partially synchronized in two steps: 1) estimate the values of $F_\ell^+$ and $F_\ell^-$ by using numerical simulations of the network; 2) check the feasibility of the LMIs in Theorem 5.4.2 for a given $\sigma$. As mentioned in the previous sections, the states of the network dynamics converge to a positively invariant set in finite time. Note that the value of this finite time depends on the initial data. For example, when the systems are initialized far away from the invariant set, it takes longer to reach the set. In order to reduce conservatism, at the first step, only the data after some time are used for estimation. Here, we use the data after 600 time units for estimation. In particular, we find that the LMIs (5.28) are feasible for $k = 0.3$, $\tau = 2$ with $\sigma = 0.22$ by using the MATLAB toolboxes of SDPT3 ([73]) and YALMIP ([98]). First, with this pair of $(k,\tau)$, we construct $F_1$ and $F_2$ using the estimated values of $F_i^+$ and $F_i^-$ given by

$$\begin{bmatrix} F_1^+ \\ F_2^+ \\ F_3^+ \\ F_4^+ \end{bmatrix} = \begin{bmatrix} -0.0064 \\ 0.0009 \\ 1.4035 \\ 0.0372 \end{bmatrix},\ \begin{bmatrix} F_1^- \\ F_2^- \\ F_3^- \\ F_4^- \end{bmatrix} = \begin{bmatrix} -0.0480 \\ -0.0255 \\ -0.0963 \\ -0.1279 \end{bmatrix}. \tag{5.35}$$

Now, the values of $H_1$, $H_2$, $F_1$, $F_2$, $\tau$, and $k$ are available for solving the LMIs. Second, with the help of the YALMIP and SDPT3 toolboxes, the LMIs in (5.28) are cast into an optimization problem with the matrices $P$, $Q$, $Z$, $W$, and $V$ as design variables, then MATLAB automatically constructs the matrices if it finds that such matrices exist for a given $\sigma$. To find the maximum possible value of $\sigma$, the second step is repeated for a series of values of $\sigma$ starting from 0.01 with an incremental step of 0.01. It turns out that until $\sigma = 0.22$, these matrices exist for $k = 0.3$, $\tau = 2$ and their values are presented in Appendix H. It is important to point out that for these values of $(k,\tau)$ the zero solution ($x_i = 0$) of the coupled network dynamics is (locally) exponentially unstable, which can be verified by checking the spectral abscissa (rightmost eigenvalues) of the linearized dynamics of the whole coupled network. Notice that full synchronization will not emerge as there is no full synchronization manifold.

As an illustration of the obtained results, a simulation of the network with $k = 0.3$, $\tau = 2$, and the adjacency matrix $A$ defined in (5.34) is performed, using the *dde23* solver from MATLAB with relative and absolute tolerance of $1e^{-3}$ and $1e^{-6}$, respectively. The simulation runs over 1000 time units. Unlike in Section 5.2, initial data outside the partial synchronization manifold are used in this simulation such that we can also

investigate the stability of partial synchronization. The values of the initial data used are $x_1(\theta) = [0.0170\ 0.0365]^\top$, $x_2(\theta) = [0.0035\ 0.0190]^\top$, $x_3(\theta) = [0.0208\ 0.0462]^\top$, $x_4(\theta) = [0.0283\ 0.0378]^\top$, $-\tau \leq \theta \leq 0$, which are randomly chosen in the interval of $[0, 0.05]$.

The evolution of the states $x_{i,1}$, $i = 1,2,3,4$ is depicted in Figure 5.8 in the time interval $[0, 100]$ (before partial synchronization is fully established) and Figure 5.9 in the time interval $[990, 1000]$ (after partial synchronization is fully established). As can be seen from those figures, $x_{1,1}$ and $x_{3,1}$, respectively, $x_{2,1}$ and $x_{4,1}$ converge to each other (up to a barely visible tolerance). In addition, Figure 5.9 also shows the solutions converge to a limit cycle, which is induced by the nonlinear terms $-10x_{i,1}(t)(x_{i,1}^2(t) + x_{i,2}^2(t))$ and $-10x_{i,2}(t)(x_{i,1}^2(t) + x_{i,2}^2(t))$ in (5.5). To compare exact synchronization with practical synchronization, we have also simulated the network with the same values of $(k, \tau)$ and the nominal adjacency matrix $A_n$. The synchronization errors $e_1 = x_{3,1} - x_{1,1}$, $e_2 = x_{3,2} - x_{1,2}$, $e_3 = x_{4,1} - x_{2,1}$ and $e_4 = x_{4,2} - x_{2,2}$ of the networks with $A$ and $A_n$ are shown in Figure 5.10 and Figure 5.11. As in Figure 5.5, the errors are scaled by the maximum magnitudes of the relevant variables in these figures. Clearly, the synchronization errors in the network with $A$ are larger than those in the network with $A_n$. Besides, the figures also indicate that the practically synchronized states only converge to each other up to some bound, but cannot become identical in the network with $A$.

To demonstrate the relation between the size of the perturbation and the bound of the synchronization errors, we run a series of the simulations of this network with the same $k$ and $\tau$, but with a series of different adjacency matrices defined below

$$A = A_n + k_\Delta \cdot \Delta A, \tag{5.36}$$

with $A_n$, $\Delta_A$ defined in (5.34), and $k_\Delta = 0, 0.05, 0.10, \ldots, 0.5$.
The relation between $\|e_i\|$ and $\delta = \|k_\Delta \Delta A\|_\infty$ obtained from these simulations is shown in Figure 5.12. Here, the signals $x_{1,1}$, $x_{2,1}$, $x_{1,2}$, $x_{2,2}$, $x_{3,1}$ and $x_{4,1}$ are sampled in the time interval $[990, 1000]$ to compute the synchronization errors $e_i$. Figure 5.12 shows that the synchronization error grows with the size of perturbation, and it converges to zero when the perturbation goes to zero.

## 5.6 Conclusions

In this chapter, practical partial synchronization of delay-coupled systems is studied. First, the synchronization error dynamics are separated from the whole network dynamics for the case where there are perturbations on the coupling, which render exact synchronization impossible. Second, the notion of practical partial synchronization is introduced and sufficient conditions, formulated in terms of LMIs, are provided. The

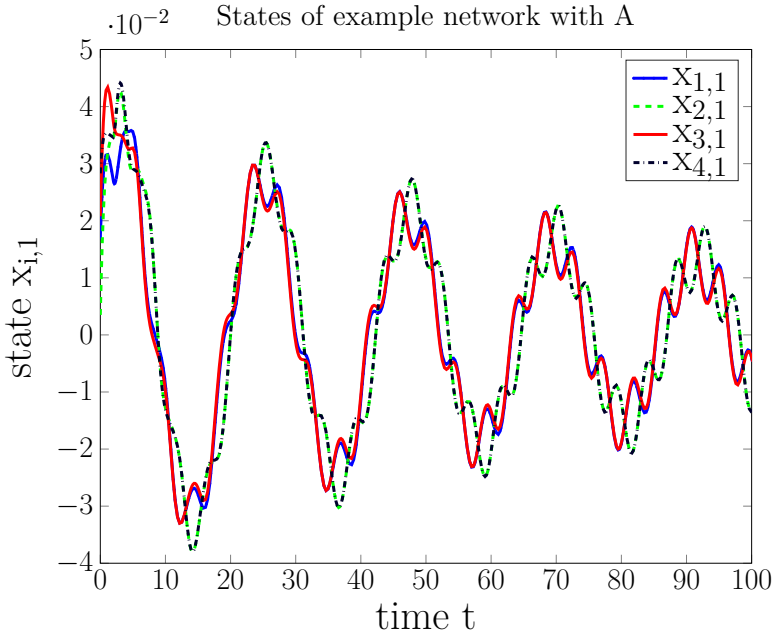



Figure 5.8: The states $x_{i,1}$ of the network with $k = 0.3, \tau = 2$ in the time interval $[0, 100]$

relation between the bounds on the synchronization errors and those of the perturbations is made explicit. From a methodological point of view, a main challenge in the analysis is that the zero solution is no longer an equilibrium point of the synchronization error system. Finally, a numerical example is presented, where the conditions for a network of nonlinear systems to be practically partially synchronized are derived by using one of the theorems proposed in this chapter.

The study of practical partial synchronization in this chapter is under the assumption of perturbations of the adjacency matrix. It is worth noting that this methodology can also be applied to other situations where the bounded perturbations or uncertainties are present in other components of the network that lead to a violation of the existence conditions for the partial synchronization manifold under consideration (for instances, perturbations of the vector fields $f_i$, input matrices $B_i$, output matrices $C_i$). In these cases, one can use some small tolerance when comparing the network components. If the differences between these components are within the tolerance, they can be considered as identical, thus potential partial synchronization patterns can be identified. Once a partial synchronization pattern is identified, a system in each cluster can be appointed as a nominal system such that the differences between the nominal system

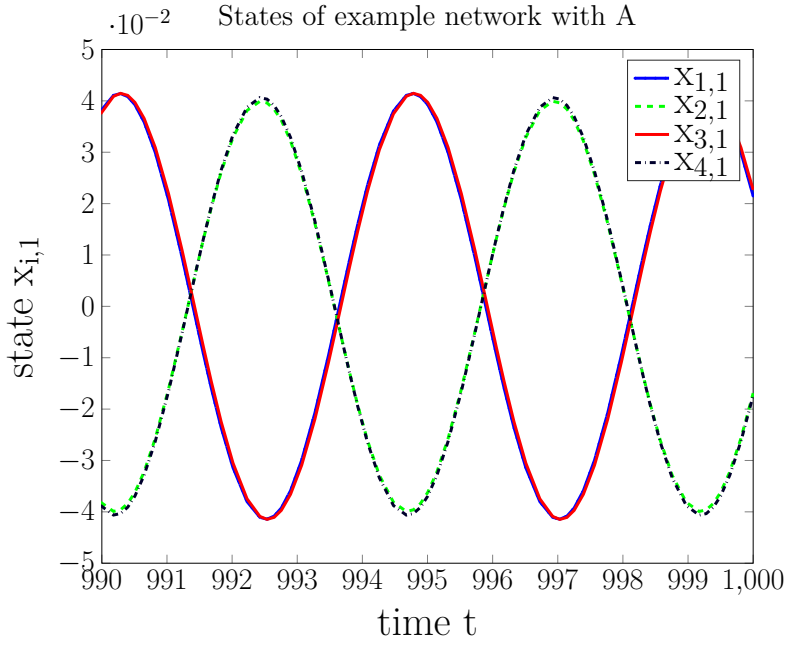



Figure 5.9: The states $x_{i,1}$ of the network with $k = 0.3, \tau = 2$ in the time interval $[990, 1000]$

and the other systems in each cluster are minimized. After that, one can use the same procedure presented in this chapter to obtain the conditions for practical partial synchronization.

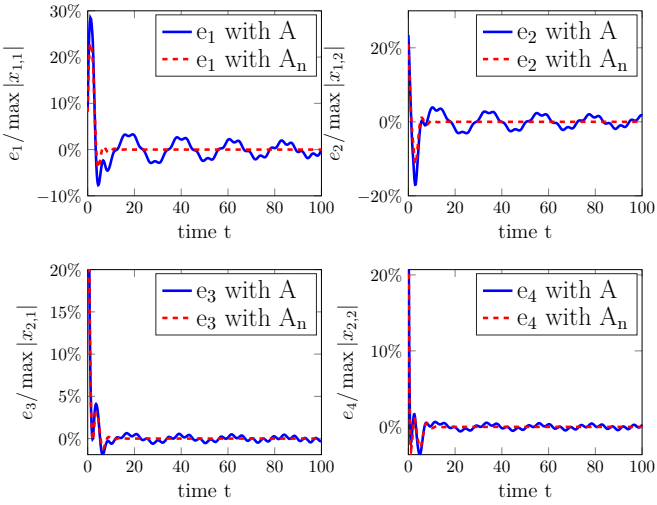



Figure 5.10: Synchronization errors of the network with $A$ (practical synchronization, blue colored) and $A_n$ (exact synchronization, red colored) in the time interval $[0, 100]$

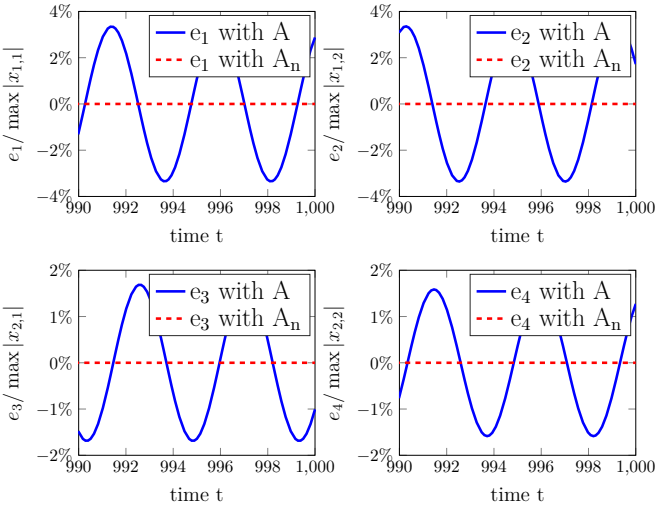



Figure 5.11: Synchronization errors of the network with $A$ (practical synchronization, blue colored) and $A_n$ (exact synchronization, red colored) in the time interval $[990, 1000]$

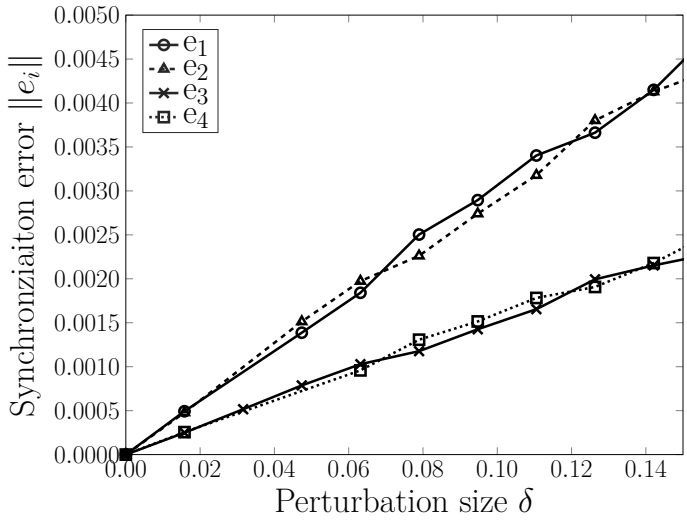



Figure 5.12: Synchronization errors with different sizes of perturbations

# Chapter 6

# Experimental synchronization of delay-coupled neurons

**Abstract:** This chapter aims to demonstrate the theoretical results on partial synchronization that are presented in Chapter 4 via experiments with networks of electronic neurons. The electronic neurons are the implementations of the Hindmarsh-Rose neuron model in circuit boards. First, some basic concepts of neurons and the Hindmarsh-Rose model are introduced. Then, results of the experiments of delay-coupled Hindmarsh-Rose neurons are presented along with the theoretical results.

## 6.1 Introduction

Synchronous oscillations of neurons have been observed in many parts of the nervous system, especially in its center, the brain. Such synchrony is reported to underlie specific brain functions, such as memory formulation [99] and language comprehension [100], and even disorders, for example, epilepsy and Parkson's disease [32]. Therefore, it is important to understand when and which patterns of such synchronous behaviors occur in neuronal networks.

The fundamental unit of a neuronal network is a neuron. A neuron can be considered as a dynamical system with the membrane potential as the output and the input current as the input. The membrane potential of a neuron is the difference in electric potential between the interior and exterior of the neuron. The activities of a neuron can be categorized into three different cases by how the membrane potential is changing. First, the membrane potential can be constant over time, called *resting*. Second, the neuron

can fire action potential at a constant rate, called *spiking*. An action potential is a rapid rise and subsequent fall of the neuron's membrane potential [101]. Third, the neuron may produce action potentials between relatively long periods of quiescence, called *bursting*. In this chapter, the Hindmarsh-Rose model is used which is able to mimic all these behaviors of a neuron. It has been also proved that all models of a neuron, including the Hindmarsh-Rose neuron model, are semipassive [93]. Another advantage of the Hindmarsh-Rose model is that it can be easily implemented on an electrical circuit board. The Hindmarsh-Rose neuron is a three-dimensional model of the following form [92]:

$$\begin{aligned} \dot{y}(t) &= -ay^3(t) + by^2(t) + z_1 - z_2 + E + u(t), \\ \dot{z}_1(t) &= c - dy^2(t) - z_1(t), \\ \dot{z}_2(t) &= r(s(y(t) + y_0) - z_2(t)), \end{aligned} \tag{6.1}$$

where $y \in \mathbb{R}$ is the membrane potential, $z_1, z_2 \in \mathbb{R}$ are internal variables, $u \in \mathbb{R}$ is the input, and $a, b, c, r, s \in \mathbb{R}_+$, $E, y_0 \in \mathbb{R}$ are constant parameters. Depending on the choice of the parameters, the model can produce different types of behaviors.

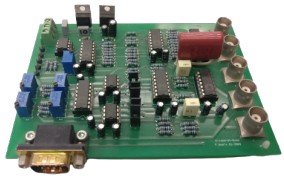
Another important component in networks of neurons is the communication interface. The most common communication interface between neurons is called a synapse. Two different types of synapses exist: *chemical synapses* and *electrical synapses* [102]. In a chemical synapse, the presynaptic neuron converts the electrical activity into the release of a chemical called neurotransmitter which induces a synaptic current at the postsynaptic neuron [103]. In an electrical synapse, the presynaptic and postsynaptic neuron membranes are connected by *gap junctions* which induce a direct high conductance pathway between the presynaptic and postsynaptic neurons [103]. In other words, the synaptic current in an electrical synapse is in the form of $g \cdot (V_1 - V_2)$, with $g$ being the synaptic conductance, $V_1$ being the membrane potential of the presynaptic neuron and $V_2$ being the membrane potential of the postsynaptic neuron. Electrical synapses play an important role in the synchronization of neurons [32]. It worth noting that the electrical synapse can be modeled by the diffusive coupling introduced in the previous chapters. By experimenting with a network of Hindmarsh-Rose neurons interconnected via diffusive coupling, this chapter mainly aims to demonstrate the theoretical results obtained in Chapter 4. In addition, the experimental results provide some insights into the collective behaviors in neuronal networks.

The structure of this chapter is as follows. Section 6.2 briefly introduces the experimental setup. In Section 6.3, the experimental results on networks of electronic Hindmarsh-Rose neurons with different network settings are presented, accompanied by the theoretical results using the method in Chapter 4 for comparison. Finally, Section 6.4 provides conclusions.

## 6.2 Experimental setup

In this section, the experimental setup is briefly introduced. More details about the setup can be found in [104].

The experimental setup used in this chapter consists of up to sixteen circuit realizations of Hindmarsh-Rose neurons and a coupling interface, see Figure 6.1.



(a) Electronic Hindmarsh-Rose neuron.



(b) Coupling interface [105].

Figure 6.1: The experimental setup. The electronic Hindmarsh-Rose neuron (a) and the coupling interface (b)

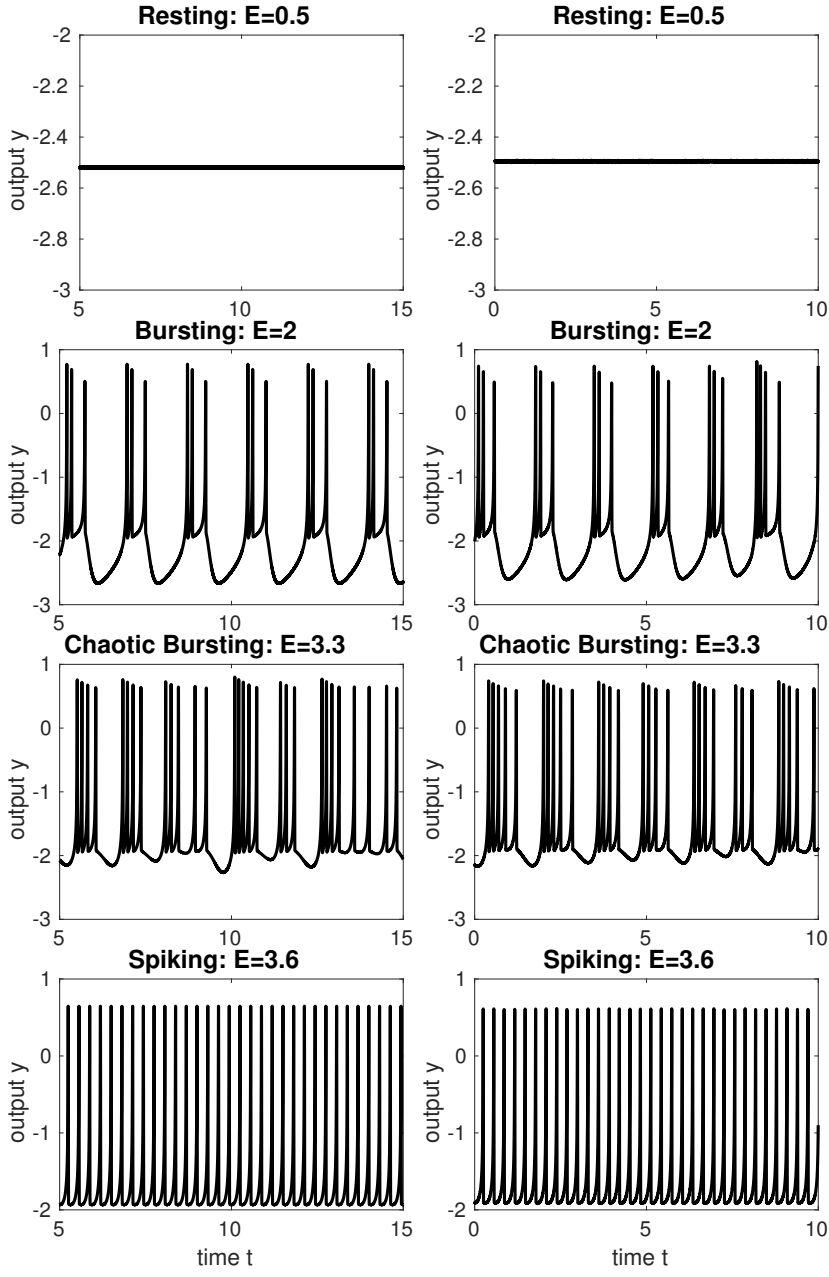
The dynamics of each electronic Hindmarsh-Rose neuron is governed by the following equations:

$$\begin{aligned}
\dot{z}_{i,1}(t) &= 100(-y_i^2(t) - 2y_i(t) - z_{i,1}(t)),\\
\dot{z}_{i,2}(t) &= 0.5(4y_i(t) + 4.472 - z_{i,2}(t)),\\
y_i(t) &= 100(-y_i^3(t) + 3y_i(t) - 8 + 5z_{i,1}(t) - z_{i,2}(t) + E_i + u_i(t)),
\end{aligned} \tag{6.2}$$

where $y_i(t)$ is the output of the $i$th model neuron, $z_{i,1}(t), z_{i,2}(t)$ are the internal states, and $u_i(t)$ is the input that is used to communicate with other neurons. Depending on the values of $E_i$, the Hindmarsh-Rose neuron can exhibit different types of behaviors, see Figure 6.2. It has been found that the neuron behaves in resting mode when $0 \leq E_i \leq 1.4$, in bursting mode (with 1 to 5 spikes) when $1.4 \leq E_i \leq 3.1$, in chaotic bursting mode when $3.1 \leq E_i \leq 3.5$, and in spiking mode when $E_i \geq 3.5$ [106]. By setting $E_i$ to be different values, we can have different types of neurons.

Note that the model (6.2) is modified from the original Hindmarsh-Rose model (6.1) by redefining the variables $t, z_{i,1}, z_{i,2}$ and $y_i$:

$$t \mapsto 100t,\ z_{i,1}(t) \mapsto \frac{1}{5}(z_{i,1}(t) + 4),\ z_{i,2} \mapsto z_{i,2}(t) - 6,\ y_i(t) \mapsto y_i(t) - 1.$$




(a) Simulation (b) Experiment

Figure 6.2: Different types of neuron behaviors due to different values of the parameter $E_i$ (when $u_i \equiv 0$).

This modification is used to avoid the saturation of signals in the circuit such that off-the-shelf components such as resistors, capacitors, analog voltage multipliers, and operational amplifiers can be used. Every state signal $z_{i,1}$, $z_{i,2}$ or $y_i$ can be measured as a voltage through one of the five coaxial connectors at the right side of the circuit board shown in Figure 6.1(a). Concerning the remaining two coaxial connectors, one is used to set the parameter $E_i$, and the other one will be connected to the coupling interface to get $u_i(t)$.

The coupling interface is connected with the input and output channels of the electronic Hindmarsh-Rose neurons via coaxial cable. The coupling function, network topology, and the time-delays are specified in the coupling interface. The core component of the coupling interface is a 210 MHz Atmel ARM© Thumb©-based AT91SAM9260 microcontroller which is used to compute the coupling inputs $u_i$ based on the outputs $y_i$ using the coupling functions defined in a custom-made software program. In principle, any type of coupling function can be implemented. Considering that time-delays always occur in the signal transmission between real neurons due to finite transmission speed, we focus on invasive coupling in this chapter

$$u_i = k\sum_{j=1}^{N} a_{ij}(y_j(t-\tau)-y_i(t)), i = 1,\ldots,N, \tag{6.3}$$

where $a_{ij}$ is the $ij$th entry of the adjacency matrix $A$, $k$ is the coupling strength, and $\tau$ is the time-delay (transmission delay).

## 6.3 Experimental synchronization

In this section, we present the experimental results on synchronization of delay-coupled Hindmarsh-Rose neurons to support the theoretical results of Chapter 4. A network of four Hindmarsh-Rose neurons coupled via invasive coupling (6.3) with different network topologies and model parameters is considered.

Note that no electrical components are identical in practice. Therefore, every electronic Hindmarsh-Rose neuron is slightly different from each other. Hence, exact synchronization may not be possible. A notion of practical synchronization is needed in this case. One way to define such a form of synchronization is to specify a bound on the differences between the outputs of neurons as in, for example, Chapter 5 and [105]. In [105], two systems are considered to be practically synchronized when their outputs after some time $t_1$ satisfy the condition that $\|y_1(t)-y_2(t)\| \leq \mu, \forall t \geq t_1$ with $\mu$ being a small positive number. However, this definition is not ideal for neuronal systems. When the neurons produce spikes, the differences between the outputs of two synchronized neurons can still be large, which will lead to the choice of a relatively large $\mu$.

