Nonlinear Multi-Agent Control with Application to Networked Systems

Lorenzo Ricciardi Celsi

NNT : 2018SACLS017

Thèse de doctorat

Commande non linéaire multi-agents : applications aux systèmes en réseau

Thèse de doctorat de l'Università degli Studi di Roma "La Sapienza" et de l'Université Paris-Saclay, préparée à l'Université Paris-Sud, au sein du Dipartimento di Ingegneria Informatica, Automatica e Gestionale (DIAG) et du Laboratoire des Signaux et Systèmes (L2S), CentraleSupélec

École doctorale n°580 ED STIC
Sciences et technologies de l'information et de la communication
Spécialité de doctorat : Automatique

Thèse présentée et soutenue à Rome, le 22 janvier 2018, par

M. Lorenzo Ricciardi Celsi

Composition du Jury :

M. Didier Dumur Professeur, CentraleSupélec (L2S – Pôle Systèmes	Président
M. Saverio Mascolo Professeur, Politecnico di Bari (DEI)	Rapporteur
M. Malek Ghanes Professeur des Universités, Centrale Nantes (LS2N)	Rapporteur
M. Didier Dumur Professeur, CentraleSupélec (L2S – Pôle Systèmes)	Examinateur
M. Vincenzo Suraci Professeur, Università eCampus (SMARTEST)	Examinateur
Mme Marie-Dorothée Normand-Cyrot Directeur de recherche, CNRS – Université Paris-Saclay	Directrice de thèse
M. Salvatore Monaco Professeur, Università degli Studi di Roma "La Sapienza" (DIAG)	Co-Directeur de thèse
M. Francesco Delli Priscoli Professeur, Università degli Studi di Roma "La Sapienza" (DIAG)	Invité

SAPIENZA
UNIVERSITÀ DI ROMA

UNIVERSITÉ FRANCO ITALIENNE | UNIVERSITÀ ITALO FRANCESE

Facoltà di Ingegneria dell'Informazione,
Informatica e Statistica

PhD Program in Automatica, Bioengineering and Operation Research in Co-Tutelle with Ecole Doctorale STIC de l'Université Paris-Saclay

Curriculum in Automatica

Nonlinear Multi-Agent Control with Application to Networked Systems

Advisors

Chiar.mo Prof. Salvatore Monaco

Chiar.ma Prof.ssa Marie-Dorothée Normand-Cyrot

Candidate

Ing. Lorenzo Ricciardi Celsi

DIAG – Dipartimento di Ingegneria Informatica, Automatica e Gestionale Antonio Ruberti

A.A. 2016-2017

Le savant digne de ce nom, le géomètre surtout,
éprouve en face de son oeuvre la même impression que l'artiste ;
sa jouissance est aussi grande et de même nature.
- Jules Henri Poincaré

In order to arrive there,
to arrive where you are, to get from where you are not,
you must go by a way wherein there is no ecstasy.
In order to arrive at what you do not know
you must go by the way which is the way of ignorance.
In order to possess what you do not possess
you must go by the way of dispossession.
In order to arrive at what you are not
you must go through the way in which you are not.
And what you do not know is the only thing you know
and what you own is what you do not own
and where you are is where you are not.
- Thomas Stearns Eliot

To my Grandma

Declaration of Authorship

Date: January 10th, 2018.

I, Lorenzo Ricciardi Celsi, declare that this PhD thesis, titled "Nonlinear Multi-Agent Control with Application to Networked Systems" ("Commande non linéaire multi-agents : application aux systèmes en réseau") and the work presented in it are my own. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given and, with the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Acknowledgements

I would like to express my deepest gratitude to my Italian advisor, Prof. Salvatore Monaco, for his untiring encouragements and guidance and for giving me the opportunity to study such an interesting and stimulating topic. His immense knowledge and priceless suggestions gave me the strength and positive attitude to face and overcome various difficulties. I would like to extend my sincerest thanks to my French advisor, Prof. Marie-Dorothée Normand-Cyrot, for welcoming me at Supélec, for introducing me to the French academic environment (which is so different from the Italian one), for giving me the wonderful opportunity of getting the *double diplôme de doctorat*, for encouraging me and for wisely guiding the investigation of the literature and my research in general. Also, I wish to express my deepest gratitude to Prof. Francesco Delli Priscoli for his invaluable teachings and insightful comments and advice, as well as for giving me the opportunity to collaborate in several research projects with an international outlook.

I am sincerely indebted and grateful to Prof. Antonio Pietrabissa, Prof. Vincenzo Suraci and Dr. Alessandro Di Giorgio for helping me find and select the sources for my research and for withstanding my hour-long questions. I am so grateful to my colleague Mattia Mattioni, for helping me with the *procédure d'inscription* of the *École Doctorale STIC*, especially when I still could not speak French very well. I also wish to thank Dr. Silvia Canale and Dr. Martina Panfili for their help and suggestions, as well as my colleagues Raffaello Bonghi, Federico Cimorelli, Francesco Liberati, Raffaele Gambuti, Alessandro Giuseppi, Federico Lisi and Giovanni Mattei for the great time we had together, not just at university but also during our missions all over Europe.

I would like to express my thanks to the Department of Computer, Control and Management Engineering (DIAG) and to the Università Italo Francese (UFI) for providing full financial support for my research activities. I wish to thank Ing. Manlio Proia and the CRAT and ARES2T teams for the constant support.

Some of the work appearing in this thesis was also partially supported

by: the MIUR-PON PLATINO project (`www.progettoplatino.it`), funded by the Italian Ministry of Education, Research and University under Grant Agreement no. PON01_01007; the FP7 T-NOVA project (`www.t-nova.eu`) funded by the European Commission under Grant Agreement no. 619520; the H2020 BONVOYAGE project (`bonvoyage2020.eu`), funded by European Commission under Grant Agreement no. 635867; and the H2020 ATENA project (`www.atena-h2020.eu`), funded by the European Commission under Grant Agreement no. 700581. So, I wish to express my thanks to all the partners in such projects for the constant and constructive cooperation and helpful suggestions.

Lastly, I would like to give my special thanks to my parents, my sister and my grandmother for their unconditional and patient love. If I had not been endowed with all their support and consideration, I certainly would not have reached this wonderful goal.

Résumé

La science des réseaux est aujourd'hui un puissant paradigme conceptuel en ingénierie. Une attention sans précédent est actuellement consacrée à la théorie des réseaux et à ses applications par la communauté scientifique. D'une part, il est devenu vital qu'on comprend mieux le rôle des interactions inter-élémentaires dans la fonctionnalité collective des systèmes multicouches dans les sciences biologiques et, d'autre part, les progrès technologiques ont considérablement amélioré notre capacité de synthétiser des systèmes d'ingénierie en réseau (systèmes á plusieurs véhicules et pelotons, vol en formation, équipes de robots, réseaux de capteurs sans fil, réseaux d'énergie, réseaux sociaux) qui ressemblent à leurs homologues naturels en termes de complexité opérationnelle. Habituellement, les agents des systémes en réseau doivent fonctionner de concert les uns avec les autres afin d'atteindre des objectifs du système avec des ressources de calcul limitées, des capacités de détection limitées et des communications locales.

Ceci a suscité chez le doctorant un intérêt significatif pour le cadre méthodologique des *systèmes dynamiques multi-agents en réseau*. En particulier, l'étude de la littérature récente a amené le doctorant à découvrir que, du point de vue méthodologique, il existe un dénominateur commun qui jette les bases de presque tous les résultats novateurs liés aux systémes dynamiques multi-agents : en effet, ces résultants sont fondés sur :

- la théorie algébrique des graphes et
- la théorie du consensus.

Dans cet esprit, en accord avec ses superviseurs, le doctorant a orienté ses recherches vers le développement de résultats méthodologiques innovants en termes de réalisation de comportements de multi-consensus dans les systèmes multi-agents.

Cependant, ces aspects ont été étudiés non seulement du point de vue *méthodologique*, mais aussi du point de vue *applicatif*.

En effet, au cours des trois dernières années, le doctorant a dédié beaucoup de temps à résoudre des problèmes pratiques pour certaines applications

technologiques réelles dans le domaine des réseaux de télécommunication et d'énergie, dans le cadre de projets de recherche italiens et européens, en utilisant les méthodes théoriques de contrôle mentionnées ci-dessus. À ce titre, en tant que consultant interne pour le CRAT (*Consorzio per la Ricerca nell'Automatica e nelle Telecomunicazioni*), le doctorant a participé aux projets suivants, en s'engageant non seulement sur les phases et processus de gestion de projet, mais surtout sur la partie technique de ces projets et les défis connexes :

- projet MIUR-PON PLATINO (*Platform for Innovative Services in Future Internet*, juillet 2012 – juin 2015);
- projet FP7 T-NOVA (*Network Functions as-a-service over Virtualised Infrastructures*, janvier 2014 – décembre 2016);
- projet H2020 BONVOYAGE project (*From Bilbao to Oslo, intermodal mobility solutions, interfaces and applications for people and goods, supported by an innovative communication network*, mai 2015 – avril 2018);
- projet H2020 ATENA (*Advanced tools to assess and mitigate the criticality of ICT components and their dependencies over critical infrastructures*, mai 2016 – avril 2019).

En particulier, le projet PLATINO portait sur la conception et le développement d'une plate-forme de service capable de fournir services et contenus hétérogènes, par le biais de réseaux différents, suffisamment souple pour gérer une large gamme de terminaux, sensible au contexte, personnalisable sur la base des préférences de l'utilisateur, et enfin capable de garantir la Qualité de Service et de mesurer et maximiser la Qualité d'Expérience de l'utilisateur, dans le respect du paradigme de l'Internet du Futur. Ce projet a effectivement bénéficié des fonctions de contrôle de la la Qualité de l'Expérience capables de sélectionner dynamiquement les Classes de Service les plus appropriées grâce à un algorithme heuristique innovant fondé sur l'apprentissage par renforcement à plusieurs agents et conçu de manière à contourner certains problèmes d'implémentation pratique et à obtenir des performances satisfaisants, même en présence de plusieurs centaines d'agents. Le projet T-NOVA, au contraire, portait sur la conception et la mise en œuvre d'une plate-forme de gestion/orchestration pour la fourniture automatisée, la configuration, la surveillance et l'optimisation de *Network Functions-as-a-Service* sur réseaux virtualisés et/ou infrastructures informatiques en améliorant les plate-formes déjà existantes de *Software Defined Networking* (SDN) : ce projet a béneficié

d'un algorithme distribué et non-coopératif d'équilibrage de charge à temps discret fondé sur la théorie des jeux à champ moyen, dans le but d'équilibrer dynamiquement les demandes des commutateurs parmi les contrôleurs SDN pour éviter la congestion dans le trafic de contrôle. Les résultats de recherche obtenus dans le cadre des activités des projets PLATINO et T-NOVA ont donné lieu à quatre communications dans des congrès internationaux et à trois articles de journal (dont deux sont toujours en cours d'examen).

Le doctorant a également participé au projet BONVOYAGE, portant sur la conception, le développement et le test d'une plate-forme optimisant le transport porte-à -porte multimodal de passagers et de marchandises. Cette plate-forme intègre les services d'information, de planification et de billetterie en analysant automatiquement : les données collectées en temps non réel à partir de bases de données hétérogènes (sur les réseaux routiers, ferroviaires et urbains), les données mesurées en temps réel (prévisions de trafic et météo), profils et commentaires des utilisateurs. Le doctorant a ainsi développé un algorithme pour résoudre le problème du covoiturage avec l'agrégation automatisée de passagers dans le cadre de la planification multimodale des déplacements en s'assurant que la solution proposée contourne raisonnablement la complexité de calcul des scénarios considérés. Ce travail a conduit aux résultats présentés dans deux communications dans des congrès internationaux. Pourtant, un tel sujet de recherche n'est pas détaillé dans cette thèse de doctorat, car il est au-delà de la portée des systèmes dynamiques multi-agents et de leurs applications spécifiques.

Le projet ATENA, toujours en cours, concerne plutôt le développement d'une suite de composants TIC (Technologies de l'Information et de la Communication) en réseau pour la détection et la réaction aux événements indésirables dans les systèmes cyber-physiques et, en particulier, la conception d'un système de protection fondé sur le placement de systèmes de stockage d'énergie contre des attaques dynamiques de modification de charge en boucle fermée, supportant ainsi avec succès les services de régulation de fréquence dans les réseaux de transmission de puissance. Jusqu'à présent, les résultats de recherche connexes ont mené à deux communications dans des congrès internationaux.

Il ressort clairement de la description ci-dessus que le cadre méthodologique des systèmes dynamiques multi-agents en réseau représente un terrain d'entente qui se trouve sous les projets T-NOVA, PLATINO et ATENA. Cela a conduit le doctorant à enquêter sur la littérature récente et à rechercher des avancées significatives au-delà de l'état de l'art. Les activités de recherche et les résultats correspondants ont été réalisés dans le cadre du *Progetto di Ateneo* (*Sapienza, Université de Rome*) au sujet des systèmes non linéaires à temps discret et à données échantillonnées, conduisant à quatre communica-

tions dans des congrès internationaux et à un article de journal (en cours de préparation).

En particulier, le *Progetto di Ateneo* tourne autour de l'étude des modèles mathématiques décrivant le comportement des systèmes dynamiques multi-agents, avec un accent particulier sur :

- la coordination distribuée des systèmes multi-agents;
- le multi-consensus et les *almost equitable partitions* des graphes;
- la conception de lois de contrôle à données échantillonnées avec des applications aérospatiales – identifié comme un suivi des travaux effectués par le doctorant dans sa thèse de master.

Les travaux reportés ci-dessus représentent toute la production scientifique dont le doctorant a été coauteur au cours des trois années du programme de doctorat.

L'extension internationale de la portée culturelle, scientifique et professionnelle des activités menées par le doctorant n'est dû pas seulement à la participation active aux projets de recherche européens mentionnés ci-dessus, mais aussi, notamment, à la *cotutelle* avec l'*Ecole Doctorale de Sciences et Technologies de l'information et de la Communication* (ED STIC) à *Centrale Supélec, Université Paris-Saclay* et au financement reçu par l'*Université Franco-Italienne* (UFI) après avoir remporté le concours *Vinci 2016*.

La thèse de doctorat est divisé en quatre chapitres, comme suit :

- Chapitre 1: Apprentissage par renforcement à plusieurs agents, équilibrage de charge à temps discret basé sur Lyapunov et optimisation du placement des systèmes de stockage d'énergie par rapport aux attaques dynamiques de modification de charge.
- Chapitre 2: État de l'art sur la coordination distribué des systèmes multi-agents.
- Chapitre 3: Partitions des graphes et multi-consensus.
- Chapitre 4: Conception de lois de contrôle à données échantillonnées pour des applications aérospatiales.

Ainsi, la thèse est caracterisée par une première partie (composé des chapitre 1) dedié à la présentation et à la discussion des applications les plus pertinentes des systèmes de contrôle multi-agents en réseau dans le cadre de projets T-NOVA, ATENA et PLATINO. Ensuite, la deuxième

partie (composée des chapitres 2, 3 et 4) présente le contexte théorique sous-jacente par rapport à la coordination des systèmes multi-agents distribués et les résultats innovants obtenus en ce qui concerne le multi-consensus et la conception de lois de contrôle à données échantillonnées.

Abstract

La scienza delle reti si è recentemente imposta in ingegneria come un paradigma concettuale di peso significativo. Un'attenzione senza precedenti è attualmente rivolta dalla comunità scientifica a questa discriplina e alle sue applicazioni poiché, da un lato, è di importanza fondamentale nelle scienze biologiche l'acquisizione di una comprensione profonda del ruolo che nel comportamento collettivo di sistemi multi-strato svolgono le interazioni tra i singoli componenti e, dall'altro, i recenti sviluppi tecnologici hanno considerevolmente migliorato la nostra capacità di sviluppare sistemi ingegneristici su rete (e.g., sistemi multi-veicolo, plotoni di veicoli, volo in formazione e squadre di robot, reti di sensori, reti elettriche, *social network*) il cui comportamento assomiglii a quello delle loro controparti naturali in termini di complessità operativa. In genere, gli agenti di un sistema su rete devono operare in concerto tra loro al fine di raggiungere opportuni obiettivi a livello di sistemi, tenendo conto della disponibilità di risorse computazionali limitate, di capacità di *sensing* limitate e della possibilità di ricorrere solamente a comunicazioni locali. Ciò ha suscitato nel dottorando un vivissimo interesse per il corpo di metodi che riguardano l'analisi e il progetto di *sistemi dinamici mulit-agente su rete*. In particolare, lo studio della letteratura recente ha portato il dottorando a riconoscere che, dal punto di vista metodologico, è possibile individuare un denominatore comune tale da porre le fondamenta quasi per la totalità dei risultati innovativi raggiunti nella letteratura sui sistemi dinamici multi-agente: tali risultati sono infatti basati principalmente su

- la teoria algebrica dei grafi, e
- la teoria del consenso.

Pertanto, in accordo con i propri relatori della tesi, il dottorando ha indirizzato le sue ricerche verso lo sviluppo di risultati metodologici innovativi, con particolare riferimento ai comportamenti di *multi-consenso* che si possono riscontrare nei sistemi multi-agente.

Tuttavia, tali aspetti sono stati approfonditi non soltanto dal punto di vista *metodologico*, ma anche dal punto di vista *applicativo*. Infatti, nel corso

degli ultimi tre anni, il dottorando ha dedicato diverso tempo alla risoluzione di problemi di interesse pratico, relativi ad alcune applicazioni tecnologiche nel dominio delle telecomunicazioni e delle reti elettriche e inseriti nel contesto di progetti di ricerca italiani ed europei, utilizzando le metodologie di teoria del controllo sopraindicate. A tal proposito, in qualità di consulente *in-house* per il CRAT (*Consorzio per la Ricerca nell'Automatica e nelle Telecomunicazioni*), il dottorando ha partecipato ai seguenti progetti, avendo a che fare non solo con i relativi processi di gestione, ma anche e soprattutto con le attività tecniche di tali progetti e con le relative sfide:

- progetto MIUR-PON PLATINO (*Platform for Innovative Services in Future Internet*, luglio 2012 – giugno 2015);
- progetto FP7 T-NOVA (*Network Functions as-a-service over Virtualised Infrastructures*, gennaio 2014 – dicembre 2016);
- progetto H2020 BONVOYAGE project (*From Bilbao to Oslo, intermodal mobility solutions, interfaces and applications for people and goods, supported by an innovative communication network*, maggio 2015 – aprile 2018);
- progetto H2020 ATENA (*Advanced tools to assess and mitigate the criticality of ICT components and their dependencies over critical infrastructures*, maggio 2016 – aprile 2019).

In particolare, il progetto PLATINO era incentrato sulla progettazione e sullo sviluppo di una piattaforma software cognitiva (vale a dire con architettura ad anello chiuso), in grado di fornire servizi e contenuti di vario tipo attraverso reti eterogenee, abbastanza flessibile per gestire una grande varietà di terminali, personalizzabile sulla base delle preferenze degli utenti e infine in grado di garantire la Qualità del Servizio e di misurare e massimizzare la Qualità dell'Esperienza dell'utente, tutto ciò coerentemente con il paradigma del *Future Internet*. Tale progetto ha potuto trarre beneficio da funzionalità di controllo della Qualità dell'Esperienza in grado di selezionare in modo dinamico le Classi di Servizio più adeguate mediante un innovativo algoritmo euristico, basato sul *Multi-Agent Reinforcement Learning* e progettato in modo tale da aggirare alcuni problemi implementativi e da ottenere prestazioni soddisfacenti in presenza di diverse centinaia di agenti. Il progetto T-NOVA, d'altro canto, prevedeva la progettazione e l'implementazione di una piattaforma di gestione/orchestrazione per la fornitura, la configurazione, il monitoraggio e l'ottimizzazione automatici di *Network Functions-as-a-Service* su reti virtualizzate e infrastrutture IT, perfezionando le piattaforme di *Software Defined Networking* (SDN) già esistenti: tale progetto ha potuto trarre

beneficio dall'implementazione di un algoritmo distribuito, non-cooperativo e a tempo discreto, per il bilanciamento dei carichi, basato sulla teoria dei giochi di campo medio e finalizzato al bilanciamento dinamico delle richieste degli *switch* tra i cosiddetti *SDN Controllers* per evitare congestioni nel traffico di controllo. I risultati di ricerca ottenuti nell'ambito delle attività dei progetti PLATINO e T-NOVA hanno dato luogo a quattro pubblicazioni su atti di conferenze internazionali e a tre pubblicazioni su rivista (delle quali due sono attualmente in revisione).

Il dottorando ha anche partecipato al progetto BONVOYAGE, finalizzato alla progettazione, allo sviluppo e alla validazione di una piattaforma che ottimizzi il trasporto multi-modale porta-a-porta di passeggeri e merci. Tale piattaforma integra le informazioni di viaggio e i servizi di pianificazione del viaggio e di acquisto del biglietto, analizzando automaticamente: i dati raccolti non in tempo reale da *database* eterogenei (sui sistemi di trasporto stradale, ferroviario e urbano), i dati misurati in tempo reale (riferiti al traffico e alle previsioni meteorologiche), i profili degli utenti e i *feedback* degli utenti. A tal proposito, il dottorando ha sviluppato un algoritmo per risolvere il problema del *many-to-many carpooling* con aggregazione automatica dei passeggeri nel contesto della pianificazione di viaggi multi-modali, assicurando che la soluzione proposta ponga rimedio al problema della complessità computazionale che sorge in scenari di dimensioni realistiche. Questo lavoro ha condotto ai risultati mostrati in due pubblicazioni su atti di conferenze internazionali. Tuttavia, tale argomento di ricerca non è ulteriormente approfondito in questa tesi di dottorato poichè esso non rientra nell'ambito della tematica dei sistemi dinamici multi-agente e delle relative applicazioni.

Il progetto ATENA (anch'esso attualmente in corso) prevede lo sviluppo di una *suite* integrata di componenti ICT su rete per la rilevazione e la mitigazione di eventi avversi nei sistemi ciber-fisici e, in particolare, sta traendo beneficio dala progettazione di uno schema di protezione basato sul piazzamento di sistemi di accumulo energetico (*energy storage systems*) per porre rimedio ad attacchi dinamici e ad anello chiuso responsabili dell'alterazione dei carichi di rete, offrendo così un efficace supporto ai servizi di regolazione di frequenza delle reti elettriche di trasmissione. Finora, i risultati delle relative ricerche hanno prodotto due pubblicazioni su atti di conferenze internazionali.

È chiaro dalla descrizione sopra che il corpo di metodi che riguardano l'analisi e il progetto dei sistemi dinamici multi-agente su rete rappresenta il *terreno comune* che soggiace ai progetti T-NOVA, PLATINO e ATENA. Ciò ha condotto il dottorando ad approfondire la letteratura recente e a ricercare significativi avanzamenti oltre lo stato dell'arte. Le relative attività di ricerca e i risultati conseguenti sono stati inquadrati nel Progetto di Ateneo (Sapienza, Università di Roma) su *Sistemi Nonlineari a Tempo Discreto e*

a Dati Campionati e sintetizzati in due pubblicazioni su atti di conferenze internazionali e un articolo su rivista (la cui preparazione è quasi conclusa). In particolare, il Progetto di Ateneo, tra le altre cose, prevede un'indagine sui modelli matematici che descrivono il comportamento dei sistemi dinamici multi-agente, con particolare attenzione a:

- il coordinamento distribuito dei sistemi multi-agente;
- il multi-consenso e le *almost equitable graph partitions*;
- la progettazione di leggi di controllo a dati campionati con applicazioni aerospaziali (tale argomento è stato approfondito nel primo anno di dottorato come naturale proseguimento del lavoro effettuato dal dottorando nella sua tesi di laurea magistrale).

Le pubblicazioni cui si è fatto riferimento costituiscono l'intera produzione scientifica del triennio dottorale nella quale il dottorando figura come co-autore.

Il respiro internazionale delle attività scientifiche e professionali condotte dal dottorando è dovuto non soltanto alla partecipazione attiva ai progetti di ricerca europei sopraindicati, ma anche, in modo particolare, alla co-tutela con l'*École Doctorale de Sciences et Technologies de l'Information et de la Communication* (ED STIC) presso *Centrale Supélec, Université Paris-Saclay* e al finanziamento ricevuto dall'Università Italo Francese (UFI), essendo il dottorando risultato vincitore del bando *Vinci 2016*.

La tesi di dottorato è suddivisa in cinque Capitoli:

- Capitolo 1. *Multi-Agent Reinforcement Learning*, bilanciamento dei carichi distribuito basato sul metodo diretto di Lyapunov e protezione della rete di trasmissione elettrica da attacchi ciber-fisici.
- Capitolo 2. Stato dell'arte sul coordinamento distribuito dei sistemi multi-agente.
- Capitolo 3. Il multi-consenso e le *almost equitable graph partitions*.
- Capitolo 4. Progettazione di leggi di controllo a dati campionati con applicazioni aerospaziali.

Perciò, la tesi inizia con una prima parte (costituita dal Capitolo 1) dedicata alla presentazione e alla discussione delle più rilevanti applicazioni dei sistemi di controllo multi-agente su rete nel contesto dei progetti T-NOVA, PLATINO e ATENA. Dopodiché, la seconda parte (costituita dai Capitoli 2, 3 e 4) presenta il contesto teorico che fa da sfondo a tali applicazioni, con particolare

riferimento al coordinamento distribuito dei sistemi multi-agente, e i risultati innovativi raggiunti in termini di multi-consenso e progettazione di leggi di controllo a dati campionati.

Contents

Introduction

Network science has emerged as a powerful conceptual paradigm in engineering. An unprecedented attention is currently being devoted to network theory and its applications by the scientific community since, on the one hand, it has become vital to gain a deeper understanding of the role that inter-elemental interactions play in the collective functionality of multi-layered systems in the biological sciences, and, on the other hand, technological advances have considerably improved our ability to synthesize networked engineering systems (e.g., multi-vehicle systems and platoons, formation flight and multi-robot teams, wireless sensor networks, energy networks, social networks) that resemble their natural counterparts in terms of operational complexity. Usually, agents in networked systems have to operate in concert with each other so as to achieve system-level objectives with limited computational resources, limited sensing capabilities and local communications.

This aroused in the PhD candidate a significant interest in the methodological framework of *networked multi-agent dynamical systems*. In particular, the investigation of the recent literature led the PhD candidate to find out that, from the methodological viewpoint, there is a common denominator laying the foundation for almost all the innovative results related to multi-agent dynamical systems: indeed, such results extensively rely on

- algebraic graph theory, and
- consensus dynamics.

With this in mind, in agreement with his supervisors, the PhD candidate has directed his research efforts to the development of innovative methodological results in terms of achieving multi-consensus behaviors in multi-agent systems.