In neuroscience, the cross-correlation function is widely used to detect synchronization between neurons [107]. In this chapter, we adopt the cross-correlation based definition of practical synchronization of neurons in [106]. The cross-correlation coefficient of two sampled signals $x = [x_1, x_2, \ldots, x_\ell]$ and $y = [y_1, y_2, \ldots, y_\ell]$ is defined as

$$Co_{x,y} = \frac{\sum_{i=1}^{\ell}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{\ell}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{\ell}(y_i - \bar{y})^2}}, \tag{6.4}$$

where $\bar{x}$ and $\bar{y}$ denote the mean values of the signals $x$ and $y$. Obviously, $Co_{x,y} = Co_{y,x}$. In addition, the value of $Co_{x,y}$ lies in the interval $[-1, 1]$. The coefficient $Co_{x,y} = 1$ indicates a perfect positive correlation, $Co_{x,y} = -1$ indicates a perfect negative correlation, and $Co_{x,y} = 0$ indicates no correlation. The closer $Co_{x,y}$ is to 1 (-1), the stronger the positive (negative) correlation between the variables $x, y$ is. Note that negative correlation implies anti-phase synchronization. In this thesis, synchronization is considered at both magnitude and phase levels. Therefore, only the positive relation is considered in this chapter. By using the definition of cross-correlation coefficient in (6.4), two neurons $i, j$ are considered to be practically synchronized when their sampled outputs $y_i$ and $y_j$ satisfy the following condition

$$Co_{y_i,y_j} \geq 1 - \varepsilon,\ \varepsilon \text{ is a small positive number.} \tag{6.5}$$

Note that the cross-correlation coefficient can only be expected to be close to 1 in practice ($Co_{y_i,y_j} < 1, i \neq j$), which can cause a "transitivity" issue when identifying synchronization patterns. For example, the conditions $Co_{y_1,y_2} \geq 1 - \varepsilon$ and $Co_{y_2,y_3} \geq 1 - \varepsilon$ do not guarantee that $Co_{y_1,y_3} \geq 1 - \varepsilon$. That is, systems 1 and 3 are not necessarily practically synchronized when systems 1 and 2, respectively, systems 2 and 3 are practically synchronized. Therefore, it is important to compare all output signals of a network with more than two nodes to correctly identify synchronization patterns. A more detailed definition on practical partial synchronization of a network of electronic Hindmarsh-Rose neurons is given below.

**Definition 6.3.1.** *A network of N neurons is practically partially synchronized with a synchronization pattern presented by a partition (clustering)* $\mathscr{P} = \{P_1, \ldots, P_\kappa\}$, $1 < \kappa < N$, *when the following holds for all* $\ell = 1, \ldots, \kappa$ *and* $i, j = 1, \ldots, N$*:*

$$Co_{y_i,y_j} \geq 1 - \varepsilon \text{ if and only if } i, j \in P_\ell, \tag{6.6}$$

*where* $y_i, y_j$ *are the sampled outputs of neurons* $i$, $j$, *and* $\varepsilon$ *is a predefined small positive number. In addition, if* $\kappa = 1$, *the network is practically fully synchronized.*

In other words, a network of neurons is considered to be practically partially synchronized with a certain synchronization pattern if only the neurons within the same clusters satisfy condition (6.5).

The value of the threshold $\varepsilon$ is chosen as $\varepsilon = 0.001$, and this choice is made by examining the experimental data, see Section 6.3.1 for more details. In [106], it has been shown that the bounds of synchronization regions vary when this threshold changes. Therefore, it is important to choose a proper threshold such that different synchronization patterns can be characterized. All the output signals in the experiments are sampled at 1 kHz in a time span of 5s.

### 6.3.1 Network with same type of neurons

In what follows, networks consisting of the same type of neurons with different topologies are considered. The neurons are of the same type if the same value of the parameters $E_i$ is adopted. First, a series of experiments are performed in a network with a topology compatible with multiple synchronization manifolds. The influence of the value of $\varepsilon$ on synchronization identification is investigated by examining the obtained experimental data. Second, experiments are performed in a network with a topology compatible with one single synchronization manifold.

#### Multiple partial synchronization manifolds

In this setting, the network of the Hindmarsh-Rose neurons has multiple partial synchronization manifolds. The topology of the network is given by

$$A = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{bmatrix}, \tag{6.7}$$

and the neurons are interconnected via invasive coupling (6.3).

The values of parameter $E_i$ are set to $E_1 = E_2 = E_3 = E_4 = 3.3$ such that the neurons are of the same type, *i.e.*, they have the same (nominal) input-output dynamics. By using the method presented in Chapter 3, four viable partitions are identified: $\mathscr{P}_1 = \{\{1,2,3,4\}\}$, $\mathscr{P}_2 = \{\{1,2\},\{3,4\}\}$, $\mathscr{P}_3 = \{\{1,3\},\{2,4\}\}$ and $\mathscr{P}_4 = \{\{1,4\},\{2,3\}\}$. That is, the network can exhibit four possible synchronization patterns: 1) full synchronization, denoted by $FS$; 2) partial synchronization where neurons 1, 2 synchronize and neurons 3, 4 synchronize, denoted by $PS_1$; 3) partial synchronization where neurons 1, 3 synchronize and neurons 2, 4 synchronize, denoted by $PS_2$; and 4) partial synchronization where neurons 1, 4 synchronize and neurons 2, 3 synchronize, denoted by $PS_3$. In experiments, two neurons are considered to be synchronized if the correlation coefficients of their output signals satisfy condition (6.5). Specially, to identify these

synchronization patterns, a vector $v_c = [c_1\ c_2\ c_3\ c_4\ c_5\ c_6]^\top \in \mathbb{R}^6$ is used with $c_i$ being

$$c_i = \begin{cases} 1 & co_i \geq 1-\varepsilon, \\ 0 & \text{otherwise}, \end{cases}$$

where $[co_1\ co_2\ co_3\ co_4\ co_5\ co_6] = [Co_{y_1,y_2}\ Co_{y_1,y_3}\ Co_{y_1,y_4}\ Co_{y_2,y_3}\ Co_{y_2,y_4}\ Co_{y_3,y_4}]$.

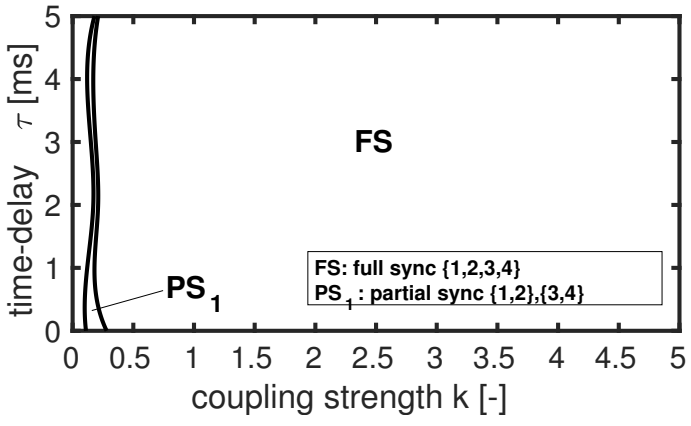
Then, the synchronization patterns can be parameterized by $v_c$, see Table 6.1. For example, synchronization pattern $PS_1$ means that $co_1 = Co_{y_1,y_2} \geq 1-\varepsilon$ (neurons 1, 2 synchronize), $co_6 = Co_{y_3,y_4} \geq 1-\varepsilon$ (neurons 3, 4 synchronize), and $co_i < 1-\varepsilon, i = 2,\ldots,5$ (all the other pairs of neurons do not synchronize), implying that the vector $v_C$ is $v_C = [1\ 0\ 0\ 0\ 0\ 1]^\top$.

Table 6.1: Synchronization patterns and $v_c$

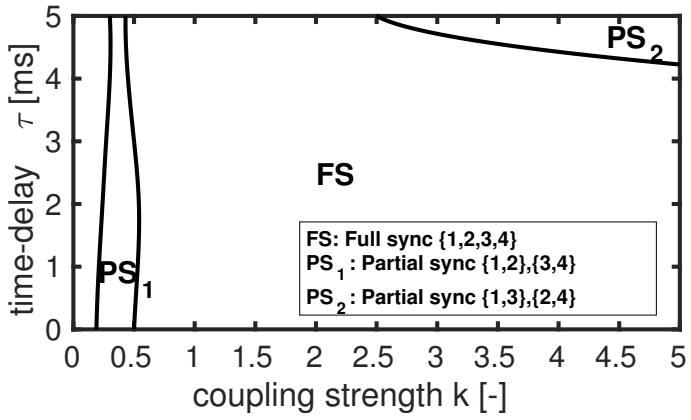
| Synchronization patterns | value of $v_c$ |
|---|---|
| Full synchronization | $[1\ 1\ 1\ 1\ 1\ 1]^\top$ |
| Partial synchronization $PS_1$ | $[1\ 0\ 0\ 0\ 0\ 1]^\top$ |
| Partial synchronization $PS_2$ | $[0\ 1\ 0\ 0\ 1\ 0]^\top$ |
| Partial synchronization $PS_3$ | $[0\ 0\ 1\ 1\ 0\ 0]^\top$ |

Experiments with different values of the coupling strength $k$ and time-delay $\tau$ are performed with this network. Different values of $\varepsilon$ are used for characterizing the synchronization patterns based on the experimental results. The resulting synchronization regions in the parameter space $(k,\tau)$ are depicted in Figure 6.3. Note that there is no region corresponding to partial synchronization $PS_3$ where nodes 1, 4 synchronize and nodes 2, 3 synchronize. When $\varepsilon = 0.1$, no synchronization region corresponding to partial synchronization $PS_2$ is detected because condition (6.5) is too relaxed to differentiate between $PS_2$ and $FS$ with $\varepsilon = 0.1$. In addition, the minimum values of the coupling strength $k$ of the full synchronization region and partial synchronization region $PS_1$ increases when $\varepsilon$ decreases. To give a better impression of the relation between the similarity of the outputs of two neurons and the value of the correlation coefficient, the experimental results of network (6.7) with different values of the parameters $(k,\tau)$ (leading to different correlation coefficients) are shown in Figure 6.4. It can be seen that when the pair of outputs has a correlation coefficient of 0.999, the two outputs are highly similar. Based on Figures 6.3-6.4, $\varepsilon$ is set as 0.001 as the threshold to characterize practical synchronization of the neurons in experiments, which provides a resolution high enough to characterize different synchronization patterns.

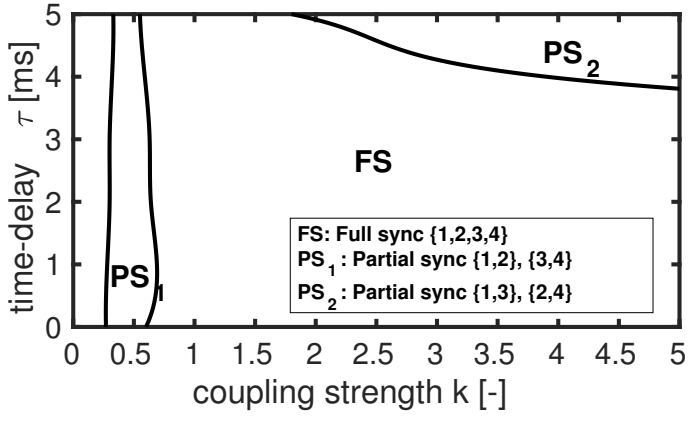
The results obtained by checking the LMI conditions in Theorem 4.4.1 from Chapter 4 are presented in Figure 6.5. Note that there is no region corresponding to partial synchronization $PS_3$ where nodes 1, 4 synchronize and nodes 2, 3 synchronize. By




(a) $\varepsilon = 0.1$

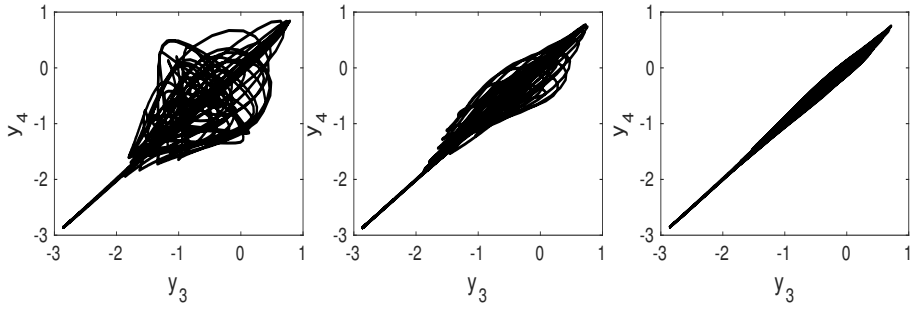



(b) $\varepsilon = 0.01$




(c) $\varepsilon = 0.001$

Figure 6.3: Synchronization regions of the network with the adjacency matrix (6.7) in the $(k, \tau)$ space via experiments, detected by using different values of $\varepsilon$: (a) $\varepsilon = 0.1$, (b) $\varepsilon = 0.01$ and (c) $\varepsilon = 0.001$.




(a) $Co_{y_3,y_4} = 0.9$ (b) $Co_{y_3,y_4} = 0.99$ (c) $Co_{y_3,y_4} = 0.999$

Figure 6.4: Outputs of neurons 3 and 4 in the network with the adjacency matrix (6.7) obtained via experiments with: (a) $k = 0.17$ and $\tau = 2$, leading to $Co_{y_3,y_4} = 0.9$; (b) $k = 0.24$ and $\tau = 2$, leading to $Co_{y_3,y_4} = 0.99$; (c) $k = 0.3$ and $\tau = 2$, leading to $Co_{y_3,y_4} = 0.999$.

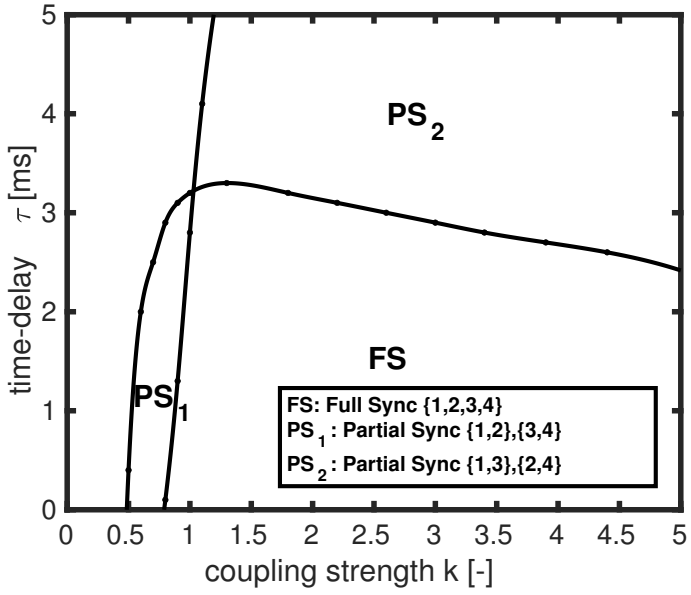



Figure 6.5: Synchronization regions of the network with the adjacency matrix (6.7) in the $(k, \tau)$ space via LMIs.

comparing Figure 6.5 with Figure 6.3(c), it can be seen that there are some mismatches between the synchronization regions obtained by experimental data and those computed by checking the LMI conditions. This can be explained by the conservatism of the theoretical results in Chapter 4. When computing the synchronization regions using the LMI conditions, the regions are computed by checking the feasibility of the LMI conditions for each partial synchronization manifold. Here, the synchronization regions corresponding to $\mathscr{P}_2$, $\mathscr{P}_3$ and $\mathscr{P}_4$ are determined by checking LMI conditions. The full synchronization region is not directly computed as it is the intersection of the partial synchronization regions. Due to the conservatism of the LMI conditions, the regions $PS_1 + FS$ corresponding to $\mathscr{P}_2$ computed by LMI conditions are smaller than those obtained by experiments, and the same goes for the regions $FS + PS_2$ corresponding to $\mathscr{P}_3$ , which causes the mismatches.

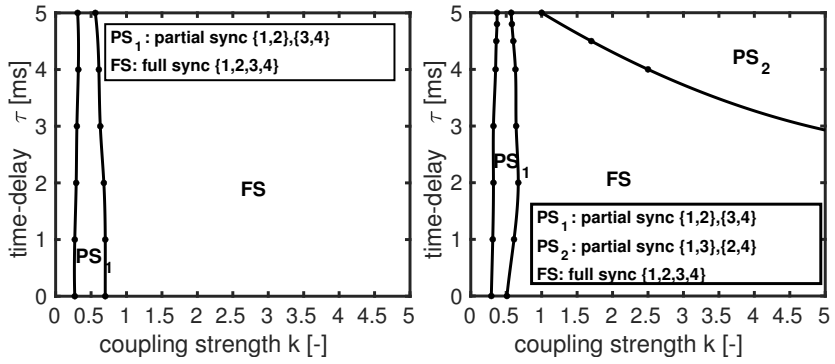
It should be noted that the parameter $E_i$ also impacts the values of coupling strength and time-delay that induce synchronization patterns. The parameter $E_i$ is often used to model the effects of external clues on neurons, for example, the effect of light on neurons in suprachiasmatic nucleus which controls the circadian rhythms of mammals [106]. In particular, it has been shown that the suprachiasmatic nucleus neurons exhibit different fire activities when exposed to different light stimulations [108]. It is possible that during a light-dark cycle, the neurons behave in different modes because of the changing light density. This leads the neurons to exhibit different synchronization patterns such that the circadian rhythm is entrained to the 24-hour light-dark cycle. To demonstrate that, a series of experiments with different values of $E_i$ (representing different light densities) are conducted, and the resulting synchronization regions are shown in Figure 6.6. Recall that the neurons operate in bursting mode when $E_i = 2$, in chaotic bursting mode when $E_i = 3.3$, and in spiking mode when $E_i = 5$. By comparing the synchronization regions in Figures 6.3(c) and 6.6, it can be seen that the synchronization pattern $PS_2$ is more likely to occur when the neurons can produce sufficient numbers of spikes (chaotic bursting and spiking). In addition, the corresponding synchronization region $PS_2$ increases when the behaviors of the neurons become spikier.

### Single partial synchronization manifold

In this experiment, a network consisting of the same type of Hindmarsh-Rose neurons which has only one partial synchronization manifold is considered.

The adjacency matrix is given by

$$A = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 3 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 1 & 0 & 3 & 0 \end{bmatrix}. \tag{6.8}$$




(a) $E_i = 2$ (b) $E_i = 5$

Figure 6.6: Synchronization regions of the network with the adjacency matrix (6.7) in the $(k,\tau)$ space via experiments with (a) $E_i = 2$ and (b) $E_i = 5$, detected by adopting $\varepsilon = 0.001$ as the synchronization threshold.

The values of $E_i$ are set to $E_1 = E_2 = E_3 = E_4 = 3.3$.

In this case, only one viable partition exists, $\mathscr{P}_3 = \{\{1,3\},\{2,4\}\}$. The experimental results are presented in Figure 6.7.

Surprisingly, there are two other synchronization patterns (full synchronization $FS$ and partial synchronization $PS_1$) observed in the experiment. This discrepancy between theory and experiment will be explained by using results from simulations and another series of experiments described in what follows. One possible explanation is the usage of a less stringent criterion for practical synchronization in experiments (compared to exact synchronization or synchronization in simulations). In contrast to experiments, a more strict synchronization criterion is used to check synchronization in simulations where the neurons can be considered to be identical. For simulation data, in addition to the inequality condition (6.5), it is also required that the differences between the outputs of two synchronized neurons should remain within the simulation tolerance. Two simulations are performed with the parameters $(k,\tau)$ being $(0.5,1.6)$ in the $PS_1$ region and $(4,1.6)$ in the $FS$ region. The numerical MATLAB solver *dde23* is used in the simulations with relative and absolute tolerance of $1e^{-5}$ and $1e^{-6}$. Every simulation runs over 10 time units and only the data after 5 time units is used to avoid transient behaviors. The simulation results are shown in Figures 6.8 – 6.9. At the first glance of Figure 6.8, the network seems to be partially synchronized with $k = 0.5, \tau = 1.6$. However, through a scrutiny of the synchronization errors shown in Figure 6.8(b), it can be seen that nodes 1, 2, and nodes 3, 4 are in fact not synchronized. Although the synchronization errors $y_2 - y_1$ and $y_4 - y_3$ are smaller than the other synchronization errors, they are still much larger than the simulation tolerances. According to the

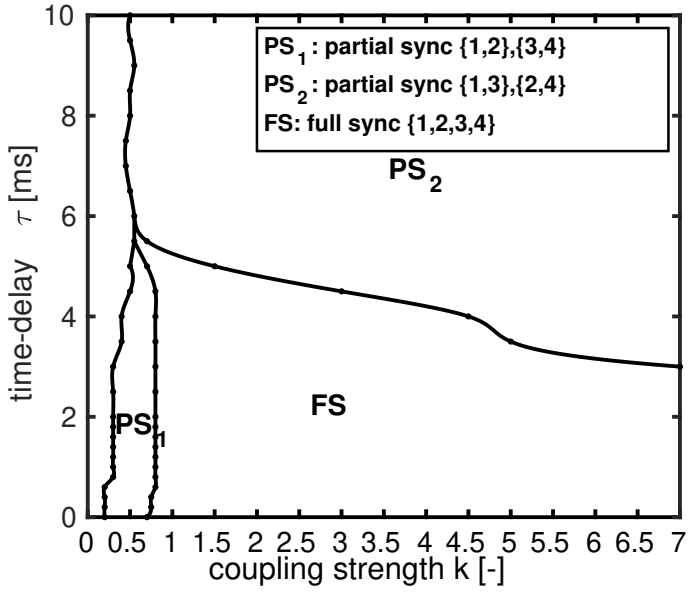


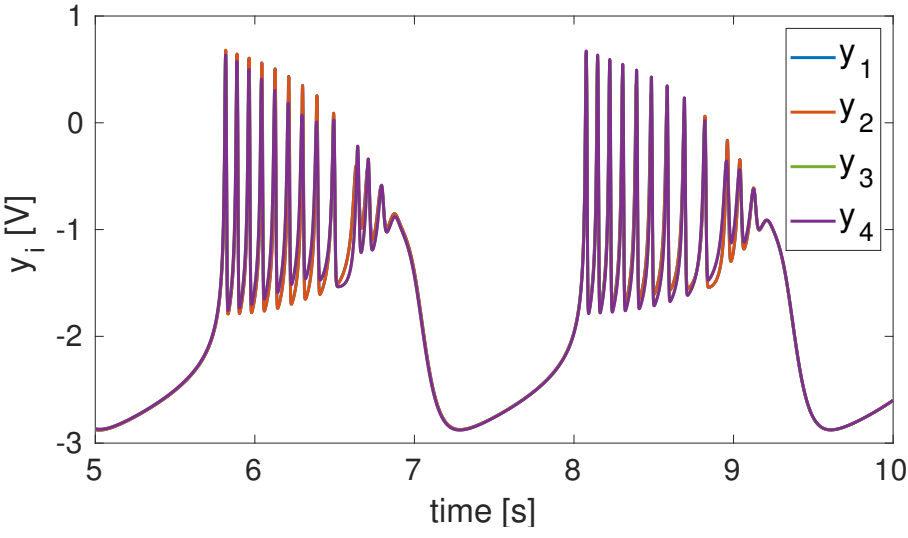

Figure 6.7: Synchronization regions of the network with (6.8) in the $(k,\tau)$ space via experiments, detected by adopting $\varepsilon = 0.001$ as the synchronization threshold.

synchronization errors shown in Figure 6.9(b), it can also be found that the network with $k = 4$ and $\tau = 0.16$ is not fully synchronized, but partially synchronized with nodes 1, 3 in a cluster, and nodes 2, 4 in another cluster. While dealing with experimental results, the less stringent practical synchronization criterion is used. Then by this criterion, similar cases appearing in experiments will be considered as partial synchronization $\{\{1,2\},\{3,4\}\}$ or full synchronization.