Yet, these aspects have been investigated not just from the *methodological* viewpoint, but also from the point of view of *applications*.

Indeed, over the last three years, the PhD candidate has devoted a great effort to the task of solving practical problems in some real tecnological applications arising in the domain of telecommunication and power networks, within the context of Italian and European research projects, by resorting to

the above-mentioned control-theoretical methodologies. In this respect, as in-house consultant for CRAT (*Consorzio per la Ricerca nell'Automatica e nelle Telecomunicazioni*), the PhD candidate has participated in the following projects, dealing not just with the related project management phases and processes, but especially with the technical part of such projects and the related challenges:

- MIUR-PON PLATINO project (*Platform for Innovative Services in Future Internet*, July 2012 – June 2015);
- FP7 T-NOVA project (*Network Functions as-a-service over Virtualised Infrastructures*, January 2014 – December 2016);
- H2020 BONVOYAGE project (*From Bilbao to Oslo, intermodal mobility solutions, interfaces and applications for people and goods, supported by an innovative communication network*, May 2015 – April 2018);
- H2020 ATENA project (*Advanced tools to assess and mitigate the criticality of ICT components and their dependencies over critical infrastructures*, May 2018 – April 2019).

In particular, the PLATINO project dealt with the design and development of a service platform capable of delivering heterogeneous services and contents, through dissimilar networks, flexible enough to handle a wide range of terminals, context-aware, customable on the basis of the user preferences, and finally capable of guaranteeing the Quality of Service and of measuring and maximizing the customer's Quality of Experience, all in compliance with the Future Internet paradigm. This project could actually benefit from Quality of Experience control functions capable of dynamically selecting the most appropriate Classes of Service based on an innovative heuristic Multi-Agent Reinforcement Learning algorithm designed so as to get around some practical implementation problems and achieve satisfactory performance results even in the presence of several hundreds of agents. The T-NOVA project, instead, dealt with the design and implementation a management/orchestration platform for the automated provision, configuration, monitoring and optimization of Network Functions-as-a-Service over virtualised Network/IT infrastructures by enhancing the already existing Software Defined Networking (SDN) platforms: this project could benefit from a discrete-time distributed and non-cooperative load balancing algorithm based on mean field game theory aimed at dynamically balancing the requests of the switches among the SDN Controllers to avoid congestion in the control traffic. The research results achieved within the framework of the activities of the PLATINO and T-NOVA projects led to the following publications/paper submissions:

- L. Ricciardi Celsi, S. Battilotti, F. Cimorelli, C. Gori Giorgi, S. Monaco, M. Panfili, V. Suraci, and F. Delli Priscoli, "A Q-Learning Based Approach to Quality of Experience Control in Cognitive Future Internet Networks," in *Proceedings of the 23rd Mediterranean Conference on Control and Automation (MED 2015)*, pp. 1045-1052, June 16-19, 2015, Torremolinos, Spain;

- S. Battilotti, S. Canale, F. Delli Priscoli, L. Fogliati, C. Gori Giorgi, F. Lisi, S. Monaco, L. Ricciardi Celsi, and V. Suraci, "A Dynamic Approach to Quality of Experience Control in Cognitive Future Internet Networks," poster appearing in *Proceedings of the 24th European Conference on Networks and Communications (EuCNC 2015)*, June 29 - July 2, 2015, Paris, France;

- S. Battilotti, F. Delli Priscoli, C. Gori Giorgi, S. Monaco, M. Panfili, A. Pietrabissa, L. Ricciardi Celsi, and V. Suraci, "A Multi-Agent Reinforcement Learning Based Approach to Quality of Experience Control in Future Internet Networks," in *Proceedings of the 34th Chinese Control Conference (CCC 2015)*, pp. 6495-6500, July 28-30, 2015, Hangzhou, China;

- F. Cimorelli, F. Delli Priscoli, A. Pietrabissa, L. Ricciardi Celsi, V. Suraci, and L. Zuccaro, "A Distributed Load Balancing Algorithm for the Control Plane in Software Defined Networking," in *Proceedings of the 24th Mediterranean Conference on Control and Automation (MED 2016)*, pp. 1033-1040, June 21-24, 2016, Athens, Greece;

- F. Delli Priscoli, A. Di Giorgio, F. Lisi, S. Monaco, A. Pietrabissa, L. Ricciardi Celsi, and V. Suraci, "Multi-Agent Quality of Experience Control," *International Journal of Control, Automation, and Systems*, vol. 15, no. 2, pp. 892-904, 2017;

- V. Suraci, L. Ricciardi Celsi, A. Giuseppi, and A. Di Giorgio, "A distributed Wardrop control algorithm for load balancing in smart grids," in *Proceedings of the 25th Mediterranean Conference on Control and Automation (MED 2017)*, art. no. 7984210, pp. 761-767, July 3-6, 2017, Valletta, Malta;

- A. Pietrabissa, L. Ricciardi Celsi, F. Cimorelli, V. Suraci, F. Delli Priscoli, A. Di Giorgio, and S. Monaco, "Lyapunov-based design of a distributed Wardrop load balancing algorithm with application to Software Defined Networking," submitted to the *IEEE Transactions on Control Systems Technology*, 2017;

- A. Pietrabissa and L. Ricciardi Celsi, "Discrete-Time Load Balancing Converging to the Wardrop Equilibrium," submitted to the *IEEE Transactions on Automatic Control*, 2017.

The PhD candidate has also participated in the BONVOYAGE project, dealing with the design, development and testing of a platform optimizing the multi-modal door-to-door transport of passengers and goods. Such a platform integrates travel information, planning and ticketing services, by automatically analyzing: non-real-time data collected from heterogeneous databases (on road, railway and urban transport systems), real-time measured data (traffic, weather forecasts), user profiles, and user feedback. In this respect the PhD candidate developed an algorithm for solving the many-to-many carpooling problem with automated passenger aggregation in the context of multi-modal trip planning, ensuring that the proposed solution reasonably copes with the computational complexity that arises in real-sized scenarios. This work led to the results shown in the following publications:

- S. Canale, A. Di Giorgio, F. Lisi, M. Panfili, L. Ricciardi Celsi, V. Suraci, and F. Delli Priscoli, "A Future Internet Oriented User Centric Extended Intelligent Transportation System," in *Proceedings of the 24th Mediterranean Conference on Control and Automation (MED 2016)*, pp. 1133-1139, June 21-24, 2016, Athens, Greece;

- L. Ricciardi Celsi, A. Di Giorgio, R. Gambuti, A. Tortorelli, and F. Delli Priscoli, "On the many-to-many carpooling problem in the context of multi-modal trip planning," in *Proceedings of the 25th Mediterranean Conference on Control and Automation (MED 2017)*, art. no. 7984135, pp. 303-309, July 3-6, 2017, Valletta, Malta.

Yet, such a research topic is not detailed further in this PhD thesis, as it is beyond the scope of multi-agent dynamical systems and their specific applications.

The ongoing ATENA project deals, instead, with the development of a suite of integrated ICT networked components for detection of and reaction to adverse events in Cyber-Physical Systems and, in particular, benefits from the design of a protection scheme based on energy storage system placement against closed-loop dynamic load altering attacks, thus successfully supporting frequency regulation services in power transmission networks. So far, the related research results have led to the following publications:

- A. Di Giorgio, F. Liberati, R. Germanà, M. Presciuttini, L. Ricciardi Celsi, and F. Delli Priscoli, "On the Control of Energy Storage Systems for Electric Vehicles Fast Charging in Service Areas," in *Proceedings of*

the 24th Mediterranean Conference on Control and Automation (MED 2016), pp. 955-960, June 21-24, 2016, Athens, Greece;

- A. Di Giorgio, A. Giuseppi, F. Liberati, A. Ornatelli, A. Rabezzano and L. Ricciardi Celsi, "On the optimization of energy storage system placement for protecting power transmission grids against dynamic load altering attacks," in *Proceedings of the 25th Mediterranean Conference on Control and Automation (MED 2017)*, pp. 986-992, July 3-6, 2017, Valletta, Malta.

It is clear from the description above that the methodological framework of networked multi-agent dynamical systems represents the *common ground* that lies beneath the T-NOVA, PLATINO and ATENA projects. This has led the PhD candidate to investigate the recent literature and search for meaningful advances beyond the state of the art. The corresponding research activities and results have been carried out and achieved within the framework of the *Progetto di Ateneo* (*Sapienza, Università di Roma*) on *Discrete-Time and Sampled-Data Nonlinear Systems*, leading to the following publications/paper submissions:

- L. Ricciardi Celsi, R. Bonghi, S. Monaco, and D. Normand-Cyrot, "On the Exact Steering of Finite Sampled Nonlinear Dynamics with Input Delays," in *Proceedings of the 1st Conference on Modelling, Identification and Control of Nonlinear Systems (MICNON 2015)*, *IFAC-PapersOnLine*, vol. 48, no. 11, pp. 674-679, June 24-26, 2015, Saint-Petersburg, Russia;

- L. Ricciardi Celsi, R. Bonghi, S. Monaco, and D. Normand-Cyrot, "Sampled-Data Stabilization Around the L2 Translunar Libration Point," in *Proceedings of the 3rd IAA Conference on University Satellite Missions and CubeSat Workshop & International Workshop on Lean Satellite Standardization*, Roma, Italy, November 30 - December 5, 2015, in press;

- S. Monaco and L. Ricciardi Celsi, "On Multi-Consensus and Almost Equitable Graph Partitions," submitted to *Automatica*, 2017.

In particular, the *Progetto di Ateneo* revolves around the study of the mathematical models describing the behavior of multi-agent dynamical systems, with specific focus on:

- the distributed coordination of multi-agent systems;

- multi-consensus and almost equitable graph partitions;

- sampled-data design with aerospace applications (investigated as a follow-up to the work carried out by the PhD candidate in his M.Sc. thesis).

The works reported above represent all the scientific production co-authored by the PhD candidate over the three years of the doctoral programme.

The international extension of the cultural, scientific and professional scope of the activities carried out by the PhD candidate is due not just to the active participation in the above-mentioned European research projects, but also, notably, to the *co-tutelle* with the *Ecole Doctorale de Sciences et Technologies de l'Information et de la Communication* (ED STIC) at *Centrale Supélec, Université Paris-Saclay* and to the funding received by the *Université Franco Italienne* (UFI) after winning the *Vinci 2016* competition.

The PhD thesis is organized into four Chapters, as follows:

- Chapter 1: Multi-Agent Quality of Experience Control, Lyapunov-Based Discrete-Time Load Balancing, and Power Transmission Grid Protection Against Cyber-Physical Attacks.
- Chapter 2: State of the Art on the Distributed Coordination of Multi-Agent Systems.
- Chapter 3: Multi-Consensus and Almost Equitable Graph Partitions.
- Chapter 4: Sampled-Data Design with Aerospace Applications.

So, the thesis begins with a first part (represented by Chapter 1) devoted to the presentation and discussion of the most relevant applications of networked multi-agent control systems in the context of the T-NOVA, PLATINO and ATENA projects. Then, the second part (consisting of Chapters 2, 3 and 4) presents the underlying theoretical background with respect to the distributed coordination of multi-agent systems and the innovative results achieved with respect to multi-consensus and sampled-data design.

Chapter 1

Multi-Agent Quality of Experience Control, Lyapunov-Based Discrete-Time Load Balancing, and Power Transmission Grid Protection Against Cyber-Physical Attacks

This Chapter collects the results of the research activites carried out with respect to telecommunication networks in the framework of the PLATINO project (see Section 1.1) and of the T-NOVA project (see Sections 1.2 and 1.3). It also collects the research results the PhD candidate has achieved so far with respect to the protection of power transmission networks against cyber-physical attacks, in the framework of the ATENA project.

More precisely, Section 1.1 presents the contents of the publications [156], [12], [38], and [49], whereas Sections 1.2 and 1.3 present the contents of the two submitted papers [147] and [148], which are currently under review. In Section 1.4, instead, a power system protection scheme based on energy storage system placement against closed-loop dynamic load altering attacks is proposed as discussed in [159]. The protection design consists in formulating a non-convex optimization problem, subject to a Lyapunov stability constraint and solved using a two-step iterative procedure. Simulation results confirm the effectiveness of the approach and the potential relevance of using energy storage systems in support of primary frequency regulation services.

1.1 Multi-Agent Quality of Experience Control

A key Future Internet target is to allow applications to transparently, efficiently and flexibly exploit the available resources, with the aim of achieving a satisfaction level that meets the personalized users needs and expectations. Such expectations could be expressed in terms of a properly defined *Quality of Experience* (QoE). In this respect, the International Telecommunication Union (ITU-T) defines QoE as the overall acceptability of an application or service, as perceived subjectively by the end-user: this means that QoE could be regarded as a personalized function of plenty of parameters of heterogeneous nature and spanning all layers of the protocol stack (e.g., such parameters can be related to Quality of Service (QoS), security, mobility, contents, services, device characteristics, etc.).

Indeed, a large amount of research is ongoing in the field of QoE Evaluation, i.e., of the identification, on the one hand, of the personalized expected QoE level (*Target QoE*) for a given user availing her/himself of a given application in a given context (e.g., see [89] and [169] for voice and video applications, respectively), and, on the other hand, of the personalized functions for computing the *Perceived QoE*, including the monitorable Feedback Parameters which could serve as independent variables for these functions (e.g., see [27]). In particular, several works focus on studying the relation between QoE and network QoS parameters (e.g., see [64]).