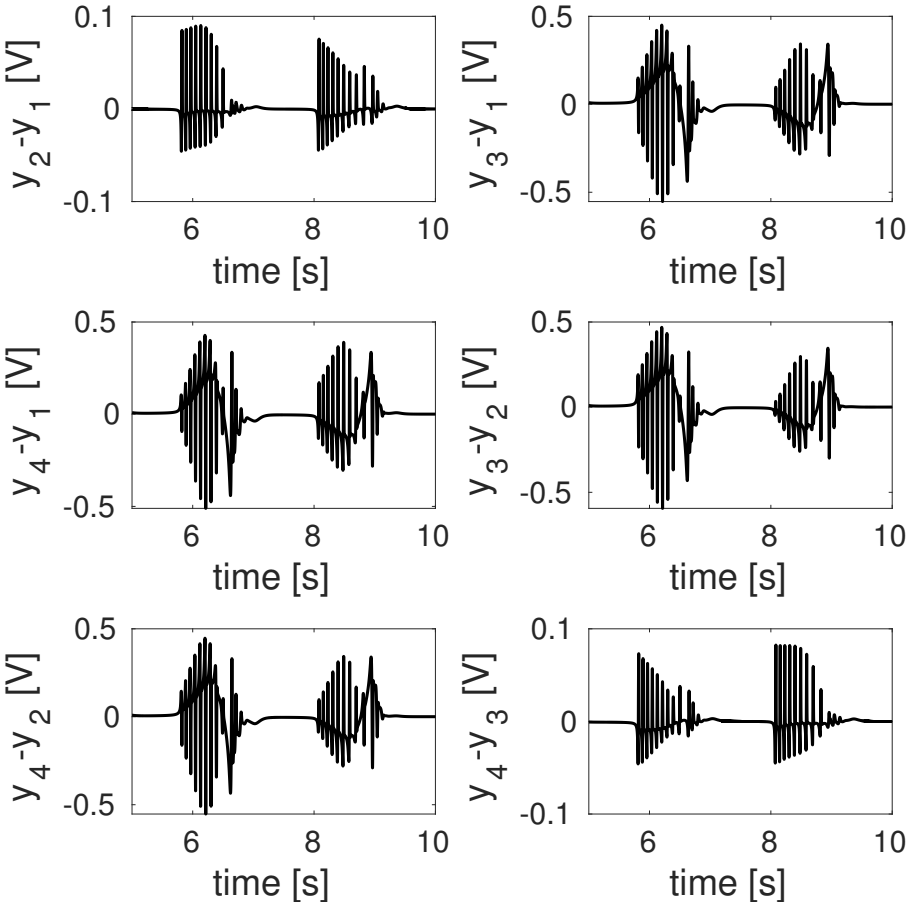
On the other hand, the adjacency matrix (6.8) can be viewed as the adjacency matrix (6.7) with an additive perturbation:

$$A = A_n + \Delta A = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{6.9}$$

Based on the experimental and simulation results, we can infer that the synchronization patterns $FS$ and $PS_1$of the network with the adjacency matrix (6.7) are somewhat robust to perturbations. It is seen that the shapes of regions in Figure 6.3(c) and Figure 6.7 share some similarities. Moreover, if the perturbations increase, the regions corresponding to $FS$ and $PS_1$ should shrink. To verify this inference, a network with

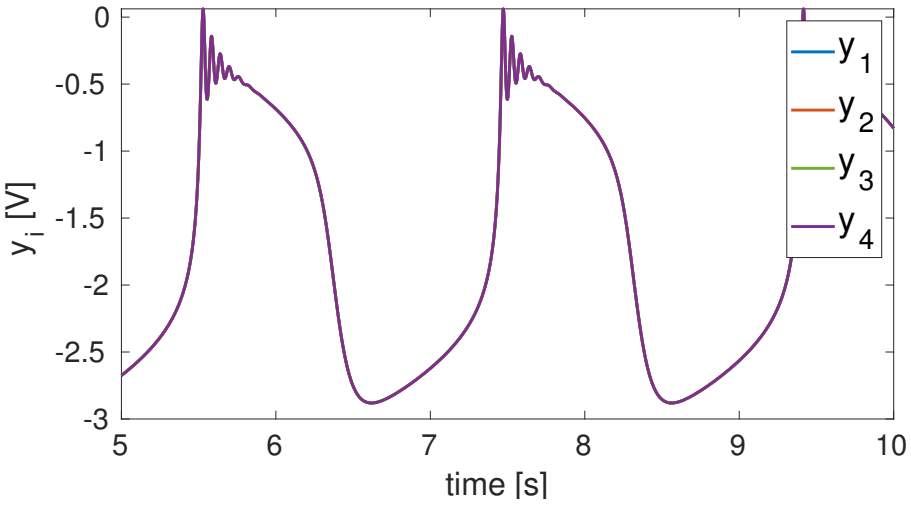



(a) Outputs

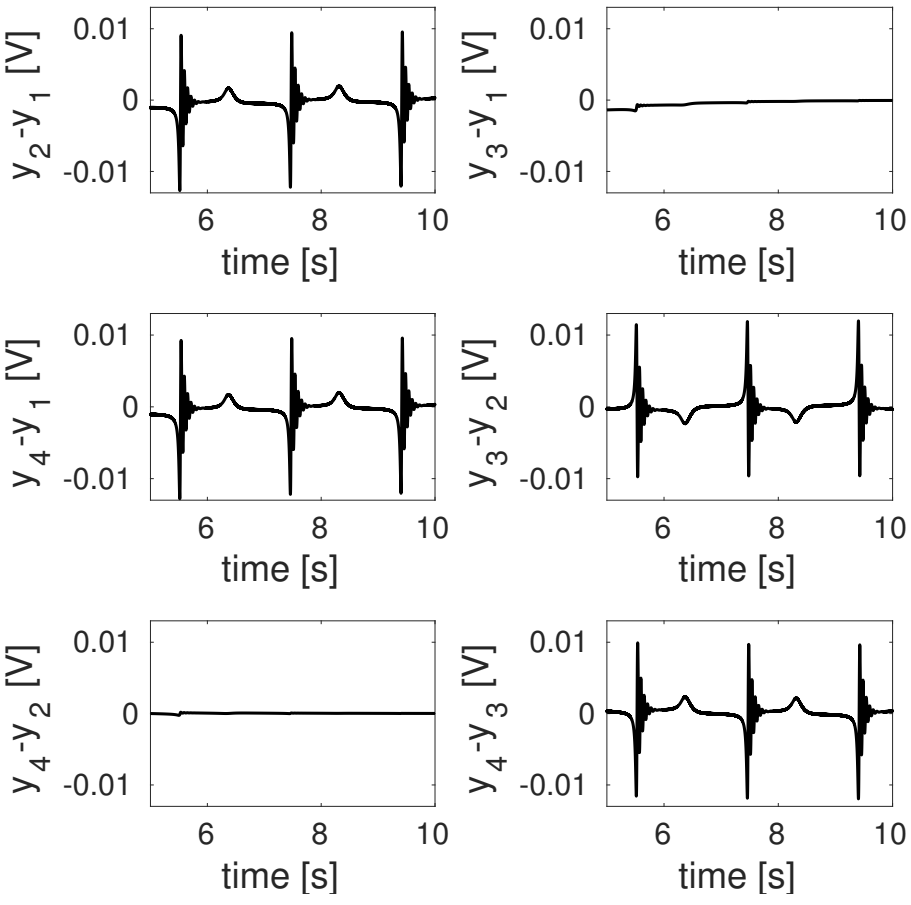



(b) Differences between outputs

Figure 6.8: Behaviors of the neurons in the network with the adjacency matrix (6.8), $k = 0.5$ and $\tau = 0.16$ in simulation.




(a) Outputs




(b) Difference of outputs

Figure 6.9: Behaviors of the neurons in the network with the adjacency matrix (6.8), $k = 4$ and $\tau = 0.16$ in simulation.

the following adjacency matrix is tested

$$A = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 6 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 1 & 0 & 6 & 0 \end{bmatrix}. \tag{6.10}$$

Furthermore, the values of the parameters $E_i$ are set to $E_i = 5$. With $E_i = 5$, the behaviors of the neuron will be spikier, thus enhancing the influences of the perturbation. The synchronization regions of this network obtained via experiments are shown in Figure 6.10. As is seen in the figure, the size of the $PS_2$ region increases while the sizes of the $FS$ and $PS_1$ regions decrease.

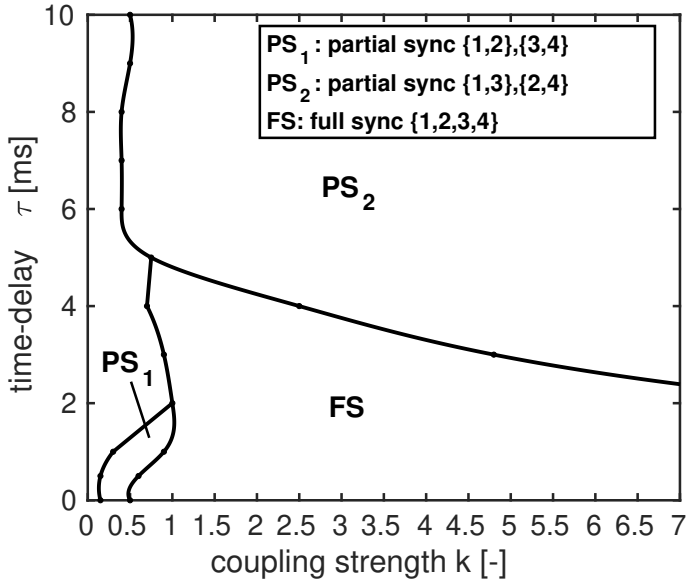



Figure 6.10: Synchronization regions of the network with the adjacency matrix (6.10) in the $(k, \tau)$ space via experiments, detected by adopting $\varepsilon = 0.001$ as the synchronization threshold.

In summary, the results mentioned above indicate that the adopted synchronization measure can, sometimes, fail to accurately characterize synchronization patterns. In this case, a further examination of the differences between the systems' outputs is needed. Comparisons between the sizes of these differences can help to correct the conclusions on synchronization patterns.

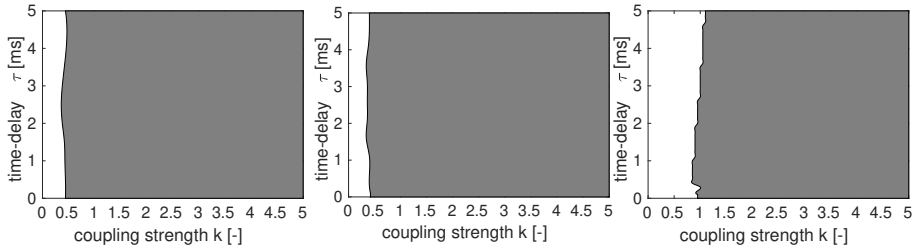
### 6.3.2 Network with different types of neurons

In what follows, a network consisting of different types of neurons is considered. The neurons are of different types when different values of the parameters $E_i$ are adopted. In this experiment, the values of $E_i$ is set as $E_1 = E_3 = 3.3$ and $E_2 = E_4 = 5.5$. The adjacency matrix of the network is given by

$$A = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{bmatrix}, \tag{6.11}$$

and the neurons are interconnected via invasive coupling (6.3).

Using the method presented in Chapter 3, only one viable partition $\mathscr{P} = \{\{1,3\},\{2,4\}\}$ is identified. The experimental results with different values of the coupling strength $k$ and time-delay $\tau$ are presented in Figure 6.11(a), where the grey area is the synchronization region in the $(k,\tau)$ space. The results obtained by a series of simulations are presented in Figure 6.11(b). The results obtained by checking the LMI conditions in Theorem 4.4.1 from Chapter 4 are presented in Figure 6.11(c). Qualitatively, the synchronization regions in these three figures have similar shapes. The region obtained via simulations is a good match of the one obtained via experiments, while the region obtained via LMIs is smaller than the other two, which can be explained by the conservatism of the results in Chapter 4.




(a) Experiments (b) Simulations (c) LMIs

Figure 6.11: Synchronization regions of the network with the adjacency matrix (6.11) in the $(k,\tau)$ space via experiments (a), simulations (b) and LMIs (c)

It has also been observed that when $k$ is increased, the network tends to show full synchronous behavior. For example, when $k = 5$ and $\tau = 0.1$, the network is considered to be fully synchronized by the correlation criterion ($Co \geq 0.999$), see the behaviors of the network in Figure 6.12. This can be explained by the interacting term becoming dominant in the dynamics when increasing $k$, which causes the neurons in different

clusters to behave almost identically. However, by comparing the synchronization errors shown in Figure 6.13, it can be found that the differences between $y_1$ and $y_3$, respectively, $y_2$ and $y_4$ are much smaller than the other cases, indicating that the network is partially synchronized rather than fully synchronized. In particular, except for the pairs of outputs $y_1$ and $y_3$, respectively, $y_2$ and $y_4$, there exist relatively big biases between the other pairs of outputs, which is caused by adopting different values of $E_i$. Note that the adjacency matrix (6.11) is the same as (6.7) in Section 6.3.1. Based on the results in Section 6.3.1, the network is expected to be practically fully synchronized with $k = 5, \tau = 0.1$, when the values of $E_i$ are identical. That is, if the same values of $E_i$ are adopted, the biases between the outputs in Figure 6.13 will vanish.

## 6.4 Conclusions

In this chapter, results of experiments in networks of delay-coupled Hindmarsh-Rose neurons are presented. A cross correlation-based notion of practical synchronization is introduced to quantify the synchronization of the electronic Hindmarsh-Rose neurons. The experiments have been performed with different network settings. The comparisons between the experimental and the theoretical results show the conservatism of the method developed in Chapter 4. However, although the LMI based analysis cannot provide sharp bounds on the coupling parameters for which the systems synchronize, it is able to provide an estimation of these bounds, and predict transitions from one regime to another when parameters are changed. It has been noticed that sometimes, the adopted practical synchronization criterion is not able to accurately identify partial synchronization patterns, as the definition of practical synchronization assumes an explicit separation of the nominal dynamics and perturbations, which is not available. The latter makes the conclusions about the occurrence of patterns sensitive to the thresholds used in the criteria for synchronization. In practice, (small) mismatches between a physical system and its nominal theoretical model are inevitable. To reduce the sensitivity to the synchronization criteria threshold, a more accurate nominal theoretical model can be identified, which is a subject for further research.

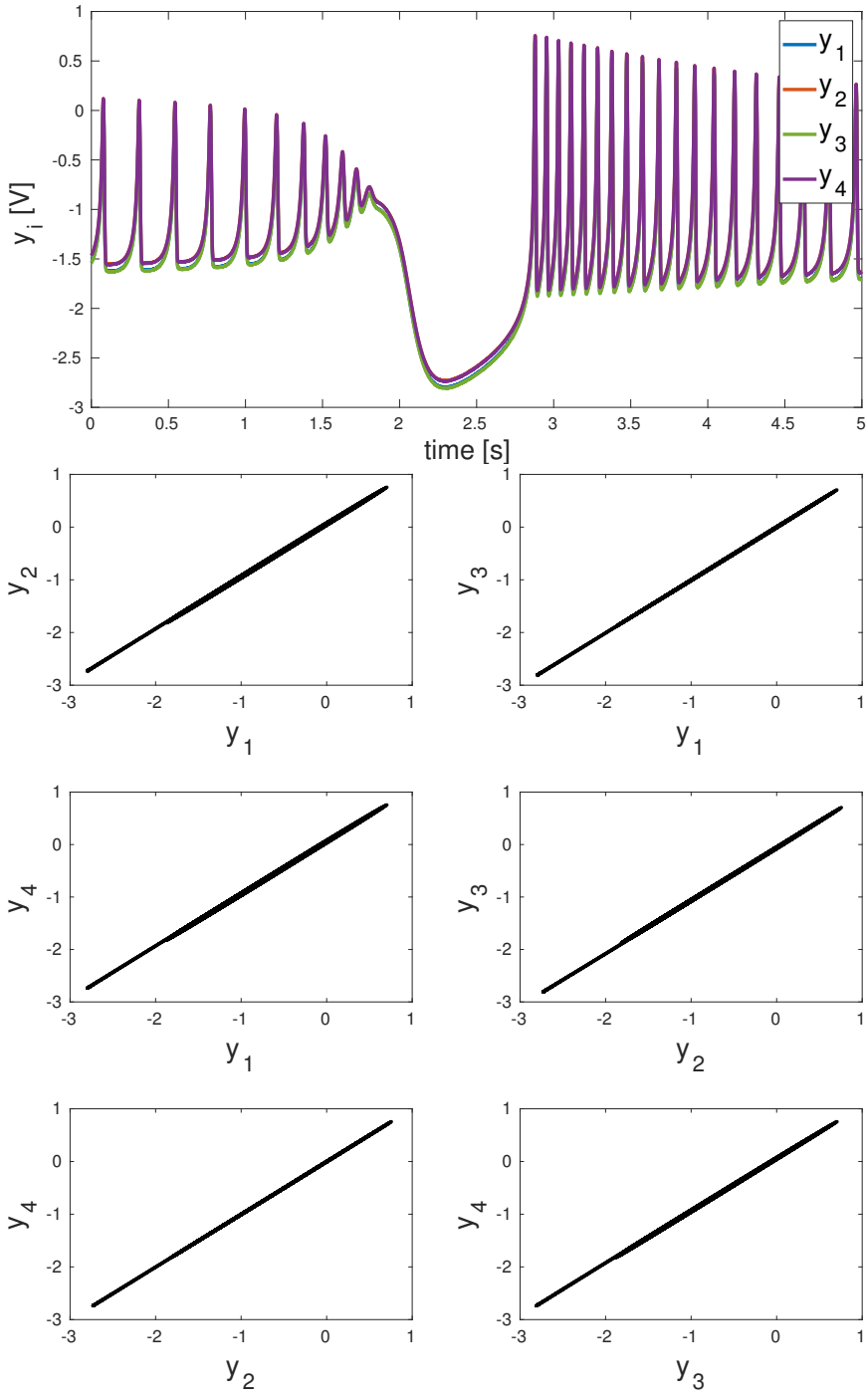



Figure 6.12: Behaviors of the neurons in the network with the adjacency matrix (6.11), $k = 5$ and $\tau = 0.1$ in experiment

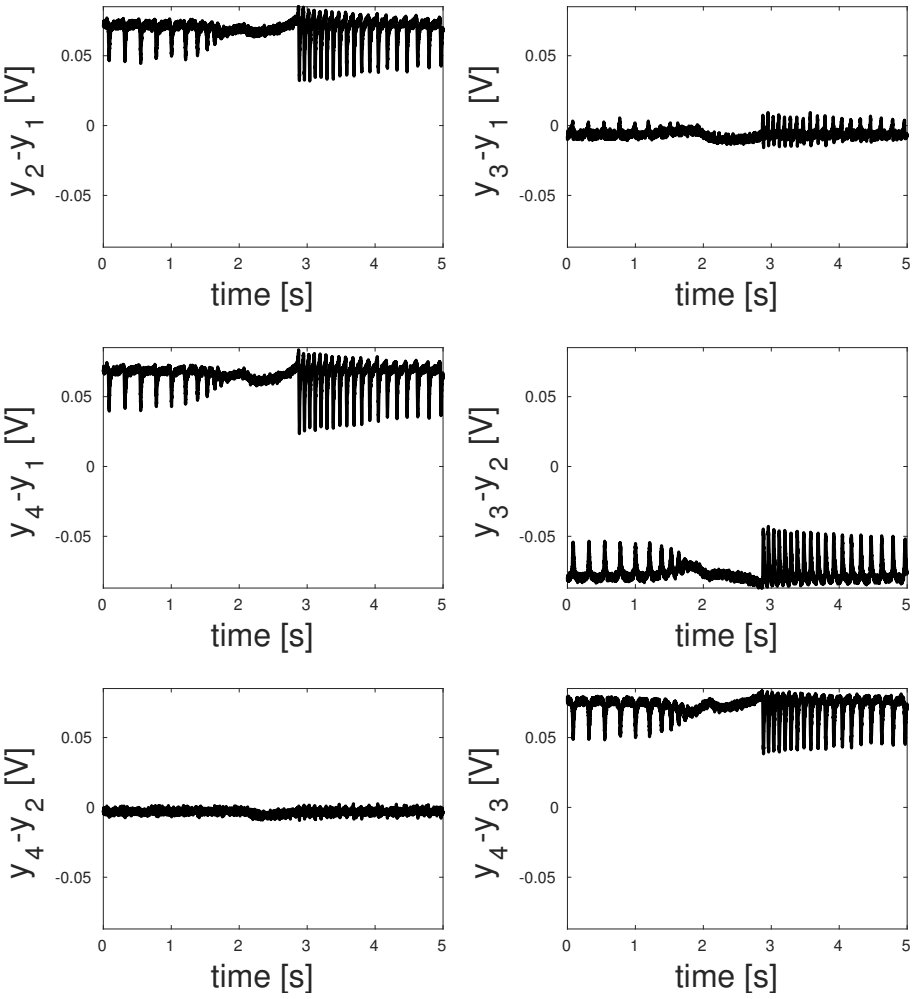



Figure 6.13: Synchronization errors of the network with the adjacency matrix (6.11), $k = 5$ and $\tau = 0.1$.

# Chapter 7

# A model predictive control approach to truck platooning

**Abstract:** Vehicle platooning is an example of controlled network synchronization applications. To improve its performance in terms of the cohesion of a platoon, a Model Predictive Control (MPC) based Cooperative Adaptive Cruise Control (CACC) strategy is proposed in this chapter for vehicle platooning. This approach guarantees that the vehicles in the platoon keep their desired relative positions even when the acceleration limits are reached. A bi-directional topology is employed such that the acceleration capabilities are shared from the rear to the front of a platoon. As a result, the behaviors of the vehicles can be adjusted according to the acceleration limitations of their following vehicles to maintain the cohesion of a heterogeneous platoon. The proposed technique is demonstrated via simulation. The results show that choosing a proper network topology can help to improve the performance of synchronization.

## 7.1 Introduction

In recent years, vehicular platooning, as an application of controlled network synchronization, has received intensive attention, see [109, 110, 8, 4, 111]. Cooperative Adaptive Cruise Control (CACC) is commonly used to realize vehicular platooning. It employs vehicle-to-vehicle (V2V) communication to allow for inter-vehicle data exchange such that very short following distances between vehicles can be reached while guaranteeing safety. This short inter-vehicle distance platooning leads to an increase in road throughput [112, 113], and a decrease in fuel consumption [114, 115].

The latter mainly concerns truck platoons, as trucks experience larger air drag than passenger cars, see [4]. In addition, CACC techniques are often designed to guarantee the string stability of a vehicle platoon (i.e., disturbance attenuation along the vehicle string [8]), which is also necessary for improving throughput and safety.

In many recent CACC designs, one common assumption is the homogeneity of platoon vehicles, see [8, 116, 2, 117]. However, in real life, this assumption does not hold as all vehicles behave differently. In [118], a CACC controller is designed to achieve platooning of heterogeneous vehicles by reformulating the problem in terms of designing decentralized controllers for systems with time-delays and solving it using the method presented in [119]. However, like many other CACC designs in the literature, this CACC controller is designed assuming linear dynamics. In reality, vehicle dynamics are non-linear, especially for heavy-duty trucks. For example, the acceleration capability of a truck is heavily affected by its payload. Hence, trucks in a platoon can have different non-linear and time-varying acceleration capabilities due to different payloads, which can lead to a platoon breakup or other traffic cutting in between in certain scenarios (*e.g.*, dense traffic or hilly terrain) [4]. Consider a platoon of trucks whose acceleration capabilities decrease along the platoon. When the lead vehicle accelerates to a higher velocity, the following vehicles might not be able to follow closely due to lower acceleration limits, resulting in big gaps between vehicles in the transient process. This type of gap creating behavior has also been observed experimentally [120]. In [120], vehicle-following experiments are performed with a CACC equipped heavy-duty truck following a much lighter passenger vehicle. In the experiments, it has been shown that during the catch-up, large spacing errors (the difference between the actual and desired inter-vehicle distances) occur due to the lower acceleration capability of the truck (more frequent gear shifting and lower acceleration limits). Although these gaps vanish eventually (after the platoon reaches steady state again), they should be avoided since they increase the average inter-vehicle distance (less fuel saving) and the possibility of other traffic cutting in between. So, it is desirable to keep the vehicles in a platoon close to each other even in the transient period. This concept of keeping vehicles staying together is referred to as platoon cohesion [4].

However, to the best of our knowledge, few works have considered platoon cohesion explicitly in the CACC or platooning control design. In [4], a bi-directional multi-layer CACC approach is presented to achieve the cohesion of platoons subject to heterogeneous vehicle acceleration capabilities. This approach has a two-layer architecture: 1) a lower layer where a unidirectional CACC as in [8] is employed to achieve low level vehicle control goals (vehicle following); 2) an upper layer where information containing the spacing error and acceleration limitations are shared from the rear to the front of the platoon. In this way, the lead vehicle is aware of the whole platoon's acceleration capability and is able to adapt its behavior accordingly to maintain platoon cohesion. However, some parts of the information communicated

on the upper layer, such as the spacing error, may be considered to be confidential, which hinders the application of this approach. In [121], a bi-directional adaptive cruise control (BiACC) approach to vehicular platoon is proposed with the goals of string stability and platoon coherence (cohesion). Unlike the CACC approach, the BiACC approach does not use wireless V2V communications, instead, it uses an additional distance sensor to measure the inter-vehicle distance to the follower vehicle. With the inter-vehicle distance to the follower vehicle as an additional input, the BiACC controller can also look in the backward direction. In this way, the behaviors of the vehicles can also be adjusted according to their following vehicles to keep the cohesion of the platoon. However, this design requires an additional sensor. Moreover, in this design, to compute the spacing error to the follower vehicle, the spacing policy (the rule determining the desired inter-vehicle distance) of the follower vehicle needs to be known by the preceding vehicle, which can cause a confidentiality issue as the approach in [4].

In this chapter, we propose a Model Predictive Control (MPC)-based CACC strategy for heterogeneous vehicle platooning to achieve platoon cohesion. The vehicle platoon under consideration is heterogeneous in the sense that the vehicles in the platoon possess different maximum acceleration capabilities. MPC is chosen as the basis of the design because it is able to directly accommodate the acceleration limitation. An introduction to MPC can be found in Appendix I. MPC has become a popular choice to design an adaptive cruise control strategy in both research and industrial applications, see [122, 123, 2, 124]. Nowadays, the computing power in the embedded hardware of vehicles is sufficient to run MPC controllers. Inspired by [4], the proposed MPC-based CACC also adopts a bi-directional topology to keep platoon cohesion. Via V2V communication, the intended acceleration of preceding vehicles is sent to the following vehicles, and the acceleration limitation of following vehicles is sent to the preceding vehicles. Note that from the rear to the front of the platoon, only the acceleration capability information is shared among vehicles. In this way, the proposed approach avoids the confidentiality issue of the approach in [4]. Besides, no additional distance sensor to monitor the distance to the follower vehicle is required. Another benefit of sharing acceleration limitations between vehicles is that it allows the vehicles to adjust their behaviors accordingly to maintain the cohesion of a heterogeneous platoon, which is another contribution of this work compared to the MPC-based CACC strategy in [2].

The chapter is organized as follows. Section 7.2 introduces the vehicle model, platoon settings, and platoon cohesion, formulating the problem statement. Section 7.3 describes the MPC-based CACC design. The simulation results of the proposed CACC approach are presented in Section 7.4. Finally, Section 7.5 gives the concluding remarks.

## 7.2 Problem formulation

In this section, the vehicle model, platoon settings, and platoon cohesion are introduced, formulating the main problem addressed in this chapter.