Another QoE-related key research issue is that of QoE Control. Once a *QoE Evaluator* has assessed the personalized expected QoE level (Target QoE) and the personalized currently perceived QoE level (Perceived QoE), a *QoE Controller* should be in charge of making suitable Control Decisions aimed at reducing, as far as possible, the difference between the personalized Target and Perceived QoE levels. QoE Evaluation and QoE Control have also been widely studied in the context of several Future Internet related initiatives such as the MIUR PLATINO project and the FP7 Future Internet PPP initiative.

This Section focuses on QoE Control, whereas QoE Evaluation falls outside its scope. The interested readers are referred to [27] for an approach to QoE Evaluation that is fully consistent with the QoE Controller presented in this Section. Without claiming to present a ready-to-use solution, this Section provides some innovative hints that could ensure an efficient implementation of the QoE Controller. Namely, we describe how Control Decisions can practically be implemented via the dynamic selection of predefined Classes of Service. We then explain how such a dynamic selection can be performed in

a model-independent way – in the authors' opinion, a control-based approach relying on any Future Internet model is not practically viable due to the sheer unpredictability of the involved variables [156] – thanks to the adoption of a multi-agent algorithm. A suitable algorithm was identified in a *Multi-Agent Reinforcement Learning* (MARL) technique, namely the *MARL Q-Learning* algorithm presented in [103] and [26]. Then, the paper discusses the limitations of MARL Q-Learning with respect to practical implementation and how these limitations can be overcome by adopting the proposed heuristic algorithm, hereafter referred to as H-MARL-Q algorithm. Finally, some numerical simulations showing the encouraging performance results of the proposed heuristic algorithm are presented with reference to a proof-of-concept scenario which does not claim to represent any real network.

1.1.1 QoE Controller Architecture

The QoE Controller makes its decisions at discrete time instants t_k, hereafter referred to as time steps, occurring with a suitable time period T, whose duration depends on the considered environment (including technological processing constraints). We assume that each in-progress application instance is handled by an *Agent* i and we define the personalized *QoE Error* at time t_k (indicated as $e_i(t_k)$), relevant to Agent i, as

$$e_i(t_k) = PQoE_i(t_k) - TQoE_i, \tag{1.1}$$

where $PQoE_i(t_k)$ represents the Perceived QoE, i.e., the QoE currently perceived at time t_k by Agent i, and $TQoE_i$ represents the Target QoE, i.e., the personalized QoE which would satisfy the personalized Agent i requirements. So, if this QoE Error is positive, the in-progress application is said to be overperforming, since the QoE currently perceived by the Agent is greater than the desired one, whereas, if the QoE Error is negative, the in-progress application is said to be underperforming. Note that the presence of overperforming Agents might affect the system performance, since they may require an unnecessarily large amount of resources, which could cause, in turn, the underperformance of other Agents. The goal of the QoE Controller is to guarantee, at every time t_k, a nonnegative QoE Error for all Agents i (for $i = 1, \ldots, N$), i.e., to avoid the occurrence of underperforming applications. Furthermore, if it is not possible to guarantee a nonnegative QoE Error for all Agents (e.g., due to insufficient network resources), the QoE Controller should reduce, as far as possible, the QoE Errors of the various Agents while guaranteeing fairness among them. Fairness basically consists in making sure that the QoE Errors experienced by the Agents are kept, as far as possible,

close to one another. As shown in Fig. 1.1, both the Perceived and the Target QoE should be computed by a suitable QoE Evaluator based on suitable Feedback Parameters resulting from the real-time monitoring of the network, as well as from direct or indirect feedbacks coming from users and/or applications. For a more detailed description of the way the QoE functionalities are embedded in the Future Internet architecture, see [31] and [23].

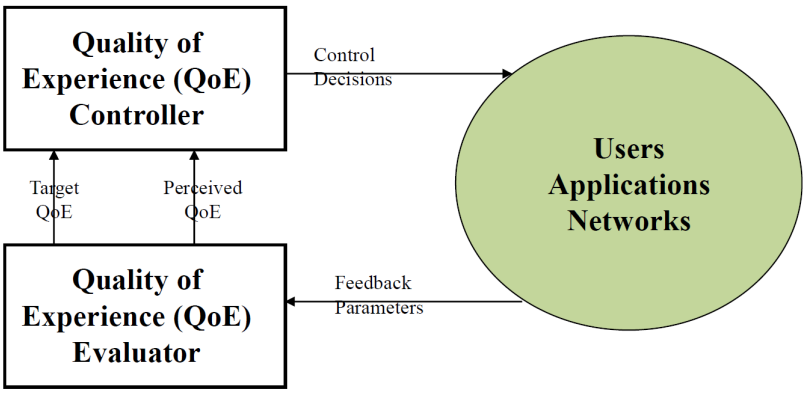


Figure 1.1: Sketch of the QoE architecture for the Future Internet.

In particular, a promising approach [27] is to relate the computation of the Perceived QoE to the application type (e.g. real-time HDTV streaming, distributed videoconferencing, File Transfer Protocol, etc.) of each in-progress application instance. Let M denote the total number of application types in the considered environment; let $m \in \{1, \ldots, M\}$ denote a generic application type; let $i(m)$ denote an Agent (i.e., an application instance) belonging to the m-th application type. Then, the Perceived QoE for Agent $i(m)$, denoted with $PQoE_{i(m)}(t_k)$, is computed as follows:

$$PQoE_{i(m)}(t_k) = g_m(\phi_m(t_k)), \tag{1.2}$$

where $\phi_m(t_k)$ represents a suitable set of Feedback Parameters for the m-th application type, computed up to time t_k, and g_m is a suitable function relating, for the m-th application type, the Feedback Parameters $\phi_m(t_k)$ with the Perceived QoE. In the following, we shows a simple implementation of (1.2). A relevant drawback that could be immediately associated with such a method of evaluating the Perceived QoE for every Agent at each time step is the fact that an Agent can intentionally underreport its own Perceived QoE in order to increase the amount of network resources allocated to it. Such a problem falls within the area of mechanism design [132]. In this

respect, in [47], the authors propose an interesting solution (compliant with the control algorithm discussed here) that allows to determine whether the Agent feedbacks are being fair or not. Such a solution is capable of ensuring an acceptable degree of robustness to possible episodes of dishonest Agent conduct.

The Target QoE, denoted with $TQoE_i$, can be derived from a suitable analysis of the available Feedback Parameters (e.g., by using unsupervised machine learning techniques), or it can simply correspond to a reference value which is assigned by the Telco operator, taking into account the commercial profile of the user.

In the following, we propose a solution in which the distributed Agents associated to the application instances are embedded in properly selected network nodes (e.g., in the mobile user terminals): the Agents are in charge of the monitoring and actuation functionalities whereas the control functionalities are centralized in the QoE Controller. In particular, whenever a new application instance is born, the associated Agent i is in charge of evaluating the personalized Target QoE $TQoE_i$ (which remains unchanged for the whole lifetime of the application instance), of computing its own personalized Perceived QoE $PQoE_i(t_k)$ and of communicating the monitored values to the QoE Controller. As a result, at each time t_k, the QoE Controller, based on the received values for $TQoE_i$ and $PQoE_i(t_j)$ up to time t_k ($i = 1, \ldots, N$; $j = 0, 1, \ldots, k$), has to choose the most appropriate action $a_i(t_k)$ (for $i = 1, \ldots, N$) which the Agent i should enforce at time t_k, i.e., the most appropriate joint action $(a_1(t_k), a_2(t_k), \ldots, a_N(t_k))$ which the N Agents should enforce at time t_k. At each time t_k, the chosen joint action is broadcast to the N Agents: then, the i-th Agent has to enforce the corresponding action $a_i(t_k)$.

Note that the proposed arrangement is based on the presence of a centralized entity (i.e., the QoE Controller), collecting the Agents observations, which runs the MARL algorithm and broadcasts the resulting Control Decisions to the Agents. Therefore, any direct signal exchange among the Agents is avoided, thus limiting the overall signalling overhead. The QoE Controller outputs, i.e, the joint action chosen by the QoE Controller, may include for each Agent the choice of QoS Reference Values (e.g., the expected priority level, the tolerated transfer delay range, the minimum throughput to be guaranteed, the tolerated packet loss range, the tolerated dropping frequency range, etc.), of Security Reference Values (e.g., the expected encryption level, the expected security level of the routing path computed by introducing appropriate metrics, etc.), and of Content/Service Reference Values (e.g., the expected content/service mix, etc.).

The QoE Controller has to dynamically select, for each in-progress appli-

cation instance, the most appropriate Reference Values which should actually drive, thanks to suitable underlying network procedures (which are outside the scope of this Section), the Perceived QoE as close as possible to the Target QoE – for further details, see [156] where the above-mentioned Reference Values are referred to as Driving Parameters. However, since the control action has a large number of degrees of freedom, the exploration of the solution space may take a large amount of time, thus making the task of the QoE Controller excessively complex. A simpler (yet less fine-grained) control task arises if the management of the underlying networks is arranged into Classes of Service (CoS), as described in [58].

We assume that each CoS is associated with a predefined set of QoS Reference Values. Nevertheless, the proposed approach can be applied even in the case when each CoS is associated with a set of Reference Values that are not necessarily related to QoS issues only, but also, for instance, to Security parameters, and/or to Content/Service characteristics, etc. Let S indicate the total number of CoSs and let $a_i(t_k) \in c_1, c_2, \ldots, c_5$ indicate the action performed by the i-th Agent (i.e., the CoS chosen by the i-th Agent) at the time instant t_k.

In current telecommunication networks, a static CoS assignment policy is adopted: each application instance is given a CoS for its entire lifetime; the CoS associated to a given application instance should be the one whose QoS Reference Values satisfy on the average the application requirements. Nevertheless, it is evident that such a static association does not take into account either personalized application requirements or contingent situations taking place in the telecommunication networks, such as congestion events. So, a static CoS assignment may generally lead to poor performance in terms of the personalized QoE perceived by each user. Hence, this Section considers dynamic CoS-to-application assignment as the methodological means to accomplish the above-mentioned goals in terms of QoE Error reduction and fairness. This means that, at each time instant t_k, the QoE Controller has to decide, in real time, which is the most appropriate CoS to be associated with each in-progress application instance (e.g., if the Agents are embedded in mobile user terminals, the QoE Controller decisions can be implemented by inserting the selected CoS identifier in the header of the packets transmitted by the terminals). Up to the authors' knowledge, apart from [31], [156], and [12], such a dynamic assignment approach has never been investigated so far.

Indeed, meeting the Target QoE for the in-progress applications, in conjunction with an efficient exploitation of the available bandwidth, could be a rather challenging issue, especially in wireless networks with limited bandwidth resources. In this respect, optimal adaptive control strategies could be

key factors to cope with such an issue. Moreover, due to the data-intensive nature of multimedia streaming services as well as due to the increasingly demanding requirements in terms of QoS/QoE, Reinforcement Learning based algorithms are being used more and more in telecommunication networks, as long as they prove to be computationally efficient and sufficiently scalable [91].

1.1.2 The MARL-Q Algorithm for the QoE Controller

This Section focuses on the problem of designing the QoE Controller algorithm. It should be evident that, in order to solve this problem by means of traditional *model-based* control techniques, the QoE Controller should know – or at least estimate – the correlation between its decisions (namely, the selected QoE Controller outputs) and the Perceived QoE. However, no model of the very complex plant regulated by the QoE Controller (namely, the plant receiving the QoE Controller outputs in input and producing the Perceived QoE as its output) can be assumed, since it depends on plenty of hardly predictable factors (such as traffic characteristics of the ongoing applications, network topologies, resource management algorithms, QoE Evaluation methods and so on).

In light of the above, the QoE Controller decision strategy must be learned online by *trial and error*. This is why we propose that the QoE Controller makes use of a *model-free* MARL algorithm in order to evaluate, at each time step t_k, the joint policy $\pi(a_1(t_k), a_2(t_k), \ldots, a_N(t_k)) = \pi(a_1, a_2, \ldots, a_N)$ which, once enforced by the Agents, tracks the discussed goals in terms of QoE Error. The proposed MARL algorithm works on the basis of the observation of a joint reward $r(t_{k+1}, a_1(t_k), a_2(t_k), \ldots, a_N(t_k)) = r(t_{k+1}, a_1, a_2, \ldots, a_N)$, i.e., of the numerical reward (the same for all the N Agents) which is received by each Agent at time t_{k+1} as a consequence of the enforcement, at time t_k, of the joint policy $\pi(a_1, a_2, \ldots, a_N)$. The MARL algorithm in question is aimed at maximizing the long-run return $R(\pi)$, namely at maximizing the expected discounted return:

$$R(\pi) = E_\pi\left\{\sum_{i=0}^{\infty} \gamma^k \cdot r(t_{k+1}, a_1, a_2, \ldots, a_N)\right\}, \tag{1.3}$$

where $\gamma \in [0, 1)$ is the discount rate, which weighs immediate versus delayed rewards, and $E_\pi\{\cdot\}$ denotes the expected value under policy π.

In order to set up a MARL problem, we have to select the state space, the action spaces and the reward function.

- We consider a *static game*, i.e., a game with only a single state: such an assumption, on the one hand, is not limiting in our context, and, on

the other hand, greatly reduces the computational complexity which in MARL is exponential in the number of state and action variables.

- Following the discussion on dynamic CoS assignment, the action set A_i of Agent i coincides with the set of CoSs, i.e., $A_i = \{c_1, c_2, \ldots, c_S\}, i = 1, \ldots, N$. In other words, action $a_i(t_k)$, performed by Agent i at time t_k, can be equal to either c_1, or $c_2, \ldots$, or c_S. The cardinality of the joint action space $\mathbf{A} = A_1 \times \ldots \times A_N$ is equal to $|A_1| \cdot |A_2| \cdot \ldots \cdot |A_N| = S^N$.

- The function expressing the joint reward $r(t_{k+1}, a_1, a_2, \ldots, a_N)$ should be consistent with the discussed goals in terms of QoE Error; in this respect, each candidate joint reward should be a non-increasing function of the N error values $|e_i(t_k)|$ (for $i = 1, \ldots, N$). Below, the choice of suitable joint reward functions will be discussed.

In particular, we propose to apply the Multi-Agent Q-Learning algorithm in [103] (hereinafter referred to as *MARL-Q algorithm*) which is proved to converge to an optimal policy $\pi^*(a_1, a_2, \ldots, a_N)$, i.e., to a policy which maximizes the expected discounted long-run return $R(\pi)$. The algorithm is the multi-agent extension of the well-known (single-agent) Q-Learning algorithm [179], already succesfully applied to QoE/QoS control in communication networks [163] [145].

The MARL-Q algorithm relies on the estimation of the optimal action-value function $Q(s, a_1, a_2, \ldots, a_N)$, defined as the expected return of the system when it starts from state s, takes the joint action $(a_1, a_2, \ldots, a_N)$, and follows policy π thereafter. In the previously defined centralized context, at each time step t_k, this algorithm (i) evaluates a joint policy $\pi(a_1, a_2, \ldots, a_N)$ which sums up the behaviour of all the N Agents and is initialized arbitrarily and (ii) improves such a policy by making it ε-greedy with respect to the current action-value function [174], thus yielding a better joint policy $\pi^{'}$ to be evaluated and improved at the next iteration.

In detail, the policy evaluation step (i) is performed by the MARL-Q algorithm by updating the action-value function $Q(t_k, a_1, a_2, \ldots, a_N)$ according to the following update rule:

$$\begin{aligned} &Q(t_k, a_1, a_2, \ldots, a_N) = (1 - \alpha(t_k))Q(t_{k-1}, a_1, a_2, \ldots, a_N) + \\ &+ \alpha(t_k)[r(t_k, a_1, a_2, \ldots, a_N) + \gamma \max_{a_1 \in A_1, \ldots, a_N \in A_N} Q(t_{k-1}, a_1, a_2, \ldots, a_N)], \end{aligned} \tag{1.4}$$

where $\gamma \in [0, 1)$ is the discount rate and $\alpha(t_k)$ is a sequence of learning rates, which are key parameters that should satisfy the standard stochastic approximation conditions for convergence [87]. The argument t_k denotes the

value of the action-value function computed at time t_k, whereas the argument s is omitted since we are considering a single state problem.

The policy improvement step (ii) consists in performing, with probability equal to ε, a random joint action $(a_1^{'}, a_2^{'}, \ldots, a_N^{'})$ and, with probability equal to 1ε, the following greedy joint action $(a_1^{'}, a_2^{'}, \ldots, a_N^{'})$:

$$(a_1^{'}, a_2^{'}, \ldots, a_N^{'}) = arg \max_{a_1 \in A_1, \ldots, a_N \in A_N} Q(t_k, a_1, a_2, \ldots, a_N). \quad (1.5)$$

The parameter $\varepsilon \in (0, 1)$ is the exploration rate. A large value of ε guarantees that different policies with respect to the current best one are explored, and thus avoids that the QoE Controller remains stuck in a local minimum (*exploration*); on the other hand, a small value of ε lets the QoE Controller choose the best action based on the current estimate of the action-value function (*exploitation*).

So, at each time step t_k, the centralized QoE Controller based on the Perceived QoE values $PQoE_i(t_k)$ $(i = 1, \ldots, N)$ transmitted by the Agents at time t_k, and on the knowledge of the Target QoE values $TQoE_i$ $(i = 1, \ldots, N)$ transmitted by the Agents at the time of their birth performs the following tasks until the optimal action-value function Q^* (and the optimal policy π^*) is found:

T1) it updates the action-value function Q according to (1.4);

T2) it determines the joint action $(a_1^{'}, a_2^{'}, \ldots, a_N^{'})$ in a random way with probability equal to ε and according to (1.5) with probability equal to $1 - \varepsilon$;

T3) it broadcasts the chosen joint action $(a_1^{'}, a_2^{'}, \ldots, a_N^{'})$ to all Agents so that Agent i consequently enforces action $a_i^{'}$;

T4) it computes the corresponding joint reward $r(t_{k+1}, a_1^{'}, a_2^{'}, \ldots, a_N^{'})$ according to the selected reward function which should include, as independent variables, the Perceived QoE values $PQoE_i(t_k)$ $(i = 1, \ldots, N)$ and the Target QoE values $TQoE_i$ $(i = 1, \ldots, N)$. The algorithm converges under a generic initial policy. By varying the learning rates, the exploration rate and the discount rate, the convergence speed of the algorithm and the quality of the solution significantly change; the parameters used in the simulations reported below have been tuned by running the simulations several times.

1.1.3 Proposed Heuristic MARL-Q Based (H-MARL-Q) Algorithm

Limitations of MARL-Q

The analysis of the contents of the previous Section offers us the opportunity to discuss the following issues.

- The main challenge arisen in MARL is the so-called *curse of dimensionality* [26]: in fact, as Reinforcement Learning algorithms (such as Q-Learning) estimate values for each possible state or state-action pair, the computational complexity of MARL is exponential in the number of state and action variables and, therefore, in the number of Agents; in addition, the Agents rewards are correlated and then they cannot be maximized independently of one another. The runtime of the MARL-Q algorithm (i.e., the time the algorithm needs to perform the specific task it has been designed for) directly depends on the cardinality S^N of the joint action space. As a matter of fact, at each time step, the *max* operator in (1.5) has to consider S^N values; in this respect, it is particularly important to note that, in a Future Internet framework where the QoE Controller should be able to handle even thousands of Agents and dozens of CoSs, S^N would become a really huge value. For this reason, the task of implementing the dynamic CoS assignment according to the MARL-Q algorithm discussed in the previous section is inherently complex from a computational point of view and, as a result, it is extremely runtime-consuming. Such a relevant issue claims for a reasonable reduction of the size of the joint action space (and, hence, of the computational effort of the learning algorithm).

- The issue of the nonstationarity of multi-agent learning arises too, since all Agents in the system are simultaneously learning: each Agent is faced with a moving-target learning problem and consequently the best policy changes as the other Agents policies change. In this respect, the exploration strategy is crucial for the efficiency of MARL algorithms. Agents explore to obtain information not only about the environment, but also about the other Agents, for the purpose of implicitly building models of these Agents. In other words, the need for coordination stems from the fact that the effect of any Agents action on the environment depends also on the actions taken by the other Agents. Nonetheless, too much exploration should be avoided, as it may destabilize the learning dynamics of the other Agents.

In order to address the above-mentioned limitations, this Section presents an

innovative heuristic algorithm, hereafter referred to as *H-MARL-Q algorithm* and derived from the MARL-Q algorithm described above. Such a heuristic algorithm, in comparison with the latter, considerably reduces the joint action space, thus significantly accelerating the task of dynamic CoS mapping, without teasing out an excessive amount of exploratory and information-gathering actions (hence, preserving an acceptable level of environment exploration). As shown below, the proposed H-MARL-Q algorithm has also turned out to be successful in addressing the issue of the algorithm scalability, yielding satisfactory results even when the number of Agents is counted in the order of thousands (as it is happening in the ongoing Internet of Things era).

H-MARL-Q Algorithm Description

The H-MARL-Q algorithm only considers a suitably selected subset of the joint action space, reasonably yielding an approximate solution to the dynamic CoS assignment problem presented above.

Basically, at each time step, the entire joint action space contains plenty of joint actions which have very few possibilities of being the best ones (i.e., the ones which meet the *max* operator in (1.5). Unfortunately, such joint actions cannot be identified and discarded *a-priori*, because we do not have any *a-priori* knowledge of the environment; nevertheless, such actions can be identified and removed by carrying out a preliminary analysis of the Agents' dynamic behaviour in a simpler emulated environment. So, the basic underlying idea of the H-MARL-Q algorithm is to perform the following two steps.

Step (a): This step, referred to as *Identification of the Reduced Joint Action Space*, is performed by the QoE Controller *una tantum*, every time a new Agent is born, in order to identify, through the emulation of suitable test environments, an appropriate *Reduced Joint Action Space*.

Step (b): This step, referred to as *Identification of the Suboptimal Joint Action*, is performed, in real time, by the QoE Controller at each time step t_k, in order to identify the joint action $(a_1, a_2, \ldots, a_N)$ to be performed at time t_k on the basis of real-time observations of the environment and considering the Reduced Joint Action Space identified in Step (a) (and not the entire joint action space $\mathbf{A}$). This yields a suboptimal joint policy which constitutes a satisfactory approximate solution to the considered problem.

H-MARL-Q Algorithm Description: Step (a)

Whenever a new Agent, say agent N, is born (i.e., a new application instance is launched), say at time t_k, in a real environment in which $N-1$ Agents i (for $i = 1, 2, \ldots, N-1$) are already active, the new Agent notifies its existence to the QoE Controller together with its own personalized QoE requirements expressed in terms of Target QoE ($TQoE_N$). Then, the QoE Controller emulates the dynamic behaviour of the system in $N-1$ two-player test games, each one involving two Agents: (i) the new Agent N and (ii) each of the already active Agents i ($i = 1, \ldots, N-1$). These two-player test games are played in emulated test environments which should reproduce only some key features of the real environment. Let $[i, j]$ denote the two-player test game involving Agents i and j. In each two-player test game $[i, j]$ the optimal policy $\pi^*(a_i, a_j)$ is obtained by applying the MARL-Q algorithm described in the previous section (clearly, in this case, the number of Agents N appearing in (1.4) and (1.5) is equal to two). The optimal policy identifies a pair of deterministic actions (a_i^*, a_j^*) where a_i^* and a_j^* represent the optimal CoS choices that the Agents i and j, respectively, should enforce.

It should be clear that, since the cardinality of the joint action space of each test environment is equal to S^2, the computational complexity of the MARL-Q algorithm is limited, i.e., the algorithm converges to the optimal policy in a limited runtime as shown below through real tests. After Step (a), at any time t_k at which N Agents are active, the QoE Controller stores $N(N-1)/2$ optimal action couples:

$$(a_i^*, a_j^*) \text{ with } i = 1, \ldots, N,\ j = 1, \ldots, N,\ i \neq j. \tag{1.6}$$

These couples are used in order to identify a Reduced Joint Action Space containing a reasonable subset of the entire joint action space $\mathbf{A}$. Let $a_i^*[i, j]$ and $a_j^*[i, j]$ denote the optimal action for the i-th Agent and the j-th Agent, respectively, resulting from the two-player test game $[i, j]$. We assume that such a Reduced Joint Action Space consists of the union of N Action Subspaces, where the i-th Action Subspace is associated to the i-th Agent (the sub-tables within the borders in bold in the table below represent such Action Subspaces). Each Action Subspace includes S candidate joint actions (i.e., the rows of each sub-table). The i-th Action Subspace is built by only considering the two-player test games involving the i-th Agent. In particular, each of the S candidate joint actions of the i-th Action Subspace is obtained as follows: for each Agent j, with $j \neq i$, the optimal action $a_j^*[i, j]$ that such an Agent would perform in the two-player test game $[i, j]$ is taken into account, whilst for the i-th Agent all the S possible actions of the $\mathbf{A}_i$ set are spanned (each one being considered in a different candidate joint action of

the Action Subspace). By so doing, the Reduced Joint Action Space includes SN candidate joint actions: this certainly entails a drastic reduction with respect to the S^N joint actions that would appear in the entire joint action space $\mathbf{A}$.

For instance, if, at the considered time step, $N = 4$ (i.e., the Agents 1, 2, 3 and 4 are active) and $S = 3$ (i.e., the action a_i that Agent i, for $i = 1, 2, 3, 4$, can perform corresponds to the selection of one of the three CoSs c_1, c_2, c_3), each of the $SN = 12$ rows of the table in Fig. 1.2 provides one of the 12 candidate joint actions (in particular, the sub-tables included within the borders in bold identify the $N = 4$ Action Subspaces), while each of the four columns of the table identifies the single actions that can be taken by Agents 1, 2, 3 and 4, respectively, in the overall Reduced Joint Action Space.

Moreover, every time a new Agent, say agent N, dies (i.e., an in-progress application terminates), the Reduced Joint Action Space is updated by eliminating the actions involving Agent N. For instance, referring to the example reported in the table in Fig. 1.2, if Agent 4 dies, the three joint actions corresponding to the three last rows are removed (i.e., the Action Subspace corresponding to Agent 4 is removed), and all the actions corresponding to the last column are removed, too.

c_1	a_2^* [1,2]	a_3^* [1,3]	a_4^* [1,4]
c_2	a_2^* [1,2]	a_3^* [1,3]	a_4^* [1,4]
c_3	a_2^* [1,2]	a_3^* [1,3]	a_4^* [1,4]
a_1^*[1,2]	c_1	a_3^* [2,3]	a_4^* [2,4]
a_1^*[1,2]	c_2	a_3^* [2,3]	a_4^* [2,4]
a_1^*[1,2]	c_3	a_3^* [2,3]	a_4^* [2,4]
a_1^* [1,3]	a_2^* [2,3]	c_1	a_4^* [3,4]
a_1^* [1,3]	a_2^* [2,3]	c_2	a_4^* [3,4]
a_1^* [1,3]	a_2^* [2,3]	c_3	a_4^* [3,4]
a_1^* [1,4]	a_2^* [2,4]	a_3^* [3,4]	c_1
a_1^* [1,4]	a_2^* [2,4]	a_3^* [3,4]	c_2
a_1^* [1,4]	a_2^* [2,4]	a_3^* [3,4]	c_3

Figure 1.2: Representation of the Reduced Joint Action Space for $N = 4$ and $S = 3$. The columns of the table identify the different Agents, the rows represent the different candidate joint actions, and the sub-tables within the borders in bold represent the so-called Action Subspaces.