### 7.2.1 Vehicle model

Let us consider a string of $n$ vehicles. Each vehicle is indexed by $i \in \mathcal{V} = \{1, 2, \ldots, n\}$ with $i = 1$ indicating the lead vehicle and $i = n$ the last vehicle. The position, velocity, acceleration at time $t$ of vehicle $i$ is denoted by $p_i(t)$, $v_i(t)$ and $a_i(t)$. The longitudinal dynamics of vehicle $i$ are described by the following non-linear system [4]:

$$\begin{cases} \dot{p}_i(t) = v_i(t), \\ \dot{v}_i(t) = \frac{1}{m_i + m_{eq}} \big( \frac{\eta_T i_d}{R_w} T_i(t) - C_{rl} v_i^2(t) - m_i B_{rl} v_i(t) \\ \qquad\qquad - A_{rl} m_i \cos\alpha - m_i g \sin\alpha \big), \\ \dot{T}_i(t) = -\frac{1}{\tau_i} T_i(t) + \frac{1}{\tau_i} T_{ref,1}(t - \phi_i), \end{cases} \tag{7.1}$$

where $T_i(t)$ is the drive torque, $T_{ref,i}(t)$ is the desired torque, $\tau_i$ is the driveline time constant, $\phi_i$ is the actuation response delay, $m_i$ is the vehicle mass, $A_{rl}$, $B_{rl}$ and $C_{rl}$ denote the road-load parameters associated with road friction, internal friction and aerodynamic drag force, respectively. Furthermore, $\eta_T$ is the transmission efficiency, $R_w$ is the wheel radius, $g$ is the gravitational constant, $\alpha$ is the road slope angle, and $i_d$ is the driveline ratio, defined as

$$i_d = i_{axle} i_g, \tag{7.2}$$

where $i_{axle}$ is the final drive ratio and $i_g$ is the current gear ratio. Here, $i_g := i_g(v_i)$ is defined as a piecewise-constant function of velocity to model the gear shifting effect. Consequently, $i_d$ is also a piecewise-constant function of velocity. That is, it takes different constant values for different velocity ranges. Finally, $m_{eq}$ is the equivalent mass of all the rotating parts of the vehicle which is defined as

$$m_{eq} = \frac{i_d^2 J_e + J_w}{R_w^2}, \tag{7.3}$$

with $J_e$ being the mass moment of inertia of the engine and $J_w$ being the mass moment of inertia of all wheels. Since the driveline ratio $i_d$ appears in (7.3), the equivalent mass $m_{eq}(v_i)$ is also a piecewise-constant function of velocity. In general, all these vehicle parameters can be different among the platoon members. In this chapter, we consider the vehicle mass as a heterogeneous parameter since it can vary significantly for different trucks, the other parameters are considered to be homogeneous, including $\phi_i$ and $\tau_i$, that is, $\phi_i = \phi$ and $\tau_i = \tau$. As shown in [4], using the following control law

for the desired torque $T_{ref,i}$:

$$\begin{aligned} T_{ref,i} = \frac{R_w}{\eta_T i_d}\big(&C_{rl}(v_i(2\tau_i\dot{v}_i + v_i) + m_i B_{rl}(\tau_i\dot{v}_i + v_i) \\ &+ m_i A_{rl}\cos\alpha + m_i g\sin\alpha \\ &+ (m_i + m_{eq})u_i\big), \end{aligned} \tag{7.4}$$

with $u_i$ being the desired acceleration (the input to the driveline), the non-linear longitudinal dynamics (7.1) can be feedback linearized. By substituting (7.4) and $a_i = \dot{v}_i$ into (7.1), the longitudinal vehicle dynamics (7.1) can be rewritten as

$$\begin{cases} \dot{p}_i(t) = v_i(t), \\ \dot{v}_i(t) = a_i(t), \\ \dot{a}_i(t) = -\frac{1}{\tau}a_i(t) + \frac{1}{\tau}u_i(t-\phi). \end{cases} \tag{7.5}$$

In addition, the input $u_i$ is constrained with a lower and upper bound,

$$a_{min,i} \le u_i \le a_{max,i}, \tag{7.6}$$

where $a_{min,i} < 0$ represents the minimum acceleration (deceleration) limit, and $a_{max,i}$ represents the maximum acceleration limit due to the limited engine torque. Here, the deceleration limit $a_{min,i}$ is modeled by a constant value, while the maximum acceleration limit $a_{max,i}$ is a time-varying value determined by the maximum engine torque, current gear ratio, velocity, etc. More precisely, by setting $T_i = T_{max,i}$ in the second equation of (7.1), the maximum acceleration limit $a_{max,i}$ is given by

$$\begin{aligned} a_{max,i}(v_i) = \frac{1}{m_i + m_{eq}}\Big(&\frac{\eta_T i_d}{R_w}T_{max,i} - C_{rl}v_i^2 - m_i B_{rl}v_i \\ &- m_i A_{rl}\cos\alpha - m_i g\sin\alpha\Big), \end{aligned} \tag{7.7}$$

where $T_{max,i}$ is the upper bound of the engine torque. In addition, it is assumed the engine and gearbox are designed such that the maximum engine torque is always available. Note that, as (7.7) indicates, $a_{max,i}$ is state-dependent, while, in many MPC-based CACC designs (*e.g.*, [109], [111], [2]), a fixed value of $a_{max,i}$ is often used.

### 7.2.2 Platoon settings

A schematic diagram of a vehicular platoon is shown in Figure 7.1. The inter-vehicle distance $d_i$ is defined as the distance between the front bumper of vehicle $i$ and the rear bumper of vehicle $i-1$, given by

$$d_i(t) = p_{i-1}(t) - p_i(t) - L_i, \quad i \in \mathscr{V}\backslash\{1\}, \tag{7.8}$$

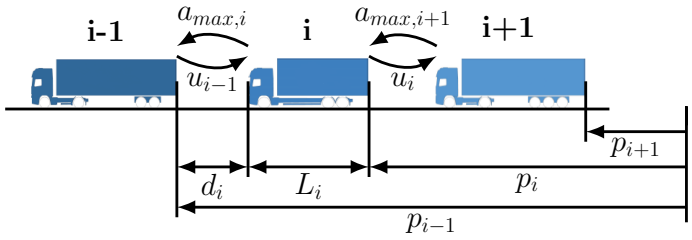



Figure 7.1: Schematic overview of a heterogeneous vehicle platoon and the bi-directional information flow topology

where $L_i$ is the length of vehicle $i$.

For each vehicle in the platoon, the objective is to longitudinally follow the preceding vehicle at the desired distance defined as below

$$d_{r,i}(t) = r_i + h_i v_i(t), \tag{7.9}$$

where $r_i$ is the desired standstill distance and $h_i$ is the time gap, respectively. This spacing policy is chosen because of its favorable properties for disturbance attenuation, see [8].

Combining (7.8) and (7.9), the spacing error $e_i(t)$ is given by

$$e_i(t) = d_i(t) - d_{r,i}(t) = p_{i-1}(t) - p_i(t) - L_i - r_i - h_i v_i(t). \tag{7.10}$$

Obviously, one essential goal of vehicle platoon control design is to asymptotically realize the desired spacing distance, i.e., zero spacing error

$$\lim_{t\to\infty} e_i(t) = 0, \forall i \in \mathscr{V}\backslash\{1\}. \tag{7.11}$$

Another important goal of vehicle platoon control design is to achieve the platoon's string stability, the attenuation of disturbances in the upstream direction of vehicles. Several definitions of string stability are available in the literature. Here, we briefly introduce the $\mathscr{L}_p (p = 2, \infty)$ induced norm-based string stability notion from [8]. Let the signal of interest for vehicle $i$ be denoted by $z_i(t)$ (*e.g.*, $a_i(t)$, $v_i(t)$ and $u_i(t)$), the string stability requirement can be expressed as

$$\|z_i(t)\|_{\mathscr{L}_p} \leq \|z_{i-1}(t)\|_{\mathscr{L}_p}, \forall i \in \mathscr{V}\backslash\{1\}. \tag{7.12}$$

### 7.2.3 Platoon cohesion

As mentioned before, in many CACC designs, vehicles are assumed to be homogeneous. Moreover, some designs also assume that the vehicle has unlimited acceleration capability. Using such controllers in practice where the vehicles have heterogeneous acceleration limitations, may cause undesired performance, as illustrated in the example in Figure 7.2.

Figure 7.2 sketches the scenario where a platoon of heterogeneous vehicles breaks up during acceleration using the controllers which do not take care of the different acceleration capabilities. Here, a platoon of four trucks with different masses is considered. The 4th truck has a larger mass than the other three trucks. Initially, all vehicles drive at 60 km/h, forming a platoon. Then, the desired speed of the lead vehicle is increased to 80 km/h. With a larger mass, the 4th vehicle has lower acceleration capability, which makes it impossible to keep up with the other vehicles when the acceleration limit is reached. Thus, a big gap between Truck 3 and Truck 4 is created, breaking up the platoon. This big gap might leave space for other traffic to cut in. Moreover, this leads to less fuel saving due to a larger average following distance. Therefore, it is important to consider the cohesion of a platoon when designing a control strategy. Loosely speaking, platoon cohesion means that the spacing errors $e_i(t)$ should remain small (also in the transient phase) when the platoon is disturbed from steady state. The requirement of platoon cohesion can be presented by

$$\|e_i(\cdot)\|_\infty \leq c, \tag{7.13}$$

where $c > 0$ is a parameter of choice. Here, $\|e_i(\cdot)\|_\infty$ denotes the $\infty$-norm of $e_i(t)$:

$$\|e_i(\cdot)\|_\infty = \sup_t |e_i(t)|. \tag{7.14}$$

## 7.3 Control design

In this section, we propose an MPC-based CACC strategy for the control problem of the vehicle platoon presented in the previous section. The control strategy is designed to achieve two goals: 1) vehicle following, i.e. $\lim_{t\to\infty} e_i(t) = 0, \forall i \in \mathscr{V}\backslash\{1\}$; 2) platoon cohesion, i.e. $\|e_i(t)\|_\infty \leq c, c > 0, \forall i \in \mathscr{V}\backslash\{1\}, \forall t > t_1$ ($t_1$ is the time instance when the vehicles form a platoon). Another important goal of CACC design, string stability, is not directly enforced here. Instead, the string stability property of the resulting platoon dynamics is assessed by means of simulation results for different MPC controller parameters. The design of the proposed strategy is inspired by [4] for adopting a bi-directional topology and by [2] for the MPC design.

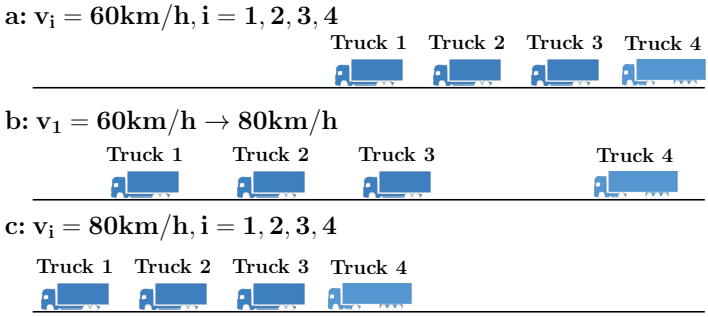



Figure 7.2: Platoon behavior during acceleration from 60 km/h to 80 km/h. Phase a: all vehicles drive at 60 km/h. Phase b: the lead vehicles accelerate to 80 km/h. In this transient phase, the 4th truck is not able to follow the preceding vehicle due to lower acceleration capability. A big gap between the 3rd truck and 4th truck is created, breaking the platoon. Phase c: The platoon reaches steady state again (the platoon is assumed to be stable). All vehicles drive at 80 km/h. The gap vanishes.

### 7.3.1 Control law design

A block diagram of the CACC controller is shown in Figure 7.3. The CACC consists of two parts: 1) a feedforward action based on the wirelessly communicated intended acceleration of a preceding vehicle; 2) a feedback action calculated by the MPC controller. Here, we adopt the control law from [2], given by

$$\dot{u}_i(t) = -\frac{1}{h_i}u_i(t) + \frac{1}{h_i}q_i(t), \tag{7.15}$$

with

$$q_i(t) = u_{i-1}(t-\theta) + u_i^{MPC}(t), \tag{7.16}$$

where $\theta$ is the communication delay, $u_{i-1}(t-\theta)$ is the (delayed) feedforward part, and $u_i^{MPC}$ is the MPC-based feedback part. Note that, by employing a bi-directional topology, $a_{max,i+1}$ is shared to vehicle $i$ and fed into the MPC controller to formulate constraints. The detailed usage of this information is explained in the following part of this section.

### 7.3.2 MPC design

First, the closed-loop system is derived for the design of the MPC controller. Let the state vector $x_i(t)$ be

$$x_i(t) = \begin{bmatrix} e_i(t) & \dot{e}_i(t) & \ddot{e}_i(t) & u_i(t-\phi_i) \end{bmatrix}^\top. \tag{7.17}$$

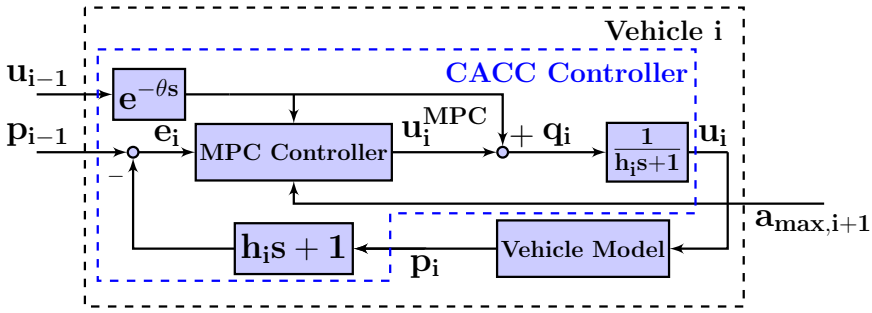



Figure 7.3: Block diagram of the CACC Controller

According to the definition of the spacing error, the third derivative of $e_i$ is given by

$$\dddot{e}_i(t) = \dot{a}_{i-1}(t) - \dot{a}_i(t) - h_i \ddot{a}_i(t) \tag{7.18}$$

As mentioned before, $u_{i-1}$ is wirelessly communicated to vehicle $i$, therefore the communication delay $\theta$ should be included in the model of vehicle $i-1$ when seen from vehicle $i$:

$$\dot{a}_{i-1}(t) = -\frac{1}{\tau} a_{i-1}(t) + \frac{1}{\tau} u_{i-1}(t - \phi - \theta). \tag{7.19}$$

Then the third derivative of $e_i$ can be expressed as:

$$\dddot{e}_i(t) = -\frac{1}{\tau} \ddot{e}_i(t) - \frac{1}{\tau}(u_i(t-\phi) + h_i \dot{u}_i(t-\phi)) + \frac{1}{\tau} u_{i-1}(t - \phi - \theta). \tag{7.20}$$

Shifting time over $\phi$ in (7.15) and (7.16), and then combining them, yields

$$\dot{u}_i(t-\phi) = \frac{1}{h_i}(u_{i-1}(t - \phi - \theta) + u_i^{MPC}(t-\phi)) - \frac{1}{h_i} u_i(t-\phi). \tag{7.21}$$

Substituting (7.21) into (7.20), we have

$$\dddot{e}_i(t) = -\frac{1}{\tau} \ddot{e}_i(t) - \frac{1}{\tau} u_i^{MPC}(t-\phi). \tag{7.22}$$

Now, the closed-loop system is given by

$$\begin{aligned} \dot{x}_i(t) &= A x_i(t) + B_1 u_i^{MPC}(t - \phi_i) + B_2 u_{i-1}(t - \theta_i - \phi_i), \\ y_i(t) &= C x_i(t), \end{aligned} \tag{7.23}$$

with

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\tau} & 0 \\ 0 & 0 & 0 & -\frac{1}{h_i} \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{\tau} \\ \frac{1}{h_i} \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{h_i} \end{bmatrix}, \tag{7.24}$$

and the outputs $y = [e_i(t)\ \dot{e}_i(t)\ u_i(t-\phi_i)]^\top$ with

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7.25}$$

Second, to implement the MPC controller, the discretization of the above system is needed. Let us denote the sampling time by $t_s$. The actuation delay and communication delay are approximated by the discrete time delays $\phi_i^d$ and $\theta_i^d$, respectively, defined as the closest integer multiple of the sampling time $\phi_i^d = \lfloor \frac{\phi_i}{t_s} \rceil \in \mathbb{N}_0$ and $\theta_i^d = \lfloor \frac{\theta_i}{t_s} \rceil \in \mathbb{N}_0$. The corresponding discrete model using zero-order-hold method (see [125]) is shown below

$$\begin{aligned} x_i(k+1) &= A_d x_i(k) + B_{d,1} u_i^{MPC}(k-\phi_i^d) \\ &\quad + B_{d,2} u_{i-1}(k-\theta_i^d-\phi_i^d), \\ y_i(k) &= C_d x_i(k), \end{aligned} \tag{7.26}$$

where

$$A_d = \begin{bmatrix} 1 & t_s & \tau^2 e^{-t_s/\tau} + t_s\tau - \tau^2 & 0 \\ 0 & 1 & \tau - \tau e^{-t_s/\tau} & 0 \\ 0 & 0 & e^{-t_s/\tau} & 0 \\ 0 & 0 & 0 & e^{-t_s/h_i} \end{bmatrix}, \tag{7.27}$$

$$B_{d,1} = \begin{bmatrix} t_s\tau + \tau^2 e^{-t_s/\tau} - \tau^2 - t_s^2/2 \\ -t_s - \tau_i e^{-t_s/\tau} + \tau \\ e^{-t_s/\tau} - 1 \\ 1 - e^{-t_s/h_i} \end{bmatrix}, \tag{7.28}$$

$$B_{d,2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 - e^{-t_s/h_i} \end{bmatrix}, C_d = C. \tag{7.29}$$

Third, two constraints are imposed for desired functionality. The first one concerns the distance. The distance should be non-negative:

$$d_i = e_i + d_{r,i} \geq 0. \tag{7.30}$$

The second one relates to the limitation of the acceleration:

$$a_{min,i} \leq u_i \leq \tilde{a}_{max,i}. \tag{7.31}$$

where, $a_{min,i}$ is prescribed, and $\tilde{a}_{max,i}$ is a virtual acceleration limit determined by comparing $a_{max,i}$ with $\tilde{a}_{max,i+1}$ (obtained via V2V communication):

$$\tilde{a}_{max,i} = \begin{cases} \min\{a_{max,i}, \tilde{a}_{max,i+1}\}, & i \in \mathscr{V}\backslash\{n\}, \\ a_{max,i}, & i = n. \end{cases} \tag{7.32}$$

With (7.32), it is guaranteed that

$$\tilde{a}_{max,i} \leq \tilde{a}_{max,i+1}, \tag{7.33}$$

that is, for each vehicle, it always has an equal or stronger acceleration ability compared to its preceding vehicle, which enables the vehicles to stay close during acceleration, achieving platoon cohesion. However, the platoon cohesion condition (7.13) is not formulated as a constraint for the MPC controller. If (7.13) is used as a constraint directly, the MPC problem will become infeasible for certain cases, for instance, for platooning vehicles with large (greater than $c$) initial spacing errors. The condition is only required after a platoon is formed (for example, for the transient periods after the platoon is perturbed from steady state). In addition, to meet the condition (7.13) with a given $c$, apart from using (7.32), the MPC design parameters also need to be well-tuned. It is also worth mentioning that like $a_{max,i}(v_i)$, $\tilde{a}_{max,i}$ is also state-dependent, thus making (7.31) a state-dependent constraint. Note that for the vehicle dynamics model, the input saturation condition (7.6) is also replaced by (7.31).

Finally, we can formulate the MPC control problem as the optimization problem as below.

$$\begin{aligned} \min_{u_i^{MPC}(\cdot|k)} \sum_{j=1}^{N_p} \Big( & y_i(k+j|k)^\top Q y_i(k+j|k) \\ & + u_i^{MPC}(k+j|k)^\top R u_i^{MPC}(k+j|k) \\ & + \Delta u_i^{MPC}(k+j|k)^\top R_\Delta \Delta u_i^{MPC}(k+j|k) \Big), \end{aligned} \tag{7.34}$$

subject to the prediction model (7.26) and the constraints

$$E_i u_i^{MPC}(k+j|k) + F_i y_i(k+j|k) \leq G_i, \tag{7.35}$$

with

$$E_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, F_i = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 1r_i \end{bmatrix}, G_i = \begin{bmatrix} r_i + r_i v(k) \\ -a_{min,i} \\ \tilde{a}_{max,i}(k) \end{bmatrix}, \tag{7.36}$$

where $\Delta u_i^{MPC}(\cdot)$ is the changing rate of the MPC control action, defined by

$$\Delta u_i^{MPC}(k) = u_i^{MPC}(k) - u_i^{MPC}(k-1),$$

$N_p$ is the prediction horizon, $Q = \text{diag}(q_1, q_2, 0)$ with $q_1 > 0$, $q_2 > 0$, and $R > 0$, $R_\Delta > 0$ are the weighing matrices.

The weighing matrix $R$ is related to fuel consumption. $R_\Delta$ is introduced for driving comfort, see [2], [126]. The inequalities (7.35) incorporate the two constraints (7.30) and (7.31), the latter allows vehicles to catch up with their preceding vehicles more easily. All these design parameters directly affect the performance of the MPC controller. To satisfy the requirements on vehicle following, platoon cohesion and string stability, a set of well-tuned parameters are required. To provide guidance on parameter tuning, the impacts of these design parameters are investigated by means of simulations with different MPC design parameters.

## 7.4 Simulation results

In this section, the effectiveness of the proposed MPC-based CACC is demonstrated. Several simulations are performed using MATLAB and Simulink. The proposed MPC-based CACC is applied to the vehicle platoon in the example of Section 7.2.3 and the same scenario is repeated. Besides, we have performed a parameter study to investigate the impact of the MPC design parameters (the weighting matrices) and thereby to provide guidance for choosing parameters.

First, the values of the vehicle and platoon parameters are specified. The masses of the vehicles are listed below:

$$(m_1, m_2, m_3, m_4) = (25, 25, 25, 40) \cdot 10^3\,[kg]. \tag{7.37}$$

Consequently, the levels of the maximum acceleration $a_{max,i}$ are different for the trucks. The other values of the trucks and platoon parameters are listed in Table 7.1 and the gear shifting rule is shown in Table 7.2. As in [4], the gear shifting rule is simply modeled by changing the gear ratio $i_g$ according to the actual speed of the truck. More specifically, when the speed reaches another speed range, the gear ratio changes to the new gear ratio in 1.5 seconds at a constant rate. In reality, there is also a dip in the acceleration due to the clutch release when a truck changes gears. However, this effect is not included here since it is not the focus of this chapter. We refer to the experimental results in [120] for more realistic behaviors of trucks with gear shifting. For the first 10 seconds, the platoon is moving at a speed of 60 km/h with zero spacing error. After $t = 10s$, the desired speed of the lead vehicle is set to $v_{des}$ =80 km/h. The lead vehicle is equipped with the following cruise controller

$$u_1(t) = k_v(v_{des} - v_1), \tag{7.38}$$

with $k_v > 0$.

Table 7.1: Values of vehicle and platoon parameters

| Parameter | Value | Unit | Parameter | Value | Unit |
|---|---|---|---|---|---|
| $R_w$ | 0.45 | $m$ | $C_{rl}$ | 1.25 | $kg/m$ |
| $i_{axle}$ | 2.5 | - | $\alpha$ | 0 | $rad$ |
| $J_w$ | 232 | $kgm^2$ | $g$ | 9.81 | $m/s^2$ |
| $J_e$ | 2.5 | $kgm^2$ | $\tau$ | 0.1 | $s$ |
| $T_{max,i}, \forall i$ | 2500 | $Nm$ | $\phi$ | 0.2 | $s$ |
| $L_i, \forall i$ | 17 | $m$ | $\theta$ | 0.15 | $s$ |
| $\eta_T$ | 1 | - | $h_i, \forall i$ | 0.7 | $s$ |
| $A_{rl}$ | 0.039 | $m$ | $r_i, \forall i$ | 2.5 | $m$ |
| $B_{rl}$ | 0.0037 | $m/s^2$ | $a_{min,i}, \forall i$ | $-6$ | $m/s^2$ |

Table 7.2: Gear ratios for different speed ranges [4]

| Speed range$[km/h]$ | Gear ratio $i_g[-]$ |
|---|---|
| $0-10$ | 9.6 |
| $10-20$ | 5.9 |
| $20-30$ | 3.5 |
| $30-45$ | 2.1 |
| $45-70$ | 1.2 |
| $>70$ | 1 |

Second, we apply the CACC strategy proposed in this chapter using the same parameters as in [2] (listed in Table 7.3) and the simulation results are depicted in Figure 7.4. For comparative purposes, the CACC proposed in [2] is also applied to this example with the same parameters and the simulation results are depicted in Figure 7.5. In this CACC design, only the local acceleration limitations are considered, that is,

$$a_{min,i} \leq u_i \leq a_{max,i}. \tag{7.39}$$

From Figure 7.4 and Figure 7.5, we can see that the spacing errors are greatly reduced when using the CACC proposed in this chapter, compared to the one from [2]. The vehicle platoon maintains platoon cohesion (with $c = 0.86$) using the MPC-based CACC technique presented in this chapter. However, in Figure 7.4, between $t = 35s$ and $t = 50s$, overshoots in the acceleration and speed response are observed and the platoon shows string unstable behavior (for example, the spacing errors are amplified along the platoon). To explain this phenomenon, let us examine the platoon response from the beginning. The leader starts to accelerate at $t = 10s$. During the period

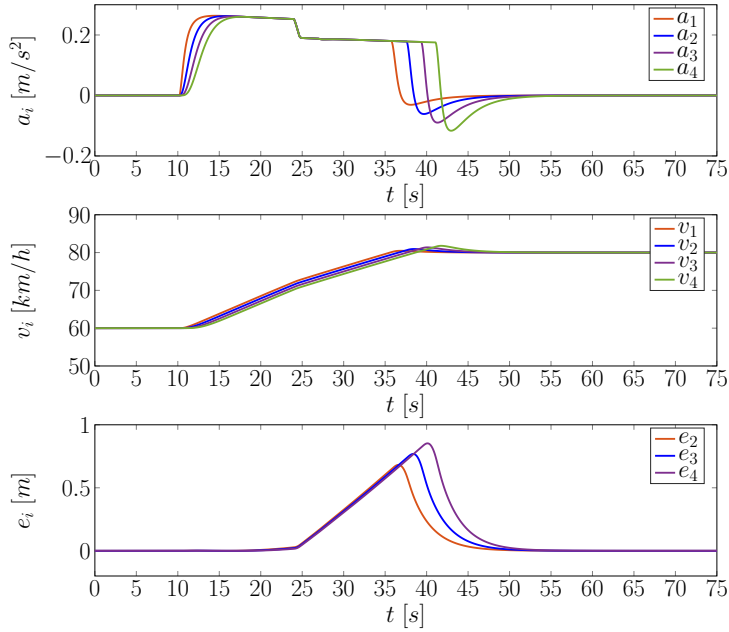



Figure 7.4: Acceleration $a_i$, speed $v_i$, and spacing error $e_i$ response of a four-truck platoon during acceleration from 60 km/h to 80 km/h using MPC-based CACC.

from $t = 15s$ to $t = 35s$, the accelerations of all the vehicles are constrained at the same levels. But the vehicles start to accelerate at different times, hence there are speed differences between the vehicles, which leads to a building up of the spacing error during this period. At $t = 35s$, the speed of the lead vehicle comes close to the desired speed and it begins to stop accelerating, while the other vehicles continue to accelerate to close the gaps. But with the current set of the MPC parameters, the CACC controller behaves like a high proportional gain Proportional Integral Derivative (PID) controller, thus making the following vehicles accelerate excessively (overshoot). Due to the excessive acceleration, the vehicles need to decelerate afterward. To reduce this undesired behavior, we have performed a simulation-based study on the MPC parameters $(Q, R, R_\Delta)$. From the simulation results we have observed:

- $q_1$ behaves similarly as a proportional gain in a PID controller. Increasing the value of $q_1$ leads to a decrease in the settling time and an increase in the overshoot.
- $q_2$ behaves similarly as a derivative gain in a PID controller. Increasing the value of $q_2$ leads to an increase in the settling time and a decrease in the overshoot.

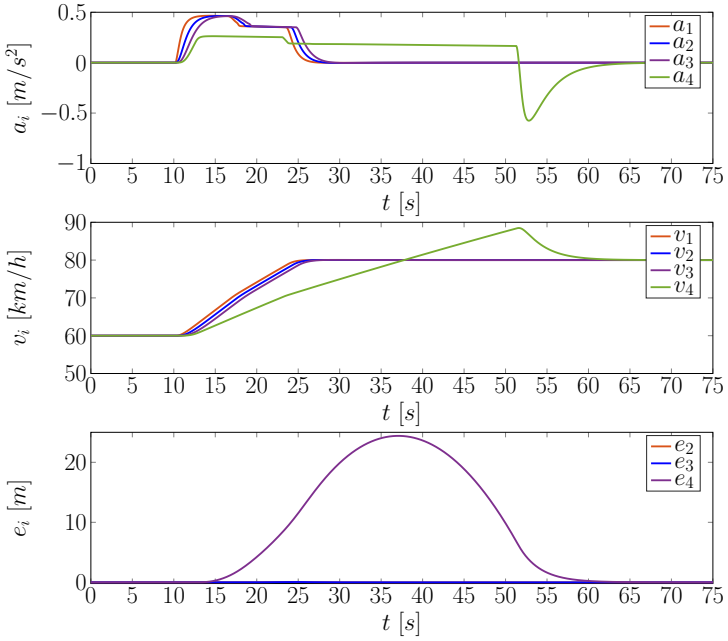



Figure 7.5: Acceleration $a_i$, speed $v_i$ and spacing error $e_i$ response of a four-truck platoon during acceleration from 60 km/h to 80 km/h using MPC-based CACC from [2].

- $R$ has a similar impact as $q_2$.
- $R_\Delta$ introduces additional oscillations to the system response. A high value of $R_\Delta$ brings oscillations with large magnitude.

Based on these observations, we have chosen another set of parameters (listed in Table 7.4) such that the undesired behaviors are avoided. Here, $q_2$ and $R$ are increased to minimize the overshoots, while $q_1$, as well as $N_p$, is increased to ensure the settling time is still acceptable. The simulation results are shown in Figure 7.6. Additionally, the cohesion of the platoon improves in terms of the upper bound of the spacing errors (now $c = 0.74$). The price paid here is a slower response. The spacing error reduces to zero after 120 seconds instead of 60 seconds.

Another way to avoid the undesired behavior is to adopt the following definition of $\tilde{a}_{max,i}, i \in \mathcal{V}\backslash\{n\}$ instead of (7.32):

$$\tilde{a}_{max,i} = \begin{cases} a_{max,i} & \text{if } a_{max,i} < \tilde{a}_{max,i+1}, \\ \tilde{a}_{max,i+1} - \varepsilon & \text{else,} \end{cases} \tag{7.40}$$

where $\varepsilon > 0$ is a design variable. By introducing $\varepsilon$, every following vehicle has the ability to accelerate faster than its preceding vehicle, i.e. $\tilde{a}_{max,i} < \tilde{a}_{max,i+1}$, while (7.32) only guarantees $\tilde{a}_{max,i} \leq \tilde{a}_{max,i+1}$. In this way, the undesired behavior can be avoided without slowing down the system response. The simulation results with $\varepsilon = 0.01$ and the parameters in Table 7.3 are shown in Figure 7.7. As can be seen from Figure 7.7, the spacing errors become even smaller and the platoon exhibits string stable behaviors. However, this rule imposes more restrictions on $\tilde{a}_{max,i}$ than (7.32), and the restrictions grow with the number of vehicles in a platoon. Hence, for a long vehicular platoon, the values of $\tilde{a}_{max,i}$ of the vehicles (especially the ones in the front part of the platoon) may be too small.

Table 7.3: MPC Parameters used in [2]

| Parameter | Value |
|---|---|
| $N_p$ | 30 |
| $t_s$ | $0.01s$ |
| $q_1$ | 0.4 |
| $q_2$ | 0.125 |
| $R$ | 0.01 |
| $R_\Delta$ | 0.005 |

Table 7.4: MPC Parameters after tuning

| Parameter | Value |
|---|---|
| $N_p$ | 50 |
| $t_s$ | $0.01s$ |
| $q_1$ | 1.2 |
| $q_2$ | 2 |
| $R$ | 2 |
| $R_\Delta$ | 0.005 |

## 7.5 Conclusions

In this chapter, an MPC-based CACC approach is proposed to achieve platooning of heterogeneous vehicles with limited acceleration capabilities. The approach has been proven to be effective to improve the performance in terms of platoon cohesion (maintaining a small spacing error), while maintaining the confidentiality of the vehicle parameters (only maximum acceleration capabilities are communicated in the downstream direction of a platoon). The main key factor leading to the improvement is the adoption of a bi-directional topology instead of the commonly used unidirectional topology, which indicates that choosing a proper topology can improve the performance of synchronization. The proposed approach has been demonstrated via simulation. The influences of the MPC parameters on platooning performance are also investigated via simulations, which provides guidance for parameter choice. However, another performance requirement, string stability, is not directly enforced in the control design, which can be a direction for future work. For instance, in [126], it has been shown that with certain choices of MPC parameters, the MPC controller can guarantee string stability when the constraints are inactive.

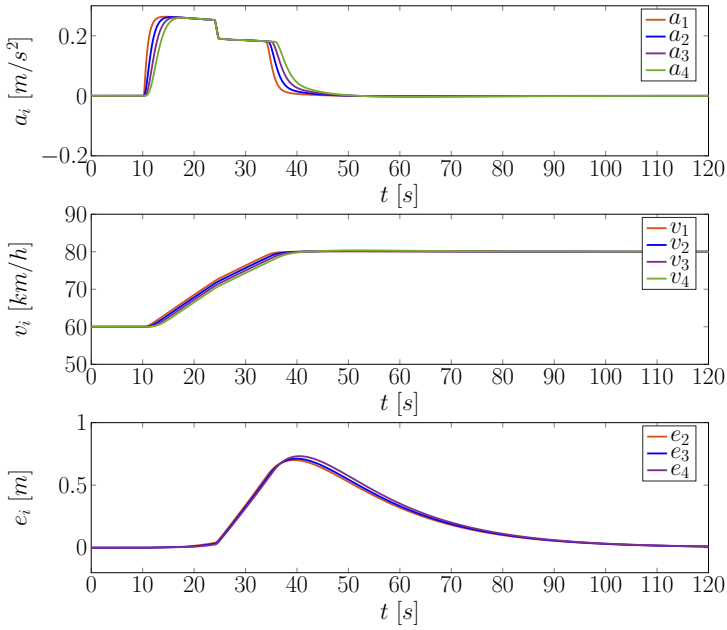



Figure 7.6: Acceleration $a_i$, speed $v_i$ and spacing error $e_i$ response of a four-truck platoon during acceleration from 60 km/h to 80 km/h using MPC-based CACC with parameters in Table 7.4.

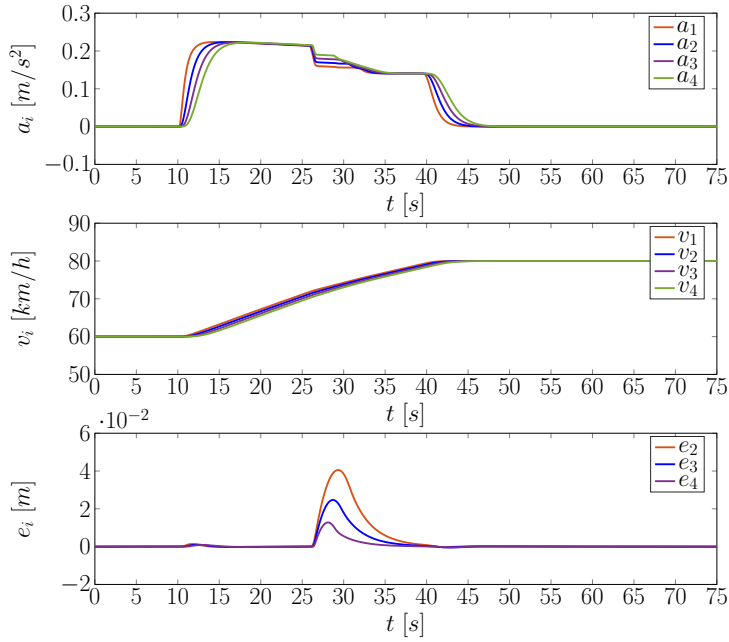



Figure 7.7: Acceleration $a_i$, speed $v_i$ and spacing error $e_i$ response of a four-truck platoon during acceleration from 60 km/h to 80 km/h using MPC-based CACC with parameters in Table 7.3 and $\varepsilon = 0.01$.

# Chapter 8

# Conclusions and recommendations

**Abstract:** This chapter summarizes the main results presented in the previous chapters and gives recommendations for future research.

## 8.1 Conclusions

In this thesis, partial synchronization in networks of diffusively time-delay coupled systems has been studied. Diffusive time-delay coupling is an important interaction found in networks of interconnected systems. First, conditions for identifying all possible synchronization patterns (presented by partial synchronization manifolds) are derived. The conditions are expressed in the form of some algebraic requirements on the adjacency matrix and restrictions on the type of input-output dynamics of the systems. A relation between the network topology and the existence of partial synchronization manifolds is established. In addition, an algorithm utilizing these conditions is developed to efficiently compute all possible partial synchronization manifolds. Second, sufficient conditions for stability of partial synchronization manifolds in such networks are derived, provided that partial synchronization manifolds are identified. These conditions are also important because the observation of partial synchronization requires not only the presence of a partial synchronization manifold but also its stability. Third, an approximate form of partial synchronization is considered, where the states of the systems in the same cluster only converge to each other up to some bound due to perturbations. Sufficient conditions for this form of partial synchronization are

also derived. Fourth, the theoretical results are demonstrated by experiments with a network of electronic neurons. Finally, as an application of synchronization, a CACC control strategy employing a bi-directional V2V communication topology to achieve platooning of connected vehicles is presented. By comparing the performances of the proposed controller with a previously proposed controller that employs a unidirectional topology, the influence of network topologies on the performance of synchronization is demonstrated.

In Chapter 3, a framework, along with an algorithm to identify possible synchronization patterns in networks of delay-coupled systems, is provided. The network setting is introduced, including the systems and couplings under consideration. The systems in a network do not necessarily have the same input-output dynamics. Two different types of couplings are considered: *invasive coupling* and *non-invasive coupling*. The invasive coupling is a diffusive coupling where time-delays appear in the received output signals from other systems, and the non-invasive coupling is a diffusive coupling where the time-delay is present in all signals. From an analysis point of view, the main difference between the two types of coupling is that non-invasive coupling vanishes when all systems are synchronized while invasive coupling does not vanish in general. The concept of a partial synchronization manifold is used to represent possible synchronization patterns. Conditions for the existence of such manifolds are derived, and an algorithm is developed, based on these conditions to compute these manifolds. The conditions show the relation between synchronization patterns and the network topology. In addition, the conditions and algorithm also apply to the case where the systems in a network have different input-output dynamics, extending the results in [3]. Furthermore, the algorithm comes with higher computational efficiency when dealing with networks with invasive coupling compared with the one developed in [3].

In Chapter 4, sufficient conditions for partial synchronization are derived. Synchronization is considered in an exact sense, which means that the differences between the states of the synchronized systems converge to zero. The conditions are derived based on the stability of the synchronization error dynamics. Synchronization errors are the differences between the states of the systems within the same clusters. It is shown that the synchronization error dynamics can be interpreted in terms of an LPV, time-delay system when linearized around the zero equilibrium. By constructing a Lyapunov functional, sufficient conditions for the stability of this system, *i.e.*, sufficient conditions for partial synchronization are derived. The conditions are expressed in the form of LMIs. By checking the conditions against the network parameters, for example, the coupling strength, the time-delay, the dependence of partial synchronization on the parameters can be revealed. More precisely, distinct synchronization regions can be identified in the parameter space. The bounds of regions can be computed by solving the LMIs. In the previous works [1, 40], it is only shown that these bounds exist, however, a tractable way to estimate them is not provided.

In Chapter 5, sufficient conditions for practical partial synchronization are derived.

Practical synchronization refers to an approximate form of synchronization where the differences between the states of synchronized systems only converge up to some bound due to perturbations induced by model mismatches, network imperfections, *etc.* In addition, the bound goes to zero when the perturbations vanish. The conditions are derived by assessing the practical stability of the synchronization error dynamics. Besides, an explicit relation between the bound and the size of perturbation is provided. Similar to the (exact) partial synchronization case, these conditions also allow to assess the parameter dependence of practical partial synchronization. The conditions together with the explicit relation between the synchronization error bound and the size of perturbations provide a tool to judge and quantify the robustness of partial synchronization when encountering perturbations.

In Chapter 6, experimental results are presented to demonstrate the theory developed in the previous chapters. The experimental setup consists of up to sixteen circuit boards, each of which consists of an electric realization of the famous Hindmarsh-Rose model. A cross correlation-based criterion is used to characterize the synchronization of the neurons from measurements. It is found that the synchronization regions in the coupling strength, time-delay $(k, \tau)$ parameter space obtained by checking the theoretical conditions are smaller than those obtained via experiments. This can be explained by the conservatism of the Lyapunov-Krosovskii method which is used to derive the theoretical results, leading to sufficient (but not necessary) conditions for synchronization or partial synchronization.

In Chapter 7, an MPC-based CACC control strategy is proposed to achieve platooning of trucks with different dynamic limitations. The proposed control design utilizes a bi-directional topology to share dynamic limitations between the vehicles such that the cohesion of a platoon can be maintained, even when dynamic limitations are encountered. In this way, the performance of synchronization (platoon cohesion) is improved compared to the commonly used unidirectional CACC control methods.

## 8.2 Recommendations

This section gives some recommendations for future research. In this thesis, the results are derived under three main assumptions: 1) the coupling functions are static and linear; 2) the time-delays are constant and uniform; 3) the network topology is fixed. So a straightforward recommendation is to extend the results to time-varying and heterogeneous time-delays, time-varying network topologies, and general nonlinear dynamical coupling functions. In what follows, a few examples are listed that serve as motivations for future research in some directions.

**Networked Control Systems** Networked Control Systems (NCSs) constitute an important class of networks of coupled systems. NCSs are control systems in which the sensors, actuators, and controllers exchange information through a shared digital communication network [127]. NCSs have received considerable attention recently, see [127] and references therein. Despite many advantages offered by NCSs, such as ease of installation and maintenance, large flexibility, *etc.*, the introduction of networks into control systems also causes some issues due to communication imperfections and constraints. For example, a finite transmission speed induces time-delays in the coupling which influence the occurrence of synchronization, as shown in the previous chapters. However, other types of communication constraints are not considered in this thesis. Another important type of communication constraints is due to signal sampling. In NCSs, a continuous-time signal has to be sampled before transmission over the digital communication network. More precisely, the continuous-time signal is firstly sampled and encoded in a digital format at the sender side, then transmitted over the network, and finally decoded at the receiver side. The sampling along with the encoding-decoding process can induce additional delays in the coupling. Note that sampling can be periodic or aperiodic in NCSs. The latter indicates that the overall delay can be time-varying. Moreover, in practical applications, systems are usually sampled at different frequencies, implying that the overall delays are not uniform. In [128, 129, 130], synchronization of sampled-data systems (NCSs) has been studied. However, in these works, the systems are assumed to have the same sampling frequency. In [131], conditions for synchronization of two coupled nonlinear oscillators under asynchronous aperiodic sampled-data transmission are provided. But, assessing the impact of non-uniform and/or aperiodic sampling on (partial) synchronization of NCSs with a *larger* network, *i.e.*, the impact of heterogeneous and/or time-varying time-delays on multiple (>2) coupled systems remains an open question. This open question is important because solving it not only helps to understand the behaviors of NCSs but also provides guidelines for control design of NCSs. By solving this open question, the relation between synchronization and the sampling parameters can be unveiled. With this relation available, one could aim at designing controllers with a maximum sample interval that still guarantees synchronization, and at the same time, minimizes the communication.

**Networks of Neurons** In Chapters 4 and 6, networks of diffusively coupled Hindmarsh-Rose neurons have been used for demonstrating the results developed in this thesis. In reality, neural networks are far more complex than the examples presented in these chapters. As mentioned in Chapter 6, there are two types of synapses: electrical synapses and chemical synapses. Unlike electrical synapses which can be modeled by linear diffusive couplings, chemical synapses are more complex and can only be modeled by nonlinear coupling functions. Therefore, to understand the collective behavior of neurons with chemical synapses, it is important to consider synchronization of interconnected systems with *nonlinear* coupling. In Chapter 6, it is

briefly mentioned that the behaviors of individual neurons in suprachiasmatic nucleus (SCN) are influenced by light, thus influencing the synchronization patterns of these neurons. It turns out light has more complicated influences on the SCN neurons. Light stimulation also alters the coupling between the SCN neurons. It has been reported that the coupling between SCN neurons is subject to diurnal variation [132, 133]. The authors of [133] observe in their experiments that the electrical coupling between SCN neurons is stronger in the middle of the day than in the late day or early night. However, the relation between the change of coupling and the synchronous behaviors of SCN neurons is still unclear. This motivates an investigation on synchronization of coupled systems with *time-varying* coupling.

**Vehicle Platooning** When designing control strategies for vehicle platooning in Chapter 7, the structure of the network is considered to be fixed. However, this is not always the case in practice. It is possible that some vehicles have to leave the platoon while some other vehicles will join the platoon at some point of time. This type of scenario motivates to investigate synchronization of coupled systems in the presence of a *time-varying* network topology.

Of course, the examples mentioned above do not cover all the open questions and interesting applications related to (partial) synchronization of coupled systems. These examples are just the tip of the iceberg. It is for future research to investigate the open questions and expand the applications of synchronization.

# Appendix A

# Partial synchronization manifolds in networks with multiple weights

In this appendix, the existence conditions for partial synchronization manifolds in networks with multiple weights are presented.

Suppose a network $\mathscr{G}$ has $r$ characteristic scalar interaction weights $\omega_\ell, \ell = 1,\dots,r$. We can decompose it as

$$\mathscr{G}(\mathscr{V},\mathscr{E},A) = \mathscr{G}_1(\mathscr{V},\mathscr{E}_1,\omega_1 A_1) \oplus \cdots \oplus \mathscr{G}_r(\mathscr{V},\mathscr{E}_r,\omega_r A_r), \tag{A.1}$$

where $A_\ell$ are adjacency matrices with non-negative integer entries and $\mathscr{E}_\ell$ is the set of edges corresponding to $A_\ell$. Here $\oplus$ denotes a sum on graphs with the same set of nodes $\mathscr{V}$, which joins the sets of edges and sums the adjacency matrices. Therefore,

$$\mathscr{G}_1(\mathscr{V},\mathscr{E}_1,\omega_1 A_1) \oplus \cdots \oplus \mathscr{G}_r(\mathscr{V},\mathscr{E}_r,\omega_r A_r) = \mathscr{G}\left(\mathscr{V},\cup_{\ell=1}^r \mathscr{E}_\ell, \sum_{\ell=1}^r \omega_\ell A_\ell\right),$$

which means

$$A = \sum_{\ell=1}^{r} \omega_\ell A_\ell \tag{A.2}$$

and

$$\mathscr{E} = \cup_{\ell=1}^r \mathscr{E}_\ell. \tag{A.3}$$

Here, the numbers $\omega_\ell$, $\ell = 1,\dots,r$ are called *basis weights*. This decomposition can be used, for example, when systems are communicating using $r$ different

types of communication channels, then $A_j$ describes the network determined by the communication using the channel of type $j$.

Clearly, satisfying the row sum criteria in item 2 of Theorem 3.2.3 for all adjacency matrices $A_\ell$ simultaneously is a *sufficient* condition for $\mathscr{M}(\Pi)$ to be a partial synchronization manifold. However, this condition becomes both *sufficient* and *necessary* when the basis weights are rationally independent. The basis weights are rationally independent if and only if the following implication holds for any integer numbers $q_1,\dots,q_r$

$$\sum_{\ell=1}^{r} \omega_\ell q_\ell = 0 \Rightarrow q_\ell = 0 \quad \forall \ell = 1,\dots,r. \tag{A.4}$$

The existence conditions of partial synchronization manifolds in those networks are given in the following theorem:

**Theorem A.1.1.** *Consider a graph $\mathscr{G}$ that decomposes as* (A.1) *and suppose that the numbers $\omega_1,\dots,\omega_r$ are rationally independent. Given an adjacency matrix $A$ and a permutation matrix $\Pi$ of the same dimension, and assuming that $C_iB_i$, $i=1,\dots,N$ are similar to positive definite matrices, the following statements are equivalent:*

*(1) $\mathscr{M}(\Pi)$ is a partial synchronization manifold for (3.1) and (3.2), respectively (3.1) and (3.3);*

*(2) for each $\ell = 1,\dots,r$, all blocks, respectively all off-diagonal blocks, of all block-structured matrices $R^\top A_l R$, partitioned in blocks of size $\kappa_i \times \kappa_j$ according to* (3.8)*, have constant row-sums, and, in addition, $\mathscr{F}$, $\mathscr{B}$ and $\mathscr{C}$ defined by*

$$\mathscr{F} := \begin{bmatrix} f_1(\cdot) \\ f_2(\cdot) \\ \vdots \\ f_N(\cdot) \end{bmatrix}, \mathscr{B} := \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix}, \mathscr{C} := \begin{bmatrix} C_1^\top \\ C_2^\top \\ \vdots \\ C_N^\top \end{bmatrix}$$

*satisfy the conditions*

$$\mathscr{F} = (\Pi \otimes I_n)\mathscr{F},\ \mathscr{B} = (\Pi \otimes I_n)\mathscr{B},\ \mathscr{C} = (\Pi \otimes I_n)\mathscr{C}. \tag{A.5}$$

Like Theorem 3.2.3, this is also an extension of Theorems 5 and 6 in [3] to allow for different systems at the nodes of the networks.

By comparing Theorem A.1.1 with Theorem 3.2.3, it can be concluded that the method presented in Section 3.3 can also be used to compute the partial synchronization manifolds of networks with multiple weights. The difference is that the row-sums check should be applied to all $A_\ell, \ell = 1,\dots,r$ provided that the weights $\omega_l$ are rationally

independent. The benefit of the decomposition of the networks according to the weights is that all the decomposed adjacency matrices $A_\ell$ only contain integer entries such that the method for computing partial synchronization manifolds can be carried out in exact arithmetic.

# Appendix B

# Construction of the row-sum inequality indicator vector $v_S$

The construction of the vector $v_S = [S_1 \cdots S_N]^\top \in \mathbb{Z}^N$ is explained, which indicates the (in)equalities between the row-sums of the adjacency matrix $A$. Here, $S_i = S_j$ if and only if the $i$-th row and $j$-th row of $A$ have the same row sum. The vector $v_S$ will be then used to determine which nodes can be put into the same clusters during the generation of all possible partitions. In what follows, we introduce the procedure to construct $v_S$ for both cases where the adjacency matrix $A$ is treated as a single matrix and where $A$ is decomposed into multiple matrices according to (A.2).

When the adjacency matrix $A$ is treated as one single matrix, the process to acquire $v_S$ is straightforward. If the adjacency matrix $A$ only contains integer elements, $v_S$ can be determined by simply calculating the row sums of $A$:

$$v_S = A\mathbf{1}_N,$$

where $\mathbf{1}_N$ is the $N$-dimensional vector with all entries equal to 1. If the adjacency matrix $A$ contains non-integer element(s), $v_S$ can be obtained by comparing the row sums of $A$ using a predefined tolerance.

Denote the number of distinct elements of $v_S$ by $d$. If $d = N$, no partial synchronization manifolds exist. The algorithm for computing partial synchronization manifold will stop and return a statement that no partial synchronization manifolds exist.

When the adjacency matrix $A$ is decomposed in the form of (A.2), some extra effort is needed to acquire $v_S$.

First, using the same method as above, we can obtain the vector $v_{\ell,S}$ indicating the (in)equalities the row-sums of each matrix $A_\ell$ ($\ell = 1,2,\ldots,r$):

$$v_{\ell,S} = [S_{\ell,1} \ \cdots \ S_{\ell,N}]^\top.$$

Again, $S_{\ell,i} = S_{\ell,j}$ if and only if the $i$-th and $j$-th rows of the matrix $A_\ell$ have the same row-sum.

Second, we determine the common row sum indicator $v_S$ for all adjacency matrices. In $v_S$, $S_i = S_j$ if and only the $i$-th and $j$-th rows of all matrices $A_\ell, \ell = 1,\ldots,r$ have the same row-sum.

We start with the first row of the matrices. For the first row, the following vectors for $\ell = 1,\ldots,r$ are constructed:

$$G_{\ell,1} = \begin{bmatrix} g_{\ell,1} \\ g_{\ell,2} \\ \vdots \\ g_{\ell,N} \end{bmatrix} \text{ with } g_{\ell,i} = \begin{cases} 1 & \text{for } i = 1 \\ 1 & \text{for } i = 2,\ldots,N \text{ if } S_{\ell,i} = S_{\ell,1} \\ 0 & \text{otherwise} \end{cases}.$$

Then, these $r$ vectors are combined into one vector $H_1$ by element-wise multiplication

$$H_1 = \begin{bmatrix} 1 \\ \prod_{\ell=1}^r g_{\ell,2} \\ \vdots \\ \prod_{\ell=1}^r g_{\ell,N} \end{bmatrix} = \begin{bmatrix} 1 \\ h_{1,2} \\ \vdots \\ h_{1,N} \end{bmatrix}.$$

Here, the $i$-th row and the 1st row in every adjacency matrix $(A_1, A_2, \ldots, A_r)$ have the same row sum if and only if $h_{1,i} = 1$ for $i = 2,\ldots,N$.