H-MARL-Q Algorithm Description: Step (b)

Step (b) of the H-MARL-Q algorithm is performed on the basis of the MARL-Q algorithm presented above and is applied to the Reduced Joint Action Space identified in Step (a). So, in Step (b), the QoE Controller has to perform the tasks T1, T2, T3, and T4, with the fundamental difference that, when performing tasks T1 and T2, the Reduced Joint Action Space (having cardinality SN), instead of the entire Joint Action Space (having cardinality S^N), is considered. Since N can be in the order of thousands, it is evident that the proposed approach drastically reduces the required computing power.

1.1.4 H-MARL-Q Algorithm Simulations

Simulation Scenario

This section presents numerical simulations, carried out using MATLAB, with reference to a simple simulation scenario which does not claim to represent any real network. The presented simulations are just aimed at providing a proof-of-concept of the proposed algorithm in order to highlight its potentialities and criticalities.

We assume the presence of $S = 3$ different CoSs (e.g., guaranteed, premium and best effort services) and $M = 3$ different application types (i.e., real-time HDTV streaming, distributed videoconferencing and simple File Transfer Protocol). The static CoS assignment policy determines a static association among application types and CoSs (i.e., an application instance belonging to a given application type is assigned the corresponding CoS for its entire lifetime), whereas in the dynamic CoS assignment case, at each time step t_k, an application instance can be assigned any CoS (regardless of the application type) according to the proposed H-MARL-Q algorithm.

We assume that, during our simulations, N Agents are active, each one being involved in an application instance. Such an application instance may belong to one of the three considered application types and is characterized by an average offered transmission bitrate b_i randomly selected in the set $\{0.6, 1.2, 2\}$ and by a personalized Target QoE $TQoE_i$ (for $i = 1, \ldots, N$) randomly selected in the set $\{0.7, 0.8, 0.9\}$.

The simulated network has a dumbbell network topology, as shown in Fig. 1.3, where each of the N transmitters corresponds to one of the N considered Agents. Router West implements a Weighted Fair Queueing (WFQ) scheduler for handling the traffic to be transmitted over the bottleneck link. The related WFQ vector [50] is assumed to be $(0.5, 0.3, 0.2)$, where the i-th element is the weight assigned to the i-th CoS (higher weight means higher priority). The bottleneck link is characterized by an available link capacity B_{link} computed

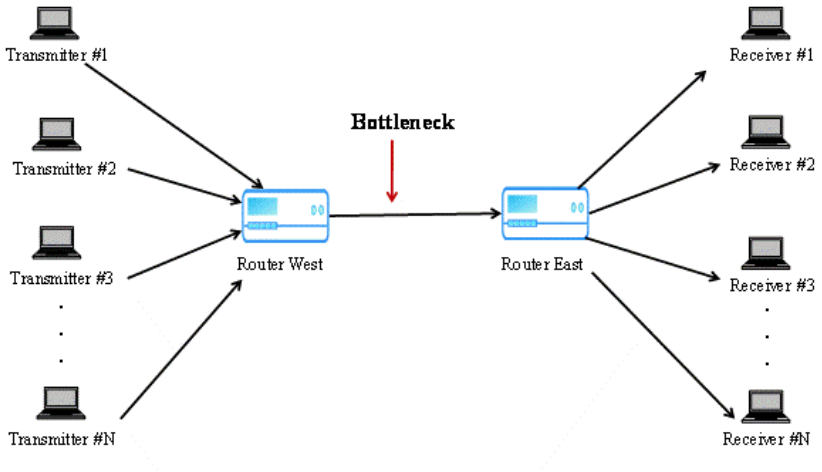


Figure 1.3: Dumbbell network topology.

as:

$$B_{link} = \omega \sum_{i=1}^{N} b_i, \tag{1.7}$$

where ω is a parameter in the range $(0, 1)$ accounting for traffic congestion; in particular, in our simulations we consider two different situations characterized by $\omega = 0.7$ and $\omega = 0.8$, which represent High Traffic and Medium Traffic conditions, respectively.

As for the number of active Agents N, in our simulations we consider two cases: $N = 100$ and $N = 1000$. For each of these two cases and for each of the two considered traffic congestion conditions, ten simulation runs or episodes have been carried out, with a duration of (15×10^3) time steps for $N = 100$ and of (15×10^4) time steps for $N = 1000$: in each simulation run a different association among application instances, application types, average offered bitrates and Target QoE values is performed. Such associations are assumed to be fixed for the entire simulation run.

In the simple proposed simulation scenario, we assume that the set of Feedback Parameters ϕ_m includes, for any $m = 1, 2, 3$, just a single element denoted as ϕ_{QoS} and that the function g_m, introduced in (1.2), is computed on the basis of the well-known IQX hypothesis [64]. This means that (1.2) becomes:

$$PQoE_{i(m)}(t_k) = p_m \cdot e^{-\sigma_m \phi_{QoS}} + \tau_m, \tag{1.8}$$

where the parameter ϕ_{QoS} has been assumed to be equal to the difference

between the traffic offered by the application instance and the corresponding bitrate currently allocated by the WFQ Scheduler. Note that the latter parameter depends on the CoS appointed at time t_k for the considered application instance, which actually impacts on the priority assigned by the WFQ Scheduler to the packets of the relevant traffic flow. We assume $\sigma_1 = 0.5, \sigma_2 = 0.7, \sigma_3 = 1$, as well as $p_m = 1$ and $\tau_m = 0$ for all values of m; with these choices, $PQoE_{i(m)}(t_k)$ is always included in the range $[0, 1]$. The learning rates $\alpha(t_k)$ appearing in (1.4), according to [120], are set to:

$$\alpha(t_k, a_1, a_2, \ldots, a_N) = \frac{1}{1 + visit(t_k, a_1, a_2, \ldots, a_N)}, \tag{1.9}$$

where $visit(t_k, a_1, \ldots, a_N)$ is the number of times that a specific joint action $(a_1, a_2, \ldots, a_N)$ has been enforced up to the iteration at time t_k. The discount rate is set to $\gamma = 0.9$. The selected joint reward function, consistent with the general criteria identified above, is:

$$r(t_k, a_1, a_2, \ldots, a_N) = \sum_{i=1}^{N} w_i(t_k), \tag{1.10}$$

where the absolute value of w_i serves as an appropriately chosen penalty, which the i-th Agent is inflicted with, any time it exhibits either underperforming or overperforming behaviour. A proper choice of w_i may be the following:

- $w_i(t_k) = -100$ if $e_i(t_k) < -0.15$ (i.e., if severe underperformance is experienced by Agent i);
- $w_i(t_k) = -10$ if $-0.15 \leq e_i(t_k) < 0$ (i.e., if minor underperformance is experienced by Agent i);
- $w_i(t_k) = -1$ if $0 \leq e_i(t_k) < 0.1$ (i.e., if acceptable overperformance is experienced by Agent i);
- $w_i(t_k) = -50$ if $e_i(t_k) \geq 0.1$ (i.e., if undesirable overperformance is experienced by Agent i).

In particular, the thresholds on the QoE Error values listed above have been arbitrarily chosen in order to suitably classify the behaviour of Agent i at time t_k as a result of the joint action taken. Moreover, the initial policy, that is, the initial CoS-to-application association, is randomly generated.

Note that, even though the proposed proof-of-concept does not claim to represent any real network, a bottleneck link characterized by limited available bandwidth capacity can represent the uplink of a given cell of a cellular network. In such a scenario, a number of Agents roaming in the considered cell (and hence sharing the cell available uplink capacity) in the order of some hundreds (as assumed in this section) seems reasonable.

1.1.5 Numerical Results

This Subsection shows the results obtained in the described simulation scenario; in particular, the H-MARL-Q algorithm is applied with a number of Agents $N = 100$ and $N = 1000$, both in the High and Medium Traffic conditions.

It should be emphasized that we can deal with such a high number of Agents due to the fact that the proposed H-MARL-Q algorithm relies on a Reduced Joint Action Space, which has cardinality $SN = 300$ in the scenario with 100 Agents ($S = 3$ and $N = 100$), and $SN = 3000$ in the scenario with 1000 Agents ($S = 3$ and $N = 1000$). If the original Joint Action Space were used, a solution relying on the MARL-Q algorithm would be unfeasible, since the cardinality would be $S^N = 3^{100} = 5.2 \cdot 10^{47}$, and $S^N = 3^{1000} = 1.42 \cdot 10^{477}$ in the two scenarios, respectively.

The results obtained with the H-MARL-Q algorithm are compared with the performance of a Static algorithm which adopts a static CoS assignment policy. The comparison with the MARL-Q algorithm is impossible due to the curse of dimensionality (as explained above). The obtained results are expressed in terms of two quantities:

(i) the *Average Absolute QoE Error*, computed as the absolute value of the QoE Error expressed by (1.1), averaged over all the considered Agents and all the simulation episodes (see Figs. 1.4 and 1.5);

(ii) the *QoE Error Standard Deviation*, computed as the standard deviation of the QoE Error vector $(e_1, e_2, \ldots, e_N)$ (where e_i, for $i = 1, 2, \ldots, N$, is expressed as in (1.1) averaged over all the simulation episodes (see Figs. 1.6 and 1.7).

Note that the standard deviation accounts for the fairness among Agents: the smaller the standard deviation, the higher the fairness among Agents. Figs. 1.4-1.7 clearly show that the H-MARL-Q algorithm remarkably outperforms the Static algorithm in all of the considered simulation cases. In particular, while under the Static algorithm the Average Absolute QoE Error is appreciably smaller in Medium rather than in High Traffic conditions, under the H-MARL-Q algorithm, for both $N = 100$ and $N = 1000$, the Average QoE Error bars corresponding to High and Medium Traffic conditions (see Figs. 1.4 and 1.5) exhibit values that are really close to each other: this means that the presented algorithm also allows to overcome the disadvantages related to the impact that the traffic congestion conditions produce on the bottleneck link.

Furthermore, the QoE Error Standard Deviation shown in Figs. 1.6 and 1.7 confirms the virtues of the H-MARL-Q algorithm, since the dispersion

of the QoE Error values of the different Agents at the end of the learning procedure is significantly closer to zero than in the case when the Static algorithm is applied.

All these results evidently show that the dynamic and personalized selection of the most appropriate CoS for the ongoing application instances yields improved performance results, if compared with a static CoS assignment policy. In addition, Fig. 1.8 shows the Average Absolute QoE Error trend, i.e., the evolution of the Average Absolute QoE Error over time.

Let the *settling time* denote the time needed by the Average Absolute QoE Error to reach a steady state. Once an acceptable preliminary agreement among Agents – yielding the selection of the most "promising" joint actions for solving the dynamic CoS assignment problem – has been reached in Step (a), the error dynamics, as highlighted in Fig. 1.8, experiences a rapid decrease over the first 100 iterations of Step (b) and then it takes some time to settle down to the steady-state value: in the figure, the settling time is approximately equal to 9000 iterations. So, the overall runtime required by the H-MARL-Q algorithm is the sum of the time t_a necessary to reach the preliminary agreement in Step (a) plus the time t_b necessary to perform Step (b), where t_b amounts to approximately 9000 iterations for $N = 100$ and t_a is negligible with respect to t_b. This is indeed an encouraging result which shows that the H-MARL-Q algorithm has to be preferred to the MARL-Q algorithm as the former achieves a satisfactory approximate solution in a reasonably smaller amount of runtime than the latter – whose runtime, instead, actually turns out to be unfeasibly long in scenarios where the number of Agents is counted in the order of hundreds or thousands.

The proposed approach to QoE Control enables a dynamic Class of Service selection aimed at reducing the error between the personalized Perceived QoE and the personalized Target QoE levels by properly driving the control procedures that handle the underlying networks. This result could be obtained by embedding an innovative Multi-Agent Reinforcement Learning algorithm, namely the proposed H-MARL-Q algorithm, in a centralized QoE Controller. Such an algorithm has been tested in a simple simulation scenario, with just the aim of providing a proof-of-concept and without claiming to represent any real network.

In conclusion, the proposed method presents several practical advantages:

(i) it does not require any *a-priori* knowledge of the environment (i.e., it is model-free) thanks to the adoption of a Reinforcement Learning based approach;

(ii) it is decoupled from QoE Evaluation, i.e., it can work in conjunction with any algorithm computing the Target QoE and the Perceived QoE

values, and it allows a personalization level up to the single application instance, since the only signal exchanged at the interface between the QoE Controller and the QoE Evaluator is the QoE Error provided by 1.1);

(iii) it requires minimal signalling overhead since no communication exchange among Agents is needed and very little information has to be exchanged among the centralized QoE Controller and the distributed Agents;

(iv) it is characterized by a very good degree of scalability (thus being able to handle several hundreds of Agents) due to the fact that, as the joint action to be carried out at each time step is sought within a suitable Reduced Joint Action Space, the complexity of the proposed H-MARL-Q algorithm is linear in the number of Agents (as opposed to the well-known MARL-Q algorithm whose complexity is exponential in the number of Agents).