For the subsequent rows, the procedure specified by Algorithm 2 is used to construct the vectors $H_i$, $i = 2,\ldots,d$, where $d$ is again the number of distinct elements of the vector $v_S$. Like $H_1$, the vectors $H_2,\ldots,H_d$ are used to indicate which rows (among the subsequent rows) in every adjacency matrix have the same row-sum. For example, given two decomposed adjacency matrices $A_1$ and $A_2$

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 2 \\ 2 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 \end{bmatrix},$$

we have $H_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\top$ by the definition of $H_1$, and $H_2 = \begin{bmatrix} 0 & 1 & 1 & 1 \end{bmatrix}^\top$ by the procedure in Algorithm 2. Here, $H_2$ indicates that the second, the third and the forth rows in each adjacency matrix have the same row-sum.

**Algorithm 2:** Calculate $H_i$ for $v_S$

**Input:** $v_{\ell,S}, \ell = 1,\dots,r$ and $H_1$
**Output:** $H_i, i = 2,\dots,d$
$d \leftarrow 1$;
$q \leftarrow 2$;
**while** $q \leq N$ **do**
  **if** $\sum_{i=1}^{d} h_{i,q} \neq 0$ **then**
    $q \leftarrow q+1$;
    **continue**;
  **else**
    $d \leftarrow d+1$ ;
    construct the following vectors for $\ell = 1,\dots,r$:

$$G_{\ell,d} = \begin{bmatrix} g_{\ell,1} \\ g_{l,2} \\ \vdots \\ g_{\ell,N} \end{bmatrix} \text{ with } g_{l,i} =$$

$$\begin{cases} 1 & \text{for } i = q \\ 1 & \text{for } i = 1,\dots,q-1,q+1,\dots,N \text{ if } S_{l,i} = S_{l,q}\,; \\ 0 & \text{otherwise} \end{cases}$$

    combine these $r$ vectors into one vector $H_d$ by element-wise multiplication:

$$H_d = \begin{bmatrix} \prod_{\ell=1}^{r} g_{\ell,1} \\ \vdots \\ \prod_{\ell=1}^{r} g_{\ell,q-1} \\ 1 \\ \prod_{\ell=1}^{r} g_{\ell,q+1} \\ \vdots \\ \prod_{\ell=1}^{r} g_{\ell,N} \end{bmatrix} = \begin{bmatrix} h_{d,1} \\ \vdots \\ h_{d,q-1} \\ 1 \\ h_{d,q+1} \\ \vdots \\ h_{d,N} \end{bmatrix};$$

    $q \leftarrow q+1$ ;
  **end**
**end**

Collect all $H_i, i = 1,\dots,d$ in one matrix $H$

$$H = \begin{bmatrix} H_1 & H_2 & \cdots & H_d \end{bmatrix}, \quad d \leq N.$$

Note that for $d = N$, no partial synchronization manifolds exist. The algorithm will stop with the statement that no partial synchronization manifolds exist.

Finally, we multiply $H$ with the vector $v = [1\ 2\ \cdots\ d]^\top$ to obtain $v_S$:

$$v_S = Hv = [S_1\ \cdots S_N]^\top.$$

We remark that the procedure presented here can be also used for networks that are decomposed according to the presence of multiple delays.

# Appendix C

# Proof of Theorem 4.4.1

*Proof.* Consider the following parameter-dependent Lyapunov-Krasovskii functional,

$$V(t) = \mathscr{X}^\top(t)P(\eta)\mathscr{X}(t) + V_1(t) + V_2(t) + V_3(t), \tag{C.1}$$

where

$$V_j(t) := \int_{t-j\frac{\tau}{3}}^{t-(j-1)\frac{\tau}{3}} \mathscr{X}^\top(s)Q_j\mathscr{X}(s)\mathrm{d}s + \int_{-j\frac{\tau}{3}}^{-(j-1)\frac{\tau}{3}} \int_{t+\theta}^{t} \dot{\mathscr{X}}^\top(s)Z_j\dot{\mathscr{X}}(s)\mathrm{d}s\mathrm{d}\theta. \tag{C.2}$$

Note that this Lyapunov-Krasovskii functional requires the initial functions to be differentiable in $[-\tau,0]$ because of the presence of state derivatives in the last terms. So, this Lyapunov-Krasovskii functional is, in fact, defined in $\mathscr{C}([-2\tau,0],\mathbb{R}^{Nn-\kappa n})$, which is a subspace of $\mathscr{C}([-\tau,0],\mathbb{R}^{Nn-\kappa n})$. Hence, the conditions derived from this functional are sufficient for the stability problem in $[-\tau,0]$.

Differentiating (C.1) along the solution of (4.14) yields

$$\begin{aligned}
\dot{V}(t) =& \dot{\mathscr{X}}(t)^\top P(\eta)\mathscr{X}(t) + \mathscr{X}^\top(t)P(\eta)\dot{\mathscr{X}}(t) + \mathscr{X}^\top(t)\dot{P}(\eta)\mathscr{X}(t) + \sum_{j=1}^{3}\dot{V}_j(t) \\
=& \mathscr{X}^\top\left[\mathscr{A}^\top(\eta)P(\eta) + P(\eta)\mathscr{A}(\eta) + \dot{P}(\eta)\right]\mathscr{X}(t) \\
&+ \mathscr{X}^\top(t-\tau)\mathscr{A}_d^\top P(\eta)\mathscr{X}(t) + \mathscr{X}^\top(t)P(\eta)\mathscr{A}_d\mathscr{X}(t-\tau) + \sum_{j=1}^{3}\dot{V}_j(t),
\end{aligned} \tag{C.3}$$

where

$$\begin{aligned}\dot{V}_j(t) =& \mathscr{X}^\top\left(t-(j-1)\frac{\tau}{3}\right)Q_j\mathscr{X}\left(t-(j-1)\frac{\tau}{3}\right)-\mathscr{X}^\top\left(t-j\frac{\tau}{3}\right)Q_j\mathscr{X}\left(t-\frac{\tau}{3}\right)\\&+\frac{\tau}{3}\dot{\mathscr{X}}^\top(t)Z_j\dot{\mathscr{X}}(t)-\int_{t-j\frac{\tau}{3}}^{t-(j-1)\frac{\tau}{3}}\dot{\mathscr{X}}^\top(s)Z_j\dot{\mathscr{X}}(s)\mathrm{d}s\\=&\mathscr{X}^\top\left(t-(j-1)\frac{\tau}{3}\right)Q_j\mathscr{X}\left(t-(j-1)\frac{\tau}{3}\right)-\mathscr{X}^\top\left(t-j\frac{\tau}{3}\right)Q_j\mathscr{X}\left(t-\frac{\tau}{3}\right)\\&+\frac{\tau}{3}\Big[\mathscr{X}^\top(t)\mathscr{A}^\top(\eta)Z_j\mathscr{A}(\eta)\mathscr{X}(t)+\mathscr{X}^\top(t)\mathscr{A}^\top(\eta)Z_j\mathscr{A}_d\mathscr{X}(t-\tau)\\&\quad+\mathscr{X}^\top(t-\tau)\mathscr{A}_d^\top Z_j\mathscr{A}(\eta)\mathscr{X}(t)+\mathscr{X}^\top(t-\tau)\mathscr{A}_d^\top Z_j\mathscr{A}_d\mathscr{X}(t-\tau)\Big]\\&-\int_{t-j\frac{\tau}{3}}^{t-(j-1)\frac{\tau}{3}}\dot{\mathscr{X}}^\top(s)Z_j\dot{\mathscr{X}}(s)\mathrm{d}s.\end{aligned}\tag{C.4}$$

Applying Jensen's inequality[1] to the right side of (C.4), we arrive at

$$\begin{aligned}\dot{V}_j(t) \leq& \mathscr{X}^\top\left(t-(j-1)\frac{\tau}{3}\right)Q_j\mathscr{X}\left(t-(j-1)\frac{\tau}{3}\right)-\mathscr{X}^\top\left(t-j\frac{\tau}{3}\right)Q_j\mathscr{X}\left(t-\frac{\tau}{3}\right)\\&+\frac{\tau}{3}\Big[\mathscr{X}^\top(t)\mathscr{A}^\top(\eta)Z_j\mathscr{A}(\eta)\mathscr{X}(t)+\mathscr{X}^\top(t)\mathscr{A}^\top(\eta)Z_j\mathscr{A}_d\mathscr{X}(t-\tau)\\&\quad+\mathscr{X}^\top(t-\tau)\mathscr{A}_d^\top Z_j\mathscr{A}(\eta)\mathscr{X}(t)+\mathscr{X}^\top(t-\tau)\mathscr{A}_d^\top Z_j\mathscr{A}_d\mathscr{X}(t-\tau)\Big]\\&-\frac{3}{\tau}\int_{t-j\frac{\tau}{3}}^{t-(j-1)\frac{\tau}{3}}\dot{\mathscr{X}}^\top(s)\mathrm{d}sZ_j\int_{t-j\frac{\tau}{3}}^{t-(j-1)\frac{\tau}{3}}\dot{\mathscr{X}}(s)\mathrm{d}s.\end{aligned}\tag{C.5}$$

[1]For any positive definite matrix $M\in\mathbb{R}^{n\times n}$, scalar $\tau>0$ and function $\phi\in\mathscr{C}([-\tau,0],\mathbb{R}^n)$, assuming that the integrations concerned are well defined, the inequality holds [68]:

$$\tau\int_{-\tau}^{0}\phi^\top(s)M\phi(s)\mathrm{d}s\geq\int_{-\tau}^{0}\phi(s)\mathrm{d}sM\int_{-\tau}^{0}\phi^\top(s)\mathrm{d}s.$$

It follows that

$$\begin{aligned}\dot{V}(t) \leq & \mathscr{X}^{\top}\left[\mathscr{A}^{\top}(\eta)P(\eta)+P(\eta)\mathscr{A}(\eta)+\dot{P}(\eta)\right]\mathscr{X}(t)\\ & +\mathscr{X}^{\top}(t-\tau)\mathscr{A}_d^{\top}P(\eta)\mathscr{X}(t)+\mathscr{X}^{\top}(t)P(\eta)\mathscr{A}_d\mathscr{X}(t-\tau)\\ & +\sum_{j=1}^{3}\left[\mathscr{X}^{\top}\left(t-(j-1)\frac{\tau}{3}\right)Q_j\mathscr{X}\left(t-(j-1)\frac{\tau}{3}\right)-\mathscr{X}^{\top}\left(t-j\frac{\tau}{3}\right)Q_j\mathscr{X}\left(t-\frac{\tau}{3}\right)\right]\\ & +\sum_{j=1}^{3}\frac{\tau}{3}\left[\mathscr{X}^{\top}(t)\mathscr{A}^{\top}(\eta)Z_j\mathscr{A}(\eta)\mathscr{X}(t)+\mathscr{X}^{\top}(t)\mathscr{A}^{\top}(\eta)Z_j\mathscr{A}_d\mathscr{X}(t-\tau)\right.\\ & \left.+\mathscr{X}^{\top}(t-\tau)\mathscr{A}_d^{\top}Z_j\mathscr{A}(\eta)\mathscr{X}(t)+\mathscr{X}^{\top}(t-\tau)\mathscr{A}_d^{\top}Z_j\mathscr{A}_d\mathscr{X}(t-\tau)\right]\\ & -\sum_{j=1}^{3}\frac{3}{\tau}\left[\int_{t-j\frac{\tau}{3}}^{t-(j-1)\frac{\tau}{3}}\dot{\mathscr{X}}^{\top}(s)\mathrm{d}sZ_j\int_{t-j\frac{\tau}{3}}^{t-(j-1)\frac{\tau}{3}}\dot{\mathscr{X}}(s)\mathrm{d}s\right].\end{aligned} \tag{C.6}$$

The Leibniz-Newton formula provides

$$\mathscr{X}\left(t-(j-1)\frac{\tau}{3}\right)-\mathscr{X}\left(t-j\frac{\tau}{3}\right)-\int_{t-j\frac{\tau}{3}}^{t-(j-1)\frac{\tau}{3}}\dot{\mathscr{X}}(t)\mathrm{d}s=0, j=1,2,3. \tag{C.7}$$

Introducing the free weighting matrices ([90]) $G_j\in\mathbb{R}^{(4Nn-4\kappa n)\times(Nn-\kappa n)}, j=1,2,3$, we have

$$\xi^{\top}G_j\left[\mathscr{X}\left(t-(j-1)\frac{\tau}{3}\right)-\mathscr{X}\left(t-j\frac{\tau}{3}\right)-\int_{t-j\frac{\tau}{3}}^{t-(j-1)\frac{\tau}{3}}\dot{\mathscr{X}}(t)\mathrm{d}s\right]=0, \tag{C.8}$$

and

$$\left[\mathscr{X}\left(t-(j-1)\frac{\tau}{3}\right)-\mathscr{X}\left(t-j\frac{\tau}{3}\right)-\int_{t-j\frac{\tau}{3}}^{t-(j-1)\frac{\tau}{3}}\dot{\mathscr{X}}(t)\mathrm{d}s\right]G_j^{\top}\xi=0, \tag{C.9}$$

where $\xi=\left[\mathscr{X}^{\top}(t)\quad\mathscr{X}^{\top}\left(t-\frac{\tau}{3}\right)\quad\mathscr{X}^{\top}\left(t-\frac{2\tau}{3}\right)\quad\mathscr{X}^{\top}(t-\tau)\right]^{\top}$ and $j=1,2,3$.

Adding the terms on the left side of (C.8) and (C.9) to the right side of (C.6), yields

$$\dot{V}\leq\bar{\xi}^{\top}\begin{bmatrix}\Phi & \frac{\tau}{3}G_1 & \frac{\tau}{3}G_2 & \frac{\tau}{3}G_3\\ \frac{\tau}{3}G_1^{\top} & -\frac{\tau}{3}Z_1 & \mathbf{0} & \mathbf{0}\\ \frac{\tau}{3}G_2^{\top} & \mathbf{0} & -\frac{\tau}{3}Z_2 & \mathbf{0}\\ \frac{\tau}{3}G_3^{\top} & \mathbf{0} & \mathbf{0} & -\frac{\tau}{3}Z_3\end{bmatrix}\bar{\xi}^{\top}, \tag{C.10}$$

with $\bar{\xi} = \left[\xi^\top \quad \frac{3}{\tau}\int_{t-\frac{\tau}{3}}^{t} \dot{\mathscr{X}}^\top(s)\mathrm{d}s \quad \frac{3}{\tau}\int_{t-\frac{2\tau}{3}}^{t-\frac{\tau}{3}} \dot{\mathscr{X}}^\top(s)\mathrm{d}s \quad \frac{3}{\tau}\int_{t-\tau}^{t-\frac{2\tau}{3}} \dot{\mathscr{X}}^\top(s)\mathrm{d}s\right]^\top$, and the matrix $\Phi$ is defined as below

$$\begin{cases} \Phi := \Phi_1 + \mathrm{He}\left(\Lambda_0 P(\eta)\bar{\mathscr{A}} + \sum_{j=1}^{3} G_j\Lambda_j\right) + \bar{\mathscr{A}}^\top\left(\frac{\tau}{3}\sum_{j=1}^{3} Z_j\right)\bar{\mathscr{A}}, \\ \Phi_1 := \mathrm{diag}\Big(Q_1 + \frac{\partial P(\eta)}{\partial \eta}\dot{\eta}, -Q_1 + Q_2, \\ \qquad\qquad -Q_2 + Q_3, -Q_3\Big), \\ \bar{\mathscr{A}} := \left[\begin{array}{cccc} \mathscr{A}(\eta) & \mathbf{0} & \mathbf{0} & \mathscr{A}_d \end{array}\right], \\ \Lambda_0 := \left[\begin{array}{cccc} I & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right]^\top, \\ \Lambda_1 := \left[\begin{array}{cccc} I & -I & \mathbf{0} & \mathbf{0} \end{array}\right], \\ \Lambda_2 := \left[\begin{array}{cccc} \mathbf{0} & I & -I & \mathbf{0} \end{array}\right], \\ \Lambda_3 := \left[\begin{array}{cccc} \mathbf{0} & \mathbf{0} & I & -I \end{array}\right], \end{cases} \tag{C.11}$$

where $\mathrm{He}(A) := (A + A^\top)$.

That is, when the following inequality holds, $\dot{V} < 0$, which ensures the stability of the system (4.14)

$$\begin{bmatrix} \Phi & \frac{\tau}{3}G_1 & \frac{\tau}{3}G_2 & \frac{\tau}{3}G_3 \\ * & -\frac{\tau}{3}Z_1 & \mathbf{0} & \mathbf{0} \\ * & * & -\frac{\tau}{3}Z_2 & \mathbf{0} \\ * & * & * & -\frac{\tau}{3}Z_3 \end{bmatrix} < 0. \tag{C.12}$$

□

# Appendix D

# Invariant set of coupled Hindmarsh-Rose neurons

In this appendix, it is proved that an invariant set exists in the state space of a network of delay-coupled Hindmarsh-Rose neurons. Besides, the method to compute such an invariant set is given. This work is inspired by [106] where networks of Hindmarsh-Rose neurons interconnected via delay free coupling are considered.

Here, a network of $N$ Hindmarsh-Rose neurons of the following form is considered

$$\begin{aligned}
\dot{x}_{i,1}(t) &= c - dx_{i,3}^2(t) - x_{i,1}(t),\\
\dot{x}_{i,2}(t) &= r(s(x_{i,3}(t) + v_0) - x_{i,2}(t)),\\
\dot{x}_{i,3}(t) &= -ax_{i,3}^3(t) + bx_{i,3}^2(t) + x_{i,1}(t) - x_{i,2}(t) + E_m + u_i(t),\\
y_i(t) &= x_{i,3}(t),
\end{aligned} \tag{D.1}$$

where $i = 1, 2, \ldots, N$.

The coupling between the neurons is given by

$$u_i(t) = k \sum_{j=1}^{N} a_{i,j}(y_j(t-\tau) - y_i(t)), \tag{D.2}$$

where $k > 0$ is the coupling strength, $a_{i,j}$ is the $ij$th entry of the adjacency matrix $A$, and $\tau > 0$ is the time-delay.

The state space of the network is $\mathscr{C}([-\tau,0],\mathbb{R}^{3N})$, and the state $x(t+\theta)$ is defined as

$$x(t+\theta)=\begin{bmatrix}x_1(t+\theta)\\ \vdots \\ x_N(t+\theta)\end{bmatrix}, \tag{D.3}$$

where $x_i(t+\theta)=[x_{i,1}(t+\theta)\ x_{i,2}(t+\theta)\ x_{i,3}(t+\theta)]^\top$, $\theta\in[-\tau,0]$, $i=1,2,\ldots,N$.

## D.1 Existence of invariant set

In what follows, the existence of an invariant set in the state space of $N$ delay-coupled neurons is proved.

First, recall that the Hindmarsh-Rose neuron is semipassive, which indicates that each neuron has an invariant set when uncoupled ($u_i=0$). That is, there exists a box $\mathbb{B}:=[x_{l,1},x_{h,1}]\times[x_{l,2},x_{h,2}]\times[x_{l,3},x_{h,3}]\in\mathbb{R}^3$ that is positively invariant under the dynamics of an uncoupled neuron with $x_{l,p}$ and $x_{h,p}$, $p=1,2,3$, being constant real numbers.

Second, it can be shown that the set $\mathbb{B}_I=\{x(t+\theta)\in\mathscr{C}([-\tau,0],\mathbb{R}^{3N})\,|\,x_{l,j}\le x_{i,j}(t+\theta)\le x_{h,j}, i=1,\ldots,N, j=1,2,3,\forall\theta\in[-\tau,0]\}$ is a positively invariant set of a network of $N$ neurons coupled via the invasive coupling (D.2).

Suppose $N$ uncoupled neurons are initialized inside this invariant set $\mathbb{B}$ at time $t_0$. At time $t_1$ ($t_1>t_0+\tau$), the neurons are coupled via the invasive coupling (D.2). It can be shown that the values of state variables of each neuron will remain within the box $\mathbb{B}$ as of time $t_1$.

From (D.1), it can be seen that the input $u_i$ only has a direct impact on the third state variable $x_{i,3}$. Therefore, the following analysis begins with $x_{i,3}$.

Consider the two following extreme cases:

1. The state variable $x_{i,3}$ reaches the upper boundary of the box $\mathbb{B}$ at time $t_1$, *i.e.*, $x_{i,3}(t_1)=x_{h,3}(t_1)$.

   Since $\mathbb{B}$ is an invariant set for the uncoupled neuron, we have

   $$-ax_{i,3}^3(t)+bx_{i,3}^2(t_1)+x_{i,1}(t_1)-x_{i,2}(t_1)+E_m\le 0. \tag{D.4}$$

   In addition, we have

   $$x_{j,3}(t_1-\tau)\le x_{h,3}=x_{i,3}(t_1),\forall j=1,\ldots,N, \tag{D.5}$$

leading to

$$u_i(t_1) = k\sum_{j=1}^{N} a_{i,j}(x_{j,3}(t_1-\tau) - x_{i,3}(t)) \leq 0. \tag{D.6}$$

Combining (D.4) with (D.6) yields

$$\dot{x}_{i,3}(t_1) = -ax_{i,3}^3 + bx_{i,3}^2(t_1) + x_{i,1}(t_1) - x_{i,2}(t_1) + E_m + u_i(t_1) \leq 0. \tag{D.7}$$

Therefore, $x_{i,3}$ cannot cross the upper boundary.

2. The state variable $x_{i,3}$ reaches the lower boundary of the box $\mathbb{B}$ at time $t_1$, *i.e.*, $x_{i,3}(t_1) = x_{l,3}$.

   Since $\mathbb{B}$ is an invariant set for the uncoupled neuron, we have

$$-ax_{i,3}^3(t_1) + bx_{i,3}^2(t_1) + x_{i,1}(t_1) - x_{i,2}(t_1) + E_m \geq 0. \tag{D.8}$$

   In addition, we have

$$x_{j,3}(t_1-\tau) \geq x_{l,3} = x_{i,3}(t_1), \forall j = 1,\ldots,N, \tag{D.9}$$

   leading to

$$u_i(t_1) = k\sum_{j=1}^{N} a_{i,j}(x_{j,3}(t_1-\tau) - x_{i,3}(t)) \geq 0. \tag{D.10}$$

   Combining (D.8) with (D.10) yields

$$\dot{x}_{i,3}(t_1) = -ax_{i,3}^3 + bx_{i,3}^2(t_1) + x_{i,1}(t_1) - x_{i,2}(t_1) + E_m + u_i(t_1) \geq 0. \tag{D.11}$$

   Therefore, $x_{i,3}$ cannot cross the lower boundary.

So, it can be concluded that $x_{i,3}$ will stay between the boundaries. Meanwhile, since $u_i$ does not appear in the first two equations in (D.1), it has no direct immediate influence on the state variables $x_{i,1}$ and $x_{i,2}$, which indicates they will also stay inside the box. By putting all these observations together, it can be concluded that $\mathbb{B}_I$ is an invariant set of the $N$ neurons coupled via (D.2).

## D.2 Computation of invariant set

The invariant set $\mathbb{B}_I$ is actually determined by the values of six constants $x_{l,p}$ and $x_{h,p}, p = 1,2,3$ in the invariant box $\mathbb{B}$. These constants satisfy the following conditions

$$\begin{aligned}
c - dx_{i,3}^2 - x_{h,1} \le 0, \quad & \forall x_{i,3} \in [x_{l,3}, x_{h,3}], \\
-c + dx_{i,3}^2 + x_{l,1} \le 0, \quad & \forall x_{i,3} \in [x_{l,3}, x_{h,3}], \\
r(s(x_{i,3} + v_0) - x_{h,2}) \le 0, \quad & \forall x_{i,3} \in [x_{l,3}, x_{h,3}], \\
-r(s(x_{i,3} + v_0) - x_{l,2}) \le 0, \quad & \forall x_{i,3} \in [x_{l,3}, x_{h,3}], \\
-ax_{h,3}^3 + bx_{h,3}^2 + x_{i,1} - x_{i,2} + E_m \le 0, \quad & \forall (x_{i,1}, x_{i,2}) \in [x_{l,1}, x_{h,1}] \times [x_{l,2}, x_{h,2}], \\
ax_{l,3}^3 - bx_{l,3}^2 - x_{i,1} + x_{i,2} - E_m \le 0, \quad & \forall (x_{i,1}, x_{i,2}) \in [x_{l,1}, x_{h,1}] \times [x_{l,2}, x_{h,2}].
\end{aligned} \tag{D.12}$$

First, because the second and third inequalities are linear, we can set

$$\begin{aligned}
x_{h,2} &= s(x_{h,3} + v_0), \\
x_{l,2} &= s(x_{l,3} + v_0).
\end{aligned} \tag{D.13}$$

Second, we note that

$$x_{h,1} = c. \tag{D.14}$$

Thus, $x_{l,1}$, $x_{3,l}$, and $x_{3,h}$ should satisfy

$$\begin{aligned}
-ax_{h,3}^3 + bx_{h,3}^2 + c - s(x_{l,3} + v_0) + E_m \le 0, \\
ax_{l,3}^3 - bx_{l,3}^2 - x_{l,1} + s(x_{h,3} + v_0) - E_m \le 0, \\
-c + 5\max\{x_{3,l}^2, x_{3,h}^2\} + x_{l,1} \le 0,
\end{aligned} \tag{D.15}$$

Finally, to find the smallest possible invariant box $\mathbb{B}$, the values of $x_{l,1}$, $x_{3,l}$, and $x_{3,h}$ are then determined by minimizing

$$V(x_{l,1}, x_{3,l}, x_{3,h}) = (x_{3,h} - x_{3,l})^2, \tag{D.16}$$

subject to the constraints (D.15).