Note that the algorithm presented in this Section assumes the time-invariance of the Target QoE. However, we are carrying out further studies, based on concept drift in web/telecommunication systems [187], so as to address also the case of a time-varying Target QoE. In this last case, the Target QoE depends not only on the commercial profile of the users but also on the relevant feedbacks provided by the users themselves. Moreover, we are presently carrying out further research based on a combinatorial multi-armed bandit approach to cooperative online learning [71] [181], with the aim of overcoming the centralized paradigm and, consequently, of developing a solution in which the QoE Control functionalities can be fully distributed into the Agents. Finally, note that the overall modular architecture sketched in Fig. 1.1 – within which Reinforcement Learning algorithms embedded in a QoE Controller play the role of dynamically selecting (on the basis of real-time feedbacks provided by a proper QoE Evaluator) appropriate Reference Values which should drive environment-specific procedures – has proved to be so flexible that the authors are reproducing it also in the domains of intelligent transport systems [28] and telemedicine within the framework of EU-funded research projects.

1.2 Discrete-Time Load Balancing Converging to the Wardrop Equilibrium

With respect to the research field of congestion control – e.g., [42], [44] and [30] are authoritative references, just to name a few – this Section presents a

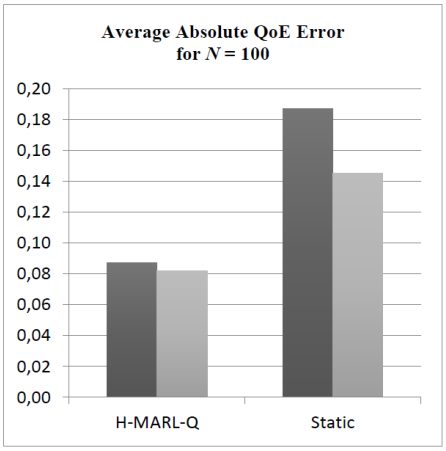


Figure 1.4: Average Absolute QoE Error for $N = 100$. The dark-grey bar and the light-grey bar represent the Average Absolute QoE Error in High and Medium Traffic conditions, respectively.

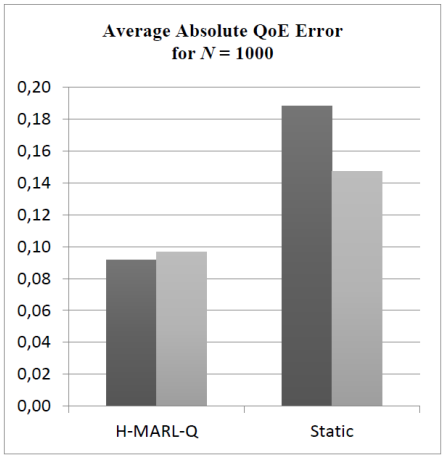


Figure 1.5: Average Absolute QoE Error for $N = 1000$. The dark-grey bar and the light-grey bar represent the Average Absolute QoE Error in High and Medium Traffic conditions, respectively.

discrete-time algorithm for load balancing, proving the convergence to a Wardrop equilibrium, which is a mean field game theoretical concept, originally introduced for network games when modelling transportation networks with congestion [178], and which can be informally described as the situation when "the journey times on all the routes actually used are equal, and less than those

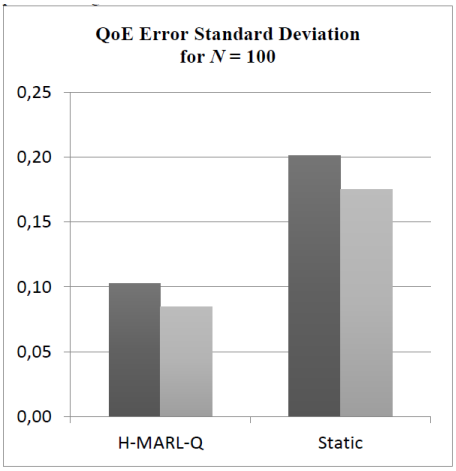


Figure 1.6: QoE Error Standard Deviation for $N = 100$. The dark-grey bar and the light-grey bar represent the QoE Error Standard Deviation in High and Medium Traffic conditions, respectively.

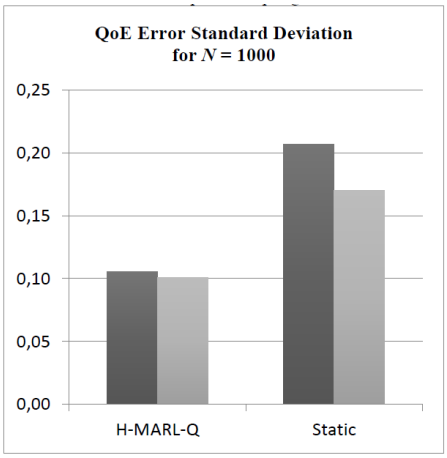


Figure 1.7: QoE Error Standard Deviation for $N = 1000$. The dark-grey bar and the light-grey bar represent the QoE Error Standard Deviation in High and Medium Traffic conditions, respectively.

which would be experienced by a single vehicle on any unused route" [40].

Via Lyapunov arguments, the algorithm is proved to asymptotically converge to a specific equilibrium condition among the traffic loads over the network paths, known as *Wardrop equilibrium*. This convergence result improves the discrete-time algorithms in the literature, which achieve approx-

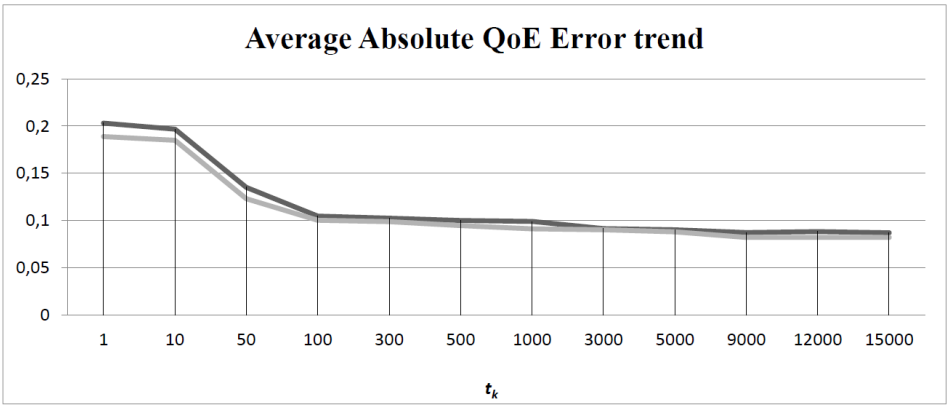


Figure 1.8: Average Absolute QoE Error trend, corresponding to Step (b) of the H-MARL-Q algorithm, in High (black line) and Medium (grey line) Traffic conditions with $N = 100$.

imate convergence to the Wardrop equilibrium. Numerical simulations show the effectiveness of the proposed approach.

In load balancing scenarios, typically characterized by a distributed network of computing hosts, the system performance crucially depends on effectively dividing up work across the admissible network paths [7]. We therefore refer to the model of a time-invariant communication topology where a certain amount of traffic load, or *flow demand*, has to be routed from a source node to a destination node over a set of admissible paths in such a way as to ensure that the network stays *balanced*, i.e., the used network paths yield minimal latencies. In particular, network traffic is modelled as an infinite stream of infinitely-many arriving agents, each being responsible for an infinitesimal amount of traffic, or *job*. Each agent is then a decision maker, yielding the *distributed* nature of the modelling setup.

Non-cooperative algorithms entail the presence of several decision makers which optimize their own response time independently of the others, since cooperation is not allowed. In case a finite number of agents is considered, a Nash equilibrium condition is reached when no agent can receive any further benefit by changing its own decision unilaterally. In other words, the stability of the network under said algorithms is analysed in terms of reaching a load distribution in which no single agent can move to any other path with a lesser number of jobs. In case the number of agents is infinite, a combination of flows such that no agent can improve its latency by deviating unilaterally yields a Wardrop equilibrium for the network. Indeed, a Nash equilibrium is said to become a Wardrop equilibrium whenever the number of agents is

assumed to be infinite [93]. In turn, *dynamic* algorithms are required for settings where the load distribution is not known *a priori* as they succeed in performing the decision-making process based on the current state of the system, which is generally made available via feedback.

In this respect, we propose a discrete-time, distributed, non-cooperative, dynamic load balancing algorithm, designed so that a Wardrop equilibrium is reached by the agents. Up to the authors' knowledge, the proposed algorithm is the first discrete-time load balancing algorithm which is proved to asymptotically converge to the exact Wardrop equilibrium.

1.2.1 Related Work and Proposed Innovation

The considered problem falls within the scope of *mean field game theory*, which provides a fruitful modelling framework for many applications, e.g., in transport, network engineering and computer science problems, and currently represents a very active area of research [13]. Namely, Wardrop equilibria have been relied upon in order to deal with several different types of congested environments, such as: routing in road traffic networks [35] [39], traffic engineering in communication networks [65], load balancing in distributed computational grids [78] and in wired and wireless networks [6] [134] [135]. In the classical problem formulation [11], a congested network represented by a graph with nodes and edges is considered. A non-decreasing latency function of traffic is associated with each edge, representing the cost of the edge (e.g., its congestion level). The network serves several commodities, characterized by a given amount of traffic load to be routed from a source to a destination and balanced over the set of admissible network paths. The job vector is the amount of traffic that is allocated for each commodity and for each path connecting the (source, destination) couples. Each agent has the possibility to distribute its own flow among a set of admissible paths.

A job vector in which, for all commodities, the latencies of all used paths are equal is called Wardrop equilibrium. The Wardrop equilibrium can be computed by centralized algorithms in polynomial time [14]. The proposed algorithm is aimed at achieving a Wardrop equilibrium via a discrete-time dynamic and distributed algorithm.

Several load balancing approaches have been proposed in the literature: in [80] it is suggested to classify load balancing algorithms as either static or dynamic. Static load balancing relies on the available knowledge of the application load, whereas dynamic load balancing algorithms are required for settings where the load distribution is known based on the current state of the system, which is generally made available via feedback.

From the large body of literature on load balancing, we recall [97] and [172]

as examples of centralized static cooperative load balancing, [79] and [172] as examples of centralized static non-cooperative load balancing, [59] and [60] as examples of centralized dynamic load balancing, and we also recall [166], which, instead, addresses the problem of distributed dynamic load balancing relying upon local cooperation among neighbouring network nodes.

The scenario considered here requires a non-cooperative dynamic load balancing approach. This kind of algorithms are widely investigated in game theoretic frameworks, where the problem can be described as a dynamic load balancing game, with users distributing their loads in a non-cooperative and selfish fashion [1] [146].

Furthermore, such a setup can be adapted to selfish routing problems: in this respect, quite a few works in the literature resort to a similar approach as the one proposed here, such as [15] and [101] relative to a continuous-time setting, and, instead, [18], [66], [67], and [68] relative to a discrete-time setting where the authors only prove convergence to an approximated neighbourhood of the Wardrop equilibrium. In [66], the authors develop a round-based distributed algorithm with a finite number of agents. Each agent is responsible for one commodity and has a set of admissible paths among which it may distribute its traffic. In this respect, the population of agents responsible for rerouting the traffic is instructed to *learn* a Wardrop equilibrium efficiently by relying on suitable adaptive sampling methods. A bulletin board is assumed to be available, where the traffic assignments are updated at every round. With a similar bulletin board scenario, in [67] a distributed routing algorithm is presented. At each round, each agent samples a different path and compares the latency of the newly chosen path with its own current latency. If the comparison shows that the agent can improve its latency by rerouting its own portion of network traffic, then it migrates to the better path with some probability depending on the latency improvement. Convergence results are given both when the agents base their decisions on up-to-date information and when the information is stale (i.e., considering delays in the bulletin board update). In [68] a round-based version of the adaptive routing algorithm with stale information shown in [67] is proposed. In [18], a distributed and asynchronous routing algorithm is proposed, relying on an estimation of the latencies of all paths and on a reinforcement learning algorithm to update the probabilities of transmission towards the different paths. All the cited algorithms converge to an approximate Wardrop equilibrium. Yet, none of these works yields exact convergence to a Wardrop equilibrium.

The main innovation of the proposed algorithm is therefore the design of a discrete-time load balancing algorithm converging to an exact Wardrop equilibrium. In detail, the proposed algorithm is designed so as to dynamically learn a Wardrop equilibrium efficiently and in a distributed fashion. We adopt

the problem formulation proposed in [6] and rely on the algorithm proposed in [68]. The convergence proof is given by using Lyapunov's second method.

1.2.2 Load Balancing Problem

As anticipated, this Section further develops a well-known model for *selfish routing* [68], where an infinite population of agents carries an infinitesimal amount of load each, with the aim of addressing a load balancing scenario and following the previous works [136] and [38] concerning distributed load balancing algorithms. Let $\mathcal{I}$ denote a set of commodities with flow demands $d^i > 0,\ \forall i \in \mathcal{I}$, generally expressed in jobs per unit of time, with total demand $d := \sum_{i\in\mathcal{I}} d^i$. Let also $\mathcal{P}^i$ denote a set of paths (or providers), which serve the traffic flows for every commodity $i \in \mathcal{I}$, and let $\mathcal{P} = \cup_{i\in\mathcal{I}}\mathcal{P}^i$ be the set of all the network paths. As an example, we may think of the considered model as a description of a network consisting of a set of edges, over which the adopted control law arranges proper paths to connect source and destination nodes. Namely, each source node (e.g., s^i for some $i \in \mathcal{I}$) is connected by the network to the destination node (i.e., t^i) through a set of paths denoted with $\mathcal{P}^i$.