Substituting $a = 1$, $b = 3$, $c = 1$, $d = 5$, $E_m = 3.25$ and $v_0 = 1.618$ into this optimization problem and solving it with the MATLAB nonlinear programming solver *fmincon*, we obtain the invariant box $\mathbb{B}$ for each neuron in the example of Chapter 4:

$$\mathbb{B} = [-73.0705, 1] \times [-3.6994, 3.8489] \times [-8.3256, 21.8676]. \tag{D.17}$$

# Appendix E

# Proof of Theorem 5.4.1

*Proof.* Considering a Lyapunov-Krasovskii functional

$$v(\mathscr{X}_t) = \mathscr{X}^\top(t)P\mathscr{X}(t) + \int_{-\tau}^{0} \mathscr{X}^\top(t+\theta)e^{2\sigma\theta}Q\mathscr{X}(t+\theta)\mathrm{d}\theta, \tag{E.1}$$

we have

$$\alpha_1\|\mathscr{X}(t)\|^2 \leq v(\mathscr{X}_t) \leq \alpha_2\|\mathscr{X}_t\|_s^2. \tag{E.2}$$

where $\alpha_1 = \lambda_{\min}(P)$ and $\alpha_2 = \lambda_{\max}(P) + \tau\lambda_{\max}(Q)$.

Then

$$\begin{aligned}\frac{d}{dt}v(\mathscr{X}_t) =& 2\mathscr{X}^\top(t)P[F(\mathscr{X}(t),\mathscr{X}_r(t)) - kA_0\mathscr{X}(t) + kA_1\mathscr{X}(t-\tau) + \Xi(t,\tau,\Delta)] \\ &+ \mathscr{X}^\top(t)Q\mathscr{X}(t) - \mathscr{X}^\top(t-\tau)e^{-2\sigma\tau}Q\mathscr{X}(t-\tau) \\ &- 2\sigma\int_{-\tau}^{0}\mathscr{X}^\top(t+\theta)e^{-2\sigma\theta}Q\mathscr{X}(t+\theta)d\theta.\end{aligned} \tag{E.3}$$

Here, $2\mathscr{X}^\top(t)P\Xi(t,\tau,\Delta) \leq 2\|\mathscr{X}(t)\|\|P\|\|\Xi\|$, which implies

$$2\mathscr{X}^\top(t)P\Xi(t) \leq 2\frac{\gamma\|P\|}{\sqrt{\alpha_1}}\sqrt{v(\mathscr{X}_t)}.$$

The other term satisfies

$$2\mathscr{X}^\top(t)PF(\mathscr{X}(t),\mathscr{X}_r(t)) \leq 2\|\mathscr{X}^\top(t)\|\|P\|L\|\mathscr{X}(t)\| \leq 2\frac{v(\mathscr{X}_t)}{\alpha_1}\|P\|L.$$

Then, we arrive at

$$\begin{aligned}\tfrac{d}{dt}v(\mathscr{X}_t) \le & \begin{bmatrix} \mathscr{X}(t) \\ \mathscr{X}(t-\tau) \end{bmatrix}^\top \mathscr{M}(P,Q) \begin{bmatrix} \mathscr{X}(t) \\ \mathscr{X}(t-\tau) \end{bmatrix} \\ & -2\sigma\textstyle\int_{-\tau}^{0} \mathscr{X}^\top(t+\theta)e^{2\sigma\theta}Q\mathscr{X}(t+\theta)d\theta \\ & +2\tfrac{\gamma\|P\|}{\sqrt{\alpha_1}}\sqrt{v(\mathscr{X}_t)}+2\tfrac{\|P\|L}{\alpha_1}v(\mathscr{X}_t).\end{aligned}$$

Note that the functional can be rewritten as

$$v(\mathscr{X}_t)=\begin{bmatrix} \mathscr{X}(t) \\ \mathscr{X}(t-\tau) \end{bmatrix}^\top \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathscr{X}(t) \\ \mathscr{X}(t-\tau) \end{bmatrix} + \int_{-\tau}^{0} \mathscr{X}^\top(t+\theta)e^{2\sigma\theta}Q\mathscr{X}(t+\theta)d\theta.$$

So we can derive

$$\begin{aligned}&\tfrac{d}{dt}v(\mathscr{X}_t)+2\big(\sigma-\tfrac{L\|P\|}{\alpha_1}\big)v(\mathscr{X}_t)-2\tfrac{\gamma\|P\|}{\sqrt{\alpha_1}}\sqrt{v(\mathscr{X}_t)} \\ &\le \begin{bmatrix} \mathscr{X}(t) \\ \mathscr{X}(t-\tau) \end{bmatrix}^\top \Big(\mathscr{M}(P,Q)+2\sigma\mathscr{N}(P)\Big) \begin{bmatrix} \mathscr{X}(t) \\ \mathscr{X}(t-\tau) \end{bmatrix},\end{aligned} \tag{E.4}$$

with $\mathscr{M}(P,Q)$ and $\mathscr{N}(P)$ defined in Theorem 5.4.1.

We can conclude from the above arguments that if condition (7) holds, then

$$\frac{d}{dt}v(\mathscr{X}_t) \le -2\Big(\sigma-\frac{L\|P\|}{\alpha_1}\Big)v(\mathscr{X}_t)+2\frac{\gamma\|P\|}{\sqrt{\alpha_1}}\sqrt{v(\mathscr{X}_t)}. \tag{E.5}$$

If $L<\frac{\alpha_1}{\|P\|}\sigma=\frac{\lambda_{\min}(P)}{\lambda_{\max}(P)}\sigma$, it can be shown that

$$\begin{aligned}\|\mathscr{X}_t\| \le & \frac{\sqrt{\lambda_{\max}(P)+\tau\lambda_{\max}(Q)}}{\sqrt{\lambda_{\min}(P)}}e^{-\left(\sigma-\frac{L\lambda_{\max}(P)}{\lambda_{\min}(P)}\right)t}\|\Psi\|_s \\ & +\frac{\gamma\lambda_{\max}(P)}{\sigma\lambda_{\min}(P)-L\lambda_{\max}(P)}\left(1-e^{-\left(\sigma-\frac{L\lambda_{\max}(P)}{\lambda_{\min}(P)}\right)t}\right).\end{aligned} \tag{E.6}$$

It follows that if $L<\frac{\lambda_{\min}(P)}{\lambda_{\max}(P)}\sigma$, the system (5.12) is practically stable with $\mu$ and $T$ chosen as below:
for $\|\Psi\|_s \le \frac{\gamma\lambda_{\max}(P)}{(\sigma-\frac{L\lambda_{\max}(P)}{\lambda_{\min}(P)})\sqrt{\lambda_{\min}(P)[\lambda_{\max}(P)+\tau\lambda_{\max}(Q)]}}$,

$$\begin{aligned}&\mu=\beta(\gamma)=\frac{\gamma\lambda_{\max}(P)}{\sigma\lambda_{\min}(P)-L\lambda_{\max}(P)}, \\ &T=0,\end{aligned} \tag{E.7}$$

for $\|\Psi\|_s > \frac{\gamma\lambda_{\max}(P)}{(\sigma-\frac{L\lambda_{\max}(P)}{\lambda_{\min}(P)})\sqrt{\lambda_{\min}(P)[\lambda_{\max}(P)+\tau\lambda_{\max}(Q)]}}$,

$$\mu = \beta(\gamma) = \frac{k_\mu \gamma\lambda_{\max}(P)}{\sigma\lambda_{\min}(P) - L\lambda_{\max}(P)},$$

$$T = \frac{1}{\sigma - \frac{L\lambda_{\max}(P)}{\lambda_{\min}(P)}} \ln\left(\frac{\left(\sigma - \frac{L\lambda_{\max}(P)}{\lambda_{\min}(P)}\right)\sqrt{\lambda_{\min}(P)[\lambda_{\max}(P)+\tau\lambda_{\max}(Q)]}\|\Psi\|_s - \gamma\lambda_{\max}(P)}{\left(\sigma\lambda_{\min}(P) - L\lambda_{\max}(P)\right)\mu - \gamma\lambda_{\max}(P)}\right), \tag{E.8}$$

where $k_\mu$ satisfies the condition $1 < k_\mu < \frac{\left(\sigma-\frac{L\lambda_{\max}(P)}{\lambda_{\min}(P)}\right)\sqrt{\lambda_{\min}(P)[\lambda_{\max}(P)+\tau\lambda_{\max}(Q)]}\|\Psi\|_s}{\gamma\lambda_{\max}(P)}$, and $\|\Psi\|_s$ is the supremum norm $\|\Psi(\theta)\|_s := \sup_{\theta\in[0-\tau]}\|\Psi(\theta)\|$. □

# Appendix F

# Proof of Theorem 5.4.2

*Proof.* Consider the Lyapunov-Krasovskii functional

$$v(\mathscr{X}_t) = \mathscr{X}^\top(t)P\mathscr{X}(t) + \int_{t-\tau}^{t} \mathscr{X}^\top(\theta)e^{2\sigma(\theta-t)}Q\mathscr{X}(\theta)\mathrm{d}\theta + \tau\int_{-\tau}^{0}\int_{t+s}^{t} \dot{\mathscr{X}}^\top(\theta)e^{2\sigma(\theta-t)}Z\dot{\mathscr{X}}(\theta)\mathrm{d}\theta\mathrm{d}s. \tag{F.1}$$

It follows from (F.1) that

$$\alpha_1\|\mathscr{X}(t)\|^2 \leq v(\mathscr{X}_t) \leq \alpha_2\|\mathscr{X}_t\|_W^2, \tag{F.2}$$

where $\alpha_1 = \lambda_{\min}(P)$ and $\alpha_2 = \lambda_{\max}(P) + \tau\lambda_{\max}(Q) + \frac{\tau^3}{2}\lambda_{\max}(Z)$. The norm $\|\mathscr{X}_t\|_W$ is given by $\|\mathscr{X}_t\|_W = \max\{\|\mathscr{X}_t\|_s, \|\dot{\mathscr{X}}_t\|_s\}$ as in (2.7).

Differentiating (F.1) with respect to time $t$ yields

$$\begin{aligned}\dot{v}(\mathscr{X}_t) \leq\ & 2\mathscr{X}^\top(t)P(F(\mathscr{X}(t),\mathscr{X}_r(t)) - kA_0\mathscr{X}(t) + kA_1\mathscr{X}(t-\tau) + n(t,\tau,\Delta)) \\ & + \mathscr{X}^\top(t)Q\mathscr{X}(t) - e^{-2\sigma\tau}\mathscr{X}^\top(t-\tau)Q\mathscr{X}(t-\tau) \\ & - 2\sigma\int_{t-\tau}^{t}\mathscr{X}^\top(\theta)e^{2\sigma(\theta-t)}Q\mathscr{X}(\theta)\mathrm{d}\theta \\ & + \tau^2\dot{\mathscr{X}}^\top(t)Z\dot{\mathscr{X}}(t) - \tau e^{-2\sigma\tau}\int_{t-\tau}^{\tau}\dot{\mathscr{X}}^\top(\theta)Z\dot{\mathscr{X}}(\theta)\mathrm{d}\theta \\ & - 2\sigma\tau\int_{-\tau}^{0}\int_{t+s}^{s}\dot{\mathscr{X}}^\top(\theta)e^{2\sigma(\theta-t)}Z\dot{\mathscr{X}}(\theta)\mathrm{d}\theta\mathrm{d}s.\end{aligned}$$

Using Jensen's inequality, it follows that

$$\begin{aligned}\dot{v}(\mathscr{X}_t) \leq & 2\mathscr{X}^\top(t)P(F(\mathscr{X}(t),\mathscr{X}_r(t))-kA_0\mathscr{X}(t)+kA_1\mathscr{X}(t-\tau)+\Xi(t,\tau,\Delta))\\ & +2\sigma\mathscr{X}^\top(t)P\mathscr{X}(t)+\mathscr{X}^\top(t)Q\mathscr{X}(t)-e^{-2\sigma\tau}\mathscr{X}^\top(t-\tau)Q\mathscr{X}(t-\tau)\\ & +\tau^2\dot{\mathscr{X}}^\top(t)Z\dot{\mathscr{X}}(t)-e^{-2\sigma\tau}(\mathscr{X}(t)-\mathscr{X}(t-\tau))^\top Z(\mathscr{X}(t)-\mathscr{X}(t-\tau))\\ & -2\sigma v(\mathscr{X}_t).\end{aligned} \tag{F.3}$$

With the inequalities (5.27) in mind, we have, for any diagonal matrix $V>0$,

$$\begin{bmatrix}\mathscr{X}(t)\\ F(\mathscr{X}(t),\mathscr{X}_r(t))\end{bmatrix}^\top\begin{bmatrix}F_1V & -F_2V\\ * & V\end{bmatrix}\begin{bmatrix}\mathscr{X}(t)\\ F(\mathscr{X}(t),\mathscr{X}_r(t))\end{bmatrix}\leq 0.$$

On the other hand, we have from (5.28) that

$$\begin{bmatrix}\mathscr{X}(t)\\ \mathscr{X}(t-\tau)\\ F(\mathscr{X}(t),\mathscr{X}_r(t))\\ \Xi(t,\tau,\Delta)\end{bmatrix}^\top\left[\Theta+H_1PH_2+H_2^\top PH_1^\top+\tau^2H_2^\top ZH_2\right]\begin{bmatrix}\mathscr{X}(t)\\ \mathscr{X}(t-\tau)\\ F(\mathscr{X}(t),\mathscr{X}_r(t))\\ \Xi(t,\tau,\Delta)\end{bmatrix}<0. \tag{F.4}$$

Combining (F.3) with (F.4), we can conclude

$$\dot{v}(\mathscr{X}_t)<-2\sigma v(\mathscr{X}_t)+\Xi(t,\tau,\Delta)^\top W\Xi(t,\tau,\Delta).$$

Hence

$$\dot{v}(\mathscr{X}_t)<-2\sigma v(\mathscr{X}_t)+\lambda_{\max}(W)\gamma^2.$$

Using Grönwall's inequality, we arrive at

$$v(\mathscr{X}_t)<v(\Psi)e^{-2\sigma t}-\frac{\gamma^2}{2\sigma}\lambda_{\max}(W)(e^{-2\sigma t}-1). \tag{F.5}$$

Now, it follows from (F.2) that

$$\|\mathscr{X}_t\|\leq\sqrt{\frac{\gamma^2}{2\sigma\alpha_1}\lambda_{\max}(W)+e^{-2\sigma t}\left(\frac{\alpha_2}{\alpha_1}\|\Psi\|_W^2-\frac{\gamma^2}{2\sigma\alpha_1}\lambda_{\max}(W)\right)}. \tag{F.6}$$

Observe that for an initial condition $\Psi$ such that $\|\Psi\|_W\leq\gamma\sqrt{\frac{\lambda_{\max}(W)}{2\sigma\alpha_2}}$ we have

$$\|\mathscr{X}_t\|\leq\mu=\gamma\sqrt{\frac{\lambda_{\max}(W)}{2\sigma\alpha_1}},\ \ \forall t\geq 0. \tag{F.7}$$

For an initial condition $\Psi$ such that $\|\Psi\|_W > \gamma\sqrt{\frac{\lambda_{\max}(W)}{2\sigma\alpha_2}}$, we have

$$\|\mathscr{X}_t\| \leq \mu, \ \ \forall t \geq T(\mu,\Psi), \tag{F.8}$$

where $\mu = k_\mu \gamma\sqrt{\frac{\lambda_{\max}(W)}{2\sigma\alpha_1}}$, $1 < k_\mu < \frac{1}{\gamma}\sqrt{\frac{2\sigma\alpha_2}{\lambda_{\max}(W)}}\|\Psi\|_W$, and the time $T(\mu,\Psi)$ is obtained from the condition

$$0 < e^{-2\sigma t}\left(\frac{\alpha_2}{\alpha_1}\|\Psi\|_W^2 - \frac{\gamma^2}{2\sigma\alpha_1}\lambda_{\max}(W)\right) < \mu^2 - \frac{\gamma^2}{2\sigma\alpha_1}\lambda_{\max}(W), \tag{F.9}$$

thus,

$$T = \frac{1}{2\sigma}\ln\left(\frac{2\sigma\alpha_2\|\Psi\|_W^2 - \gamma^2\lambda_{\max}(W)}{2\sigma\alpha_1\mu^2 - \gamma^2\lambda_{\max}(W)}\right). \tag{F.10}$$

□

# Appendix G

# Proof of semi-passivity of System (5.5)

Equation (5.5) is of the form (5.1) with

$$\begin{cases} \dot{x}_i(t) = f(x_i(t)) + B_i u_i(t), \\ y_i(t) = C_i x_i(t), \end{cases} \tag{G.1}$$

where

$$x_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \end{bmatrix}, \; y_i = \begin{bmatrix} y_{i,1} \\ y_{i,2} \end{bmatrix}, \; u_i = \begin{bmatrix} u_{i,1} \\ u_{i,2} \end{bmatrix},$$
$$f(x_i) = \begin{bmatrix} 0.2x_{i,1} + x_{i,2} - 10x_{i,1}(x_{i,1}^2 + x_{i,2}^2) \\ -x_{i,1} + 0.2x_{i,2} - 10x_{i,2}(x_{i,1}^2 + x_{i,2}^2) \end{bmatrix}, \; B_i = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \; C_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Define the so-called storage function $V(x_i)$ as

$$V(x_i) = \frac{1}{2}(x_{i,1}^2 + x_{i,2}^2), \tag{G.2}$$

and the scalar function $S(x_i)$ as

$$S(x_i) = -0.4V(x_i) + 40V^2(x_i), \tag{G.3}$$

which is positive outside the ball $\mathscr{B} = \{x_i \in \mathbb{R}^2 | \, V(x_i) \leq 0.01\}$.

Then, we have

$$\begin{aligned}\dot{V}(x_i) &= x_{i,1}u_{i,2} + x_{i,2}u_{i,1} - \left[0.2(x_{i,1}^2 + x_{i,2}^2) + 10(x_{i,1}^2 + x_{i,2}^2)^2\right] \\ &= y^\top u + 0.4V(x_i) - 40V^2(x_i) \\ &\leq y^\top u - S(x_i).\end{aligned}$$

That is, the condition (2.2) is satisfied.

# Appendix H

# Solution of LMIs (5.28) in the example of Section 5.5

$$P = \begin{bmatrix} 19.6194 & 1.8757 & 4.7948 & 2.5742 \\ 1.8757 & 19.8934 & 0.9150 & 7.3896 \\ 4.7948 & 0.9150 & 17.8032 & 10.0527 \\ 2.5742 & 7.3896 & 10.0527 & 23.7725 \end{bmatrix},$$

$$Q = \begin{bmatrix} 11.1995 & 2.3924 & 1.2749 & 0.5651 \\ 2.3924 & 9.2019 & 0.5942 & 3.8496 \\ 1.2749 & 0.5942 & 8.6784 & 4.5230 \\ 0.5651 & 3.8496 & 4.5230 & 15.9833 \end{bmatrix},$$

$$Z = \begin{bmatrix} 0.2907 & 0.0351 & 0.2097 & 0.0774 \\ 0.0351 & 0.3373 & 0.1165 & 0.4194 \\ 0.2097 & 0.1165 & 0.2348 & 0.2100 \\ 0.0774 & 0.4194 & 0.2100 & 0.6446 \end{bmatrix},$$

$$V = \begin{bmatrix} 350.6328 & 0 & 0 & 0 \\ 0 & 386.4757 & 0 & 0 \\ 0 & 0 & 16.2167 & 0 \\ 0 & 0 & 0 & 148.6860 \end{bmatrix},$$

$$W = \begin{bmatrix} 354.5293 & 0.9549 & -135.9724 & -19.9116 \\ 0.9549 & 410.8246 & -24.8248 & -116.4935 \\ -135.9724 & -24.8248 & 391.4282 & 64.0708 \\ -19.9116 & -116.4935 & 64.0708 & 302.1782 \end{bmatrix}.$$

# Appendix I

# Model predictive control

This appendix briefly introduces model predictive control (MPC) which is used to achieve vehicle platooning in Chapter 7. MPC is one of the most successful and popular control methods. It is able to control a process with optimal control efforts while accounting for constraints.

The basic idea of MPC consists of two parts: 1) predict the future evolution of the controlled system over a finite time horizon using a so-called prediction model; 2) compute the optimal control input by minimizing an objective function while satisfying the system constraints. More precisely, MPC solves an optimization problem at each sampling instant and applies the first part of the computed optimal input to the system until the next sampling instant, and then repeats the procedure at the next sampling instant. The prediction model is a discrete time model obtained by discretizing the continuous time model of the controlled system.

Consider a controlled system of the following form

$$\begin{aligned} \dot{x}(t) &= f_c(x(t), u(t)), \\ y(t) &= g_c(x(t), u(t)), \end{aligned} \tag{I.1}$$

where $x(t) \in \mathbb{X} \subset \mathbb{R}^n$, $u(t) \in \mathbb{U} \subset \mathbb{R}^m$, and $y \in \mathbb{R}^m$ are the vector of states, inputs, and outputs, respectively. The sets $\mathbb{X}$ and $\mathbb{U}$ denotes the feasible states and inputs determined by the state and control input constraints, respectively. To employ MPC controllers, (I.1) is transformed into a discrete time model

$$\begin{aligned} x(k+1) &= f(x(k), u(k)), \\ y(k) &= g(x(k), u(k)), \end{aligned} \tag{I.2}$$

where $x(k) = x(kt_s)$, $y(k) = y(kt_s)$ and $u(k) = u(kt_s)$, $k \in \mathbb{Z}_{\geq 0}$, $t_s > 0$, are the values of continuous time signals $x(t)$, $y(t)$ and $u(t)$ at sampling instant $kt_s$ respectively, and $t_s$ represents the sampling period. The functions $f$ and $g$ can be determined based on the functions $f_c$ and $g_c$. It is straightforward that $g = g_c$. For example, consider a linear continuous time system represented by the following state space model

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t), \\ y(t) &= C_c x(t) + D_c u(t), \end{aligned} \tag{I.3}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times n}$, and $D \in \mathbb{R}^{m\times m}$. The corresponding discrete time model constructed by using the zero-order hold[1] method is given by

$$\begin{aligned} \dot{x}(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k) + Du(k), \end{aligned} \tag{I.5}$$

where $A = e^{A_c t_s}$, $B = \int_0^{t_s} e^{A_c \theta} \mathrm{d}\theta B_c$, $C = C_c$, $D = D_d$, see [125].

With the discrete model (I.2) ready, one can then design an MPC controller for the controlled system.

A general formulation of the MPC control problem is:

$$\min_{u(\cdot|k)} \left[ \sum_{j=0}^{N_1-1} J_1(y(j+k|k)) + \sum_{j=0}^{N_2-1} J_2(u(j+k|k)) \right] \tag{I.6}$$

subject to

$$y(k|k) = y(k), \tag{I.7}$$

$$x(k+j+1|k) = f(x(k+j|k), u(k+j|k)), \tag{I.8}$$

$$y(k+j|k) = g(x(k+j|k), u(k+j|k)), \tag{I.9}$$

$$x(k+j|k) \in \mathbb{X}, \tag{I.10}$$

$$u(k+j|k) \in \mathbb{U}. \tag{I.11}$$

Here, the vector $y(k) \in \mathbb{R}^m$ is the output measured at time instant $k$, and $x(k+j|k) \in \mathbb{R}^n$ is the prediction of state at the time instant $k+j$ calculated at the time instant $k$. The vector $u(k+j|k) \in \mathbb{R}^m$ is the control input at the time instant $k+j$ calculated at the

[1] Zero-order hold is an algorithm to reconstruct continuous time signals from discrete time signals:

$$x(t) = x(k), \quad kt_s \leq t < (k+1)t_s, \tag{I.4}$$

where $x(k)$, $k = 0, 1, 2, \ldots, n$ represents a discrete time signal, $x(t)$ represents the reconstructed continuous time signal, and $t_s$ is the sampling period, see [134].

time instant $k$. The prediction model given by (I.8) and (I.9) (*i.e.*, (I.2)) describes the dynamics of the system. The objective function is given by (I.6). The equations (I.10) and (I.11) represent the constants imposed on the states and control inputs, respectively. The numbers $N_1$ and $N_2$ are two key design parameters, called the prediction horizon and control horizon, respectively. The MPC controller becomes more aggressive when $N_1$ decreases or $N_2$ increases [135]. After solving this optimization problem, the first part of the computed optimal input $u^*(k|k)$ is applied. This procedure is repeated at each time instant.

Solving the optimization problem (I.6) – (I.11) may be challenging depending on the property of system dynamics, the choice of the objective function, the form of constraints, *etc.* Typically, convex quadratic objective functions are chosen. The advantage of adopting a convex quadratic objective function is that when subject to linear constraints, the optimization problem with such an objective function is also convex, and therefore is numerically tractable. This type of optimization problems can be efficiently solved by, for example, the active set method or interior point method [136, 137].

# Bibliography


[1] E. Steur, T. Oguchi, C. van Leeuwen, and H. Nijmeijer, "Partial synchronization in diffusively time-delay coupled oscillator networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 4, p. 043144, 2012.

[2] E. van Nunen, J. Verhaegh, E. Silvas, E. Semsar-Kazerooni, and N. van de Wouw, "Robust model predictive cooperative adaptive cruise control subject to v2v impairments," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Institute of Electrical and Electronics Engineers (IEEE), 2017.

[3] E. Steur, H. U. Ünal, C. van Leeuwen, and W. Michiels, "Characterization and computation of partial synchronization manifolds for diffusive delay-coupled systems," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 4, pp. 1874–1915, 2016.

[4] J. C. Zegers, E. Semsar-Kazerooni, M. Fusco, and J. Ploeg, "A multi-layer control approach to truck platooning: platoon cohesion subject to dynamical limitations," in *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, Insitute of Electrical and Electronics Engineers (IEEE), 2017.

[5] S. H. Strogatz, *Sync: the Emerging Science of Spontaneous Order*. Hyperion, 1st ed., 2003.

[6] J. Buck and E. Buck, "Synchronous fireflies," *Scientific American*, vol. 234, pp. 74–9, 82–5, 1976.

[7] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Introduction to Synchronization in Nature and Physics and Cooperative Control for Multi-Agent Systems on Graphs*, pp. 1–21. Springer-Verlag London, 2014.

[8] J. Ploeg, N. van de Wouw, and H. Nijmeijer, "$\mathscr{L}_p$ string stability of cascaded systems: application to vehicle platooning," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 786–793, 2014.

[9] G. V. Osipov, J. Kurths, and C. Zhou, *Synchronization in Oscillatory Networks*. Springer Series in Synergetics, Springer-Verlag Berlin Heidelberg, 2007.

[10] C. S. Peskin, *Mathematical Aspects of Heart Physiology*. Courant Institute of Mathematical Sciences, 1975.

[11] A. Sherman, J. Rinzel, and J. Keizer, "Emergence of organized bursting in clusters of pancreatic beta-cells by channel sharing," *Biophysical Journal*, vol. 54, no. 3, pp. 411–425, 1988.

[12] C. Huygens, *Oeuvres complètes de Christiaan Huygens: L'hhorloge à pendule de 1651 à 1666*, vol. 17. Swets & Zeitlinger, 1967.

[13] J. W. S. B. Rayleigh, *The Theory of Sound*, vol. 2. Macmillan, 1896.

[14] E. V. Appleton, "Automatic synchronization of triode oscillators," in *Proceeding of the Cambridge Philosophical Society*, vol. 21, p. 231, 1922.

[15] B. van der Pol, "VII. Forced oscillations in a circuit with non-linear resistance. (Reception with reactive triode)," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 3, no. 13, pp. 65–80, 1927.

[16] W. H. Eccles and J. H. Vincent, *British Patent Specification clxiii*, 1920.

[17] A. T. Winfree, "Biological rhythms and the behavior of populations of coupled oscillators," *Journal of theoretical biology*, vol. 16, no. 1, pp. 15–42, 1967.

[18] H. Fujisaka and T. Yamada, "Stability theory of synchronized motion in coupled-oscillator systems," *Progress of Theoretical Physics*, vol. 69, no. 1, pp. 32–47, 1983.

[19] A. S. Pikovskii, "Synchronization and stochastization of array of self-excited oscillators by external noise," *Radiophysics and Quantum Electronics*, vol. 27, no. 5, pp. 390–395, 1984.

[20] V. S. Afraimovich, N. N. Verichev, and M. I. Rabinovich, "Stochastic synchronization of oscillation in dissipative systems," *Radiophysics and Quantum Electronics*, vol. 29, no. 9, pp. 795–803, 1986.

[21] I. S. Aranson and N. F. Rul'kov, "Nontrivial structure of synchronization zones in multidimensional systems," *Physics Letters A*, vol. 139, no. 8, pp. 375–378, 1989.

[22] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Physical Review Letters*, vol. 64, pp. 821–824, 1990.

[23] L. M. Pecora and T. L. Carroll, "Master stability functions for synchronized coupled systems," *Physical Review Letters*, vol. 80, pp. 2109–2112, 1998.

[24] H. G. Schuster and P. Wagner, "Mutual entrainment of two limit cycle oscillators with time delayed coupling," *Progress of Theoretical Physics*, vol. 81, no. 5, pp. 939–945, 1989.

[25] W. Kinzel, A. Englert, G. Reents, M. Zigzag, and I. Kanter, "Synchronization of networks of chaotic units with time-delayed couplings," *Physical Review E*, vol. 79, no. 5, p. 056207, 2009.

[26] M. G. Earl and S. H. Strogatz, "Synchronization in oscillator networks with delayed coupling: A stability criterion," *Physical Review E*, vol. 67, no. 3, p. 036204, 2003.

[27] C. Li and G. Chen, "Synchronization in general complex dynamical networks with coupling delays," *Physica A: Statistical Mechanics and its Applications*, vol. 343, pp. 263–278, 2004.

[28] T. Oguchi, H. Nijmeijer, and T. Yamamoto, "Synchronization in networks of chaotic systems with time-delay coupling," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 18, no. 3, p. 037108, 2008.

[29] W. Michiels and H. Nijmeijer, "Synchronization of delay-coupled nonlinear oscillators: An approach based on the stability analysis of synchronized equilibria," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, no. 3, p. 033110, 2009.

[30] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Sciences*. Cambridge University Press, 1st ed., 2001.

[31] C. M. Gray, "Synchronous oscillations in neuronal systems: mechanisms and functions.," *Journal of Computational Neuroscience*, vol. 1, pp. 11–38, 1994.

[32] M. Bennett and R. Zukin, "Electrical coupling and neuronal synchronization in the mammalian brain," *Neuron*, vol. 41, no. 4, pp. 495–511, 2004.

[33] A. Pogromsky, G. Santoboni, and H. Nijmeijer, "Partial synchronization: from symmetry towards stability," *Physica D: Nonlinear Phenomena*, vol. 172, no. 1, pp. 65 – 87, 2002.

[34] I. Stewart, M. Golubitsky, and M. Pivato, "Symmetry groupoids and patterns of synchrony in coupled cell networks," *SIAM Journal on Applied Dynamical Systems*, vol. 2, no. 4, pp. 609–646, 2003.

[35] M. Golubitsky, I. Stewart, and A. Török, "Patterns of synchrony in coupled cell networks with multiple arrows," *SIAM Journal on Applied Dynamical Systems*, vol. 4, no. 1, pp. 78–100, 2005.

[36] V. N. Belykh, G. V. Osipov, V. S. Petrov, J. A. K. Suykens, and J. Vandewalle, "Cluster synchronization in oscillatory networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 18, no. 3, p. 037106, 2008.

[37] A. Y. Pogromsky, "A partial synchronization theorem," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 18, no. 3, p. 037107, 2008.

[38] T. Dahms, J. Lehnert, and E. Schöll, "Cluster and group synchronization in delay-coupled networks," *Physical Review E*, vol. 86, p. 016202, 2012.

[39] G. Orosz, "Decomposition of nonlinear delayed networks around cluster states with applications to neurodynamics," *SIAM Journal on Applied Dynamical Systems*, vol. 13, no. 4, pp. 1353–1386, 2014.

[40] K. Ryono and T. Oguchi, "Partial synchronization in networks of nonlinear systems with transmission delay couplings," *IFAC-PapersOnLine*, vol. 48, no. 18, pp. 77–82, 2015.

[41] K. Ryono and T. Oguchi, "Delay-independent synchronization and network topology of systems with transmission delay couplings," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 3, pp. 198–203, 2017.

[42] J. M. Montenbruck, M. Bürger, and F. Allgöwer, "Practical synchronization with diffusive couplings," *Automatica*, vol. 53, pp. 235–243, 2015.

[43] E. Panteley and A. Loria, "Synchronization and dynamic consensus of heterogeneous networked systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3758–3773, 2017.

[44] E. Steur, T. Vromen, and H. Nijmeijer, *Practical synchronization in networks of diffusively coupled non-identical systems*. D&C, vol. 2015.031, Eindhoven University of Technology, 2015.

[45] E. Schöll, G. Hiller, P. Hovel, and M. A. Dahlem, "Time-delayed feedback in neurosystems," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1891, pp. 1079–1096, 2009.

[46] T. Jüngling, H. Benner, H. Shirahama, and K. Fukushima, "Complete chaotic synchronization and exclusion of mutual pyragas control in two delay-coupled Rössler-type oscillators," *Physical Review E*, vol. 84, no. 5, p. 056208, 2011.

[47] H. U. Ünal and W. Michiels, "Prediction of partial synchronization in delay-coupled nonlinear oscillators, with application to Hindmarsh-Rose neurons," *Nonlinearity*, vol. 26, no. 12, p. 3101, 2013.

[48] T. J. Lewis, "Dynamics of spiking neurons connected by both inhibitory and electrical coupling," *Journal of Computational Neuroscience*, vol. 14, no. 3, pp. 283–309, 2003.

[49] C. C. Chow and N. Kopell, "Dynamics of spiking neurons with electrical coupling," *Neural Computation*, vol. 12, no. 7, pp. 1643–1678, 2000.

[50] S. Coombes, "Neuronal networks with gap junctions: a study of piecewise linear planar neuron models," *SIAM Journal on Applied Dynamical Systems*, vol. 7, no. 3, pp. 1101–1129, 2008.

[51] J. L. P. Velazquez, "Brief note: mathematics and the gap junctions: in-phase synchronization of identical neurons," *International Journal of Neuroscience*, vol. 113, no. 8, pp. 1095–1101, 2003.

[52] M. Pérez-Armendariz, C. Roy, D. Spray, and M. Bennett, "Biophysical properties of gap junctions between freshly dispersed pairs of mouse pancreatic beta cells," *Biophysical Journal*, vol. 59, no. 1, pp. 76–92, 1991.

[53] G. D. Vries, "Diffusively coupled bursters: effects of cell heterogeneity," *Bulletin of Mathematical Biology*, vol. 60, no. 6, pp. 1167–1200, 1998.

[54] S.-J. Chung and J.-J. E. Slotine, "Cooperative robot control and concurrent synchronization of lagrangian systems," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 686–700, 2009.

[55] A. Rodriguez-Angeles and H. Nijmeijer, "Coordination of two robot manipulators based on position measurements only," *International Journal of Control*, vol. 74, no. 13, pp. 1311–1323, 2001.

[56] H. Nijmeijer and A. Rodriguez-Angeles, *Synchronization of Mechanical Systems*. WORLD SCIENTIFIC, 2003.

[57] C. W. Wu and L. Chua, "Synchronization in an array of linearly coupled dynamical systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 42, no. 8, pp. 430–447, 1995.

[58] K. Cuomo, A. Oppenheim, and S. Strogatz, "Synchronization of Lorenz-based chaotic circuits with applications to communications," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 10, pp. 626–633, 1993.

[59] W. D. Blizard, "Multiset theory," *Notre Dame Journal of Formal Logic*, vol. 30, no. 1, pp. 36–66, 1989.

[60] H. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

[61] J. C. Willems, "Dissipative dynamical systems part I: General theory," *Archive for Rational Mechanics and Analysis*, vol. 45, no. 5, pp. 321–351, 1972.

[62] A. Y. Pogromsky, "Passivity based design of synchronizing systems," *International Journal of Bifurcation and Chaos*, vol. 08, no. 02, pp. 295–319, 1998.

[63] E. Steur and H. Nijmeijer, "Synchronization in networks of diffusively time-delay coupled (semi-)passive systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 6, pp. 1358–1371, 2011.

[64] J.-P. Richard, "Time-delay systems: an overview of some recent advances and open problems," *Automatica*, vol. 39, no. 10, pp. 1667–1694, 2003.

[65] W. Michiels and S.-I. Niculescu, *Stability and Stabilization of Time-Delay Systems: An Eigenvalue-Based Approach*. Society for Industrial and Applied Mathematics, 2007.

[66] E. Fridman, *Introduction to Time-Delay Systems: Analysis and Control*. Systems & Control: Foundations & Applications, Springer International Publishing, 2014.

[67] J. K. Hale and S. M. V. Lunel, *Introduction to Functional Differential Equations*, vol. 99 of *Applied Mathematical Sciences*. Springer-Verlag New York, 1993.

[68] K. Gu, V. L. Kharitonov, and J. Chen, *Stability of time-delay systems*. Birkhäuser Boston, 2003.

[69] E. Fridman, "Tutorial on Lyapunov-based methods for time-delay systems," *European Journal of Control*, vol. 20, no. 6, pp. 271–283, 2014.

[70] M. Barreau, A. Seuret, and F. Gouaisbaut, "Wirtinger-based exponential stability for time-delay systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11984–11989, 2017.

[71] E. D. Andersen and K. D. Andersen, "The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm," in *High Performance Optimization*, pp. 197–232, Springer-Verlag Boston, 2000.

[72] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[73] K. C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3 — a MATLAB software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 545–581, 1999.

[74] P. Gahinet, A. Nemirovski, A. Laub, and M. Chilali, "LMI control toolbox user's guide," *The Math Works Inc*, 1995.

[75] B. Bollobas, *Modern Graph Theory*. Springer-Verlag New York, 1998.

[76] A. Gibbons, *Algorithmic Graph Theory*. Cambridge University Press, 1985.

[77] L. Su, W. Michiels, E. Steur, and H. Nijmeijer, "Computing partial synchronization manifolds of delay-coupled systems," in *Proceedings of the 9th European Nonlinear Dynamics Conference*, 2017.

[78] L. Su, W. Michiels, E. Steur, and H. Nijmeijer, "A method for computation and analysis of partial synchronization manifolds of delay coupled systems," in *Accounting for Constraints in Delay Systems* (G. Valmorbida, W. Michiels, and P. Pepe, eds.), Advances in Delays and Dynamics, Springer-Verlag Berlin Heidelberg, 2018. In press, a preprint available http://twr.cs.kuleuven.be/research/software/delay-control/chaptermanifolds.pdf.

[79] T. Oguchi, M. Suzuki, and D. Yanagi, "Delay-independent partial synchronization in networks of non-identical nonlinear systems with transmission delay coupling," in *Proceedings of the 9th European Nonlinear Dynamics Conference*, 2017.

[80] H. Nijmeijer and A. van der Schaft, *Nonlinear dynamical control systems*. Springer-Verlag New York, 1990.

[81] H. Nijmeijer, "Invertibility of affine nonlinear control systems: a geometric approach," *Systems & Control Letters*, vol. 2, no. 3, pp. 163–168, 1982.

[82] E. T. Bell, "Exponential numbers," *The American Mathematical Monthly*, vol. 41, no. 7, pp. 411–419, 1934.

[83] H. S. Wilf, *Generatingfunctionology*. Academic Press, 2nd ed., 1994.

[84] L. Su, Y. Wei, W. Michiels, E. Steur, and H. Nijmeijer, "A Lyapunov approach to stability analysis of partial synchronization in delay-coupled networks," *IFAC-PapersOnLine*, vol. 51, no. 33, pp. 198–204, 2018.

[85] F. Wu and K. M. Grigoriadis, "LPV systems with parameter-varying time delays: analysis and control," *Automatica*, vol. 37, no. 2, pp. 221–229, 2001.

[86] X. Zhang, P. Tsiotras, and C. Knospe, "Stability analysis of LPV time-delayed systems," *International Journal of Control*, vol. 75, no. 7, pp. 538–558, 2002.

[87] C. Briat, "Linear parameter-varying and time-delay systems," *Analysis, observation, filtering & control*, vol. 3, 2014.

[88] C. Briat, *Robust Control and Observation of LPV Time-Delay Systems*. Theses, Institut National Polytechnique de Grenoble - INPG, 2008.

[89] K. Li, S. Guan, X. Gong, and C.-H. Lai, "Synchronization stability of general complex dynamical networks with time-varying delays," *Physics Letters A*, vol. 372, no. 48, pp. 7133–7139, 2008.

[90] Y. He, M. Wu, J.-H. She, and G.-P. Liu, "Parameter-dependent Lyapunov functional for stability of time-delay systems with polytopic-type uncertainties," *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 828–832, 2004.

[91] R. C. L. F. Oliveira and P. L. D. Peres, "Parameter-dependent LMIs in robust analysis: characterization of homogeneous polynomially parameter-dependent solutions via LMI relaxations," *IEEE Transactions on Automatic Control*, vol. 52, no. 7, pp. 1334–1340, 2007.

[92] J. L. Hindmarsh and R. M. Rose, "A model of neuronal bursting using three coupled first order differential equations," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 221, no. 1222, pp. 87–102, 1984.

[93] E. Steur, I. Tyukin, and H. Nijmeijer, "Semi-passivity and synchronization of diffusively coupled neuronal oscillators," *Physica D: Nonlinear Phenomena*, vol. 238, no. 21, pp. 2119–2128, 2009.

[94] L. Su, Y. Wei, W. Michiels, E. Steur, and H. Nijmeijer, "Robust partial synchronization of delay-coupled networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 1, p. 013126, 2020.

[95] P. DeLellis, M. di Bernardo, and G. Russo, "On QUAD, Lipschitz, and contracting vector fields for consensus and synchronization of networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 3, pp. 576–583, 2011.

[96] T. G. M. Vromen, E. Steur, and H. Nijmeijer, "Training a network of electronic neurons for control of a mobile robot," *International Journal of Bifurcation and Chaos*, vol. 26, no. 12, p. 1650196, 2016.

[97] R. Villafuerte, S. Mondié, and A. Poznyak, "Practical stability of time-delay systems: LMI's approach," *European Journal of Control*, vol. 17, no. 2, pp. 127–138, 2011.

[98] J. Löfberg, "YALMIP : a toolbox for modeling and optimization in MATLAB," in *2004 IEEE International Conference on Robotics and Automation*, pp. 284–289, Insitute of Electrical and Electronics Engineers (IEEE), 2004.

[99] N. Axmacher, F. Mormann, G. Fernández, C. E. Elger, and J. Fell, "Memory formation by neuronal synchronization," *Brain Research Reviews*, vol. 52, no. 1, pp. 170–182, 2006.

[100] M. Bastiaansen and P. Hagoort, "Oscillatory neuronal dynamics during language comprehension," in *Progress in Brain Research* (C. Neuper and W. Klimesch, eds.), vol. 159 of *Event-Related Dynamics of Brain Oscillations*, pp. 179–196, Elsevier BV, 2006.

[101] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[102] C. Koch, *Biophysics of Computation: Information Processing In Single Neurons*. Oxford University Press, 2004.

[103] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, A. J. Hudspeth, and S. Mack, *Principles of Neural Science*, vol. 4. McGraw-hill New York, 2000.

[104] P. J. Neefs, E. Steur, and H. Nijmeijer, "Network complexity and synchronous behavior — an experimental approach," *International Journal of Neural Systems*, vol. 20, no. 03, pp. 233–247, 2010.

[105] E. Steur, C. Murguia, R. H. B. Fey, and H. Nijmeijer, "Synchronization and partial synchronization experiments with networks of time-delay coupled Hindmarsh–Rose neurons," *International Journal of Bifurcation and Chaos*, vol. 26, no. 07, p. 1650111, 2016.

[106] I. Castanedo-Guerra, E. Steur, and H. Nijmeijer, "Synchronization of "light-sensitive" Hindmarsh–Rose neurons," *Communications in Nonlinear Science and Numerical Simulation*, vol. 57, pp. 322–330, 2018.

[107] P. Kirkwood, "On the use and interpretation of cross-correlation measurements in the mammalian central nervous system," *Journal of Neuroscience Methods*, vol. 1, no. 2, pp. 107–132, 1979.

[108] T. J. Nakamura, K. Fujimura, S. Ebihara, and K. Shinohara, "Light response of the neuronal firing activity in the suprachiasmatic nucleus of mice," *Neuroscience Letters*, vol. 371, no. 2, pp. 244–248, 2004.

[109] R. Kianfar, P. Falcone, and J. Fredriksson, "A receding horizon approach to string stable cooperative adaptive cruise control," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Insitute of Electrical and Electronics Engineers IEEE, 2011.

[110] A. Ghasemi, R. Kazemi, and S. Azadi, "Stable decentralized control of a platoon of vehicles with heterogeneous information feedback," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4299–4308, 2013.

[111] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 899–910, 2017.

[112] S. Shladover, "Recent international activity in cooperative vehicle–highway automation systems," tech. rep., United States. Federal Highway Administration. Office of Corporate Research, 2012.

[113] A. Bose and P. Ioannou, "Mixed manual/semi-automated traffic: a macroscopic analysis," *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 6, pp. 439–462, 2003.

[114] M. P. Lammert, A. Duran, J. Diez, K. Burton, and A. Nicholson, "Effect of platooning on fuel consumption of class 8 vehicles vver a range of speeds, following distances, and mass," *SAE International Journal of Commercial Vehicles*, vol. 7, no. 2, pp. 626–639, 2014.

[115] F. Browand, J. McArthur, and C. Radovich, "Fuel saving achieved in the field test of two tandem trucks," *UC Berkeley: California Partners for Advanced Transportation Technology*, 2004.

[116] P. Barooah and J. Hespanha, "Error amplification and disturbance propagation in vehicle strings with decentralized linear control," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Insitute of Electrical and Electronics Engineers IEEE, 2015.

[117] J. Ploeg, D. P. Shukla, N. van de Wouw, and H. Nijmeijer, "Controller synthesis for string stability of vehicle platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 854–865, 2014.

[118] D. Dileep, M. Fusco, J. Verhaegh, L. Hetel, J.-P. Richard, and W. Michiels, "Achieving an $\mathscr{L}_2$ string stable one vehicle look-ahead platoon with heterogeneity in time-delays," in *2019 18th European Control Conference (ECC)*, pp. 1220–1226, IEEE, 2019.

[119] D. Dileep, R. Van Parys, G. Pipeleers, L. Hetel, J.-P. Richard, and W. Michiels, "Design of robust decentralised controllers for mimo plants with delays through network structure exploitation," *International Journal of Control*, vol. 93, no. 10, pp. 2275–2289, 2020.

[120] M. R. I. Nieuwenhuijze, T. van Keulen, S. Oncu, B. Bonsen, and H. Nijmeijer, "Cooperative driving with a heavy-duty truck in mixed traffic: experimental results," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1026–1032, 2012.

[121] M. R. I. Nieuwenhuijze, *String stability analysis of bidirectional adaptive cruise control*. D&C, vol. 2010.068, Eindhoven University of Technology, 2010.

[122] S. Li, K. Li, R. Rajamani, and J. Wang, "Model predictive multi-objective vehicular adaptive cruise control," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 556–566, 2011.

[123] T. Stanger and L. del Re, "A model predictive cooperative adaptive cruise control approach," in *2013 American Control Conference*, Insitute of Electrical and Electronics Engineers (IEEE), 2013.

[124] T. Takahama and D. Akasaka, "Model predictive control approach to design practical adaptive cruise control for traffic jam," *International Journal of Automotive Engineering*, vol. 9, no. 3, pp. 99–104, 2018.

[125] K. J. Åström and B. Wittenmark, *Computer Controlled Systems: Theory and Design*, pp. 48–52. Prentice-Hall, 1990.

[126] E. van Nunen, J. Reinders, E. Semsar-Kazerooni, and N. van de Wouw, "String stable model predictive cooperative adaptive cruise control for heterogeneous platoons," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 186–196, 2019.

[127] X. Zhang, Q. Han, X. Ge, D. Ding, L. Ding, D. Yue, and C. Peng, "Networked control systems: a survey of trends and techniques," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 1–17, 2020.

[128] T. H. Lee and J. H. Park, "Improved criteria for sampled-data synchronization of chaotic Lur'e systems using two new approaches," *Nonlinear Analysis: Hybrid Systems*, vol. 24, pp. 132–145, 2017.

[129] K. Sakai and T. Oguchi, "Synchronization of coupled nonlinear systems with bidirectional sampled-data couplings," *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 634–639, 2019.

[130] B. Shen, Z. Wang, and X. Liu, "Sampled-data synchronization control of dynamical networks with stochastic sampling," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2644–2650, 2012.

[131] J. Thomas, E. Stuer, C. Fiter, L. Hetel, and N. van de Wouw, "Exponential synchronization of nonlinear oscillators under sampled-data coupling," in

*Proceedings of the IEEE 59th Conference on Decision and Control (CDC)*, 2020.

[132] C. S. Colwell, "Bridging the gap: coupling single-cell oscillators in the suprachiasmatic nucleus," *Nature Neuroscience*, vol. 8, no. 1, pp. 10–12, 2005.

[133] M. A. Long, M. J. Jutras, B. W. Connors, and R. D. Burwell, "Electrical synapses coordinate activity in the suprachiasmatic nucleus," *Nature Neuroscience*, vol. 8, no. 1, pp. 61–66, 2005.

[134] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*. Prentice-Hall signal processing series, Prentice Hall, 2nd ed., 1997.

[135] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, and F. J. Doyle III, *Process Dynamics and Control*. John Wiley & Sons, 2010.

[136] S. J. Wright, "Applying new optimization algorithms to more predictive control," in *Proceedings of Chemical Process Control V*, pp. 147–155, 1997.

[137] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of interior-point methods to model predictive control," *Journal of Optimization Theory and Applications*, vol. 99, no. 3, pp. 723–757, 1998.

# Curriculum Vitae

Libo Su was born on January 12, 1990 in Zhejiang, China. He obtained his Bachelor's degree at Zhejiang University, China, in 2011, after which he started his study in Mechanical Engineering at Eindhoven University of Technology, Eindhoven, the Netherlands. In October 2013, he received his Master's degree from Eindhoven University of Technology. In 2014 - 2015, he worked as a system engineering at Applied Micro Electronics "AME" B.V., Eindhoven. In 2015 - 2016, he worked as a process engineer at Syncreon, Waalwijk.

In December 2016, he started as an early stage researcher in the Marie Curie project UCoCoS (Understanding and Controlling of Complex Systems) towards a joint PhD degree from KU Leuven and Eindhoven University of Technology. He was hosted by the Numerical Analysis and Applied Mathematics group at the department of Computer Science of KU Leuven and the Dynamics and Control group at the department of Mechanical Engineering of Eindhoven University of Technology. The main topic of his research was the development of theory and tools to understand and analyze partial synchronization in networks of delay-coupled systems. In April - May 2018, he spent a month in the Control and Scientific Computing group of the CNRS lab CRIStAL at Centrale Lille in the framework of an international secondment. In May - August 2019, he carried out another secondment in the Integrated Vehicle Safety group at TNO, Helmond, the Netherlands. The main results of his PhD research are presented in this thesis.

# List of publications

## Journal

- L. Su, Y. Wei, W. Michiels, E. Steur, and H. Nijmeijer, "Robust partial synchronization of delay-coupled networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 1, p. 013126, 2020.

## Book chapter

- L. Su, W. Michiels, E. Steur, and H. Nijmeijer, "A method for computation and analysis of partial synchronization manifolds of delay coupled systems," in *Accounting for Constraints in Delay Systems* (G. Valmorbida, W. Michiels and P. Pepe, eds.), Advances in Delays and Dynamics, Springer-Verlag Berlin Heidelberg, 2018. In press, preprint available on http://twr.cs.kuleuven.be/research/software/delay-control/chaptermanifolds.pdf.

## Proceedings

### Published

- L. Su, W. Michiels, E. Steur, and H. Nijmeijer, "Computing partial synchronization manifolds of delay-coupled systems," in *Proceedings of the 9th European Nonlinear Dynamics Conference*, 2017.
- L. Su, Y. Wei, W. Michiels, E. Steur, and H. Nijmeijer, "A Lyapunov approach to stability analysis of partial synchronization in delay-coupled networks," *IFAC-PapersOnLine*, vol. 51, no. 33, pp. 198–204, 2018.



FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
NUMERICAL ANALYSIS AND APPLIED MATHEMATICS
Celestijnenlaan 200A box 2402
B-3001 Leuven