The definition of *agent* is also required. As defined, for instance, in (Barth et al., 2008), each agent is an infinitesimal portion of a specified commodity. Let x_p^i be the volume of the agents, or bandwidth, of commodity i relying on a path $p \in \mathcal{P}^i$. In the considered scenario, the vector $\mathbf{x} = (x_p^i)_{p\in\mathcal{P}^i,\ i\in\mathcal{I}}$ can be defined as the *flow vector* or *population share*, whose components specify the overall amount of traffic per unit of time flowing along path $p \in \mathcal{P}^i$ and associated with commodity $i \in \mathcal{I}$. Let $x_p := \sum_{i\in\mathcal{I}} x_p^i$ denote the total traffic flow over path $p \in \mathcal{P}$.

Definition 1.1. The feasible state space, i.e., the closed set of feasible flow vectors, is

$$\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^{|\mathcal{P}|\times|\mathcal{I}|} \mid x_p^i \geq 0,\ \forall p \in \mathcal{P}^i,\ \sum_{p\in\mathcal{P}^i} x_p^i = d^i,\ \forall i \in \mathcal{I}. \tag{1.11}$$

□

A metric of interest is the average response time required by the path (or provider) $p \in \mathcal{P}$ for serving an amount of traffic load equal to x_p. The response time grows with the considered load and thus it is a reliable indicator of the path congestion status. Hence, this quantity is defined as the *latency function* associated with the path $p \in \mathcal{P}$ which is a non-negative function $l_p(x_p) :\ [0, d] \to \mathbb{R}_{\geq 0}$. The shape of the latency functions depends on the

considered application. One strength of the proposed approach is that there is no need for explicitly modelling such latency functions.

The latency of a path $p \in \mathcal{P}$ is therefore a function of its load x_p, i.e., $l_p(x_p)$ is the latency of path p with load x_p. The latency functions are only assumed to have the following properties.

Assumption 1.1. The latency functions $l_p(\xi), p \in \mathcal{P}$, are Lipschitz continuous and strictly increasing over the interval $[0, d]$. We also define β_p as the local Lipschitz constant of l_p, and $\beta_{max} := \max_{p \in \mathcal{P}} \beta_p$.

□

Assumption 1.1 is a reasonable restriction, since the response time of a path (or provider) generally increases with the total amount of traffic load routed onto that path.

The agents' aim is that of minimizing their personal latency selfishly without considering the impact on the global situation. One usually assumes that the agents will converge to some allocation in which no agent can improve its latency by deviating unilaterally. A useful characterization of such an equilibrium goes back to Wardrop. In particular, a single request of the flow is approximately considered as an agent: in fact, even if the number of requests is finite, if the flow rates are sufficiently high, the population acceptably approximates the infinite population constraint required by Wardrop theory [178].

The load balancing problem is formulated below as the problem of determining the strategies which will lead the flow vector to reach a Wardrop equilibrium. In Wardrop theory, *stable* flow assignments are the ones in which no agent (i.e., no "small" portion of a commodity directed from a source to a destination) can improve its situation by changing its strategy (i.e., the set of used paths) unilaterally. This objective is achieved if all agents reach a Wardrop equilibrium:

Definition 1.2. [66] A feasible flow vector $\mathbf{x}$ is at a Wardrop equilibrium for an instance of the considered load balancing game if, for each path $p \in \mathcal{P}^i$ such that $x_p^i > 0$, the following relation holds: $l_p(x_p) \leq l_q(x_q),\ \forall q \in \mathcal{P}^i,\ \forall i \in \mathcal{I}$.

□

In practice, at the Wardrop equilibrium, the latencies of all the loaded paths have the same value: therefore, provided that the latency functions properly represent the path performances, a fair exploitation of the network resources is achieved by driving the flows towards a Wardrop equilibrium.

In the framework of researches on Wardrop equilibria, a key role is played by the Beckmann, McGuire, and Winsten potential [14], given by:

$$\Phi(\mathbf{x}) := \sum_{p \in P} \int_0^{x_p} l_p(\xi) d\xi, \tag{1.12}$$

whose properties are summarised in Property 1.1 below.

Property 1.1. (Fischer and Vocking, 2009). Under Assumption **??**, the potential (1.12) is continuous and the following properties hold:

i) there exists a unique flow $\mathbf{x}_{min}$, over the set of feasible flows, minimizing Φ;

ii) correspondingly, there exists a unique positive minimum $\Phi_{min} = \Phi(\mathbf{x}_{min})$;

iii) in $\mathbf{x}_{min}$, no agent can improve its own latency unilaterally, i.e., $\mathbf{x}_{min}$ is at Wardrop equilibrium.

□

Property 1.1 states that, in the considered scenario, the set $\mathcal{X}_W$ collapses into a unique Wardrop equilibrium with job vector $\mathbf{x}_{min}$, hereafter denoted with $\mathbf{x}_W$.

1.2.3 Discrete-Time Control Law and Algorithm Convergence

For the sake of presentation clarity, we choose to limit the analysis to the single-commodity case only, i.e., $|\mathcal{I}| = 1$, and, therefore, the index i will be neglected.

Proposed Control Law

The system dynamics is therefore expressed component-wise by

$$x_p[k+1] = x_p[k] + \tau \sum_{q \in \mathcal{P}} (r_{qp}[k] - r_{pq}[k]),\ k = 0, 1, \ldots,\ \forall p \in \mathcal{P}, \tag{1.13}$$

where τ is the sampling period and $r_{pq}[k]$ is the so-called *migration rate* from path $p \in \mathcal{P}$ to path $q \in \mathcal{P}$. Inspired by the algorithm in [66], the migration rate is defined as

$$r_{pq}[k] = x_p[k] \sigma_{pq}[k] \mu_{pq}[k], \tag{1.14}$$

where

$$\sigma_{pq}[k] > 0,\ \forall p, q \in \mathcal{P}, \tag{1.15}$$

is the control gain, which sets the rate with which the population share of path p migrates to path q, and $\mu_{pq}[k]$ is the so-called *migration policy*, representing the decision whether (and in which percentage) the population share assigned to path p migrates to path q. Throughout this Section, we consider the following initial conditions:

$$\begin{cases} x_p[0] \geq 0, \ \forall p \in \mathcal{P}, \\ \sum_{p \in \mathcal{P}} x_p[0] = d. \end{cases} \tag{1.16}$$

There is a variety of migration policies used in the literature for continuous-time algorithms, e.g., the *better response* policy

$$\mu_{pq}[k] = \begin{cases} 0 & \text{if } \ l_p(x_p[k]) - l_q(x_q[k]) \leq 0, \\ 1 & \text{otherwise,} \end{cases} \tag{1.17}$$

and the *linear migration* policy

$$\mu_{pq}[k] = \begin{cases} 0 & \text{if } \ l_p(x_p[k]) - l_q(x_q[k]) \leq 0, \\ \frac{l_p(x_p[k]) - l_q(x_q[k])}{l_{max}} & \text{otherwise,} \end{cases} \tag{1.18}$$

where l_{max} is the maximum latency value. By using these migration policies, it can be shown that convergence cannot be guaranteed, however small the sampling time τ is chosen (see, e.g., [67]). The proposed migration policy is then a modified better response policy, defined as:

$$\mu_{pq}[k] = \begin{cases} 0 & \text{if } \ l_p(x_p[k]) - l_q(x_q[k]) \leq \alpha\delta[k], \\ 1 & \text{otherwise,} \end{cases} \tag{1.19}$$

where $\alpha \in (0, 1)$ is a tuning parameter which affects the algorithm convergence velocity, as examined below, and $\delta[k]$ is defined as the difference between the maximum measured latency value of the loaded paths and the minimum measured latency value, at time k, i.e.,

$$\delta[k] := \max_{p \in \mathcal{P} | x_p[k] > 0} l_p(x_p[k]) - \min_{q \in \mathcal{P}} l_q(x_q[k]). \tag{1.20}$$

The proposed control law is expressed by equations (1.14), (1.15) and (1.19), with the following control gain upper bound:

$$\sigma_{pq}[k] \leq \frac{\alpha\delta[k]}{\tau\beta_{max}d|\mathcal{P}|}, \tag{1.21}$$

The algorithm (1.13), (1.14), (1.15), (1.19), (1.20), and (1.21) resembles the algorithm in [66]; besides the different expression (1.21) for $\sigma_{pq}[k]$, the main difference is the tolerance $\alpha\delta[k]$ introduced in equation (1.19).

The system dynamics (1.13), under the control law (1.13), (1.14), (1.15), (1.19), (1.20), and (1.21), yields, in vector form, the closed-loop system dynamics

$$\mathbf{x}[k+1] = f(\mathbf{x}[k]), \quad \mathbf{x}(0) \in \mathcal{X}, \tag{1.22}$$

with respect to which we conduct Lyapunov stability analysis.

Convergence Proof.

Lemma 1.1. *Under Assumption 1.1, relying on the algorithm (1.13), (1.14), (1.15), (1.19), (1.20), and (1.21), with initial conditions (1.16), the latency variation of l_p in time steps is bounded according to the following relation:*

$$-\frac{(|\mathcal{P}|-1)}{|\mathcal{P}|}\alpha\delta[k] \leq l_p(x_p[k+1]) - l_p(x_p[k]) \leq \frac{1}{|\mathcal{P}|}\alpha\delta[k]. \tag{1.23}$$

Proof. In the worst case, at time k, no paths migrate their own population to path p:

$$\begin{aligned} l_p(x_p[k+1]) &= l_p\Bigg(x_p[k] + \tau \sum_{m\in\mathcal{P}|m\neq p} (r_{mp}[k] - r_{pm}[k])\Bigg) \geq \\ &\geq\ l_p\Bigg(x_p[k] - \tau \sum_{m\in\mathcal{P}|m\neq p} r_{pm}[k]\Bigg). \end{aligned} \tag{1.24}$$

Since β_p is the Lipschitz constant of the latency function l_p, it follows that

$$l_p(x_p[k+1]) \geq l_p(x_p[k]) - \beta_p\tau \sum_{m\in\mathcal{P}|m\neq p} r_{pm}[k]. \tag{1.25}$$

Considering equations (1.14), (1.19), and (1.21), the last term of equation (1.25) is written as

$$\begin{aligned} &\beta_p\tau \sum_{m\in\mathcal{P}|m\neq p} r_{pm}[k] = \\ &= \beta_p\tau \sum_{m\in\mathcal{P}|m\neq p} \sigma_{pm}[k]\mu_{pm}[k]x_p[k] \leq \\ &\leq \beta_p\tau \sum_{m\in\mathcal{P}|m\neq p} \sigma_{pm}[k]x_p[k] \leq \frac{\alpha\delta[k]}{d|\mathcal{P}|} \sum_{m\in\mathcal{P}|m\neq p} x_p[k], \end{aligned} \tag{1.26}$$

where the last inequality holds since $\beta_p \leq \beta_{max}$. Since there are at most $(|\mathcal{P}| - 1)$ terms in the last summation of equation (1.26) and since $x_p[k] \leq d$, it holds that

$$\beta_p \tau \sum_{m \in \mathcal{P} | m \neq p} r_{pm}[k] \leq \frac{\alpha \delta[k]}{|\mathcal{P}|}(|\mathcal{P}| - 1). \tag{1.27}$$

Similarly, in the worst case, at time k, the path q does not migrate its population to any other paths:

$$l_q(x_q[k+1]) \leq l_q(x_q[k]) + \beta_q \tau \sum_{n \in \mathcal{P} | n \neq q} r_{nq}[k]. \tag{1.28}$$

Also,

$$\beta_q \tau \sum_{n \in \mathcal{P} | n \neq q} r_{nq}[k] = \beta_q \tau \sum_{n \in \mathcal{P} | n \neq q} \sigma_{nq}[k] \mu_{nq}[k] x_n[k]. \tag{1.29}$$

Since $\mu_{pq}[k] \leq 1, \ \forall p, q \in \mathcal{P}$, given that $\sum_{n \in \mathcal{P}} x_n[k] = d$, it holds that

$$\beta_q \tau \sum_{n \in \mathcal{P} | n \neq q} r_{nq}[k] \leq \beta_q \tau \sum_{n \in \mathcal{P} | n \neq q} \sigma_{nq}[k] x_n[k] \leq \frac{\alpha \delta[k]}{d|\mathcal{P}|} \sum_{n \in \mathcal{P} | n \neq q} x_n[k] \leq \frac{\alpha \delta[k]}{|\mathcal{P}|}. \tag{1.30}$$

□

It is shown below that the state space $\mathcal{X}$ is a positively invariant set.

Lemma 1.2. *Under Assumption 1.1, $\mathcal{X}$ is a positively invariant set for the nonlinear discrete-time system (1.22) with control law (1.14), (1.15), (1.19), (1.20), and (1.21), with the initial conditions specified in (1.16).*

Proof. It is shown in the following that, since the initial flow vector of the considered system dynamics, $\mathbf{x}[0]$, lies in $\mathcal{X}$, the flow vector $\mathbf{x}[k]$ lies in $\mathcal{X}$ too, i.e., that a) $\sum_{p \in \mathcal{P}} x_p[k] = d$ and b) $x_p[k] \geq 0, \ \forall p \in \mathcal{P}, \ \forall k \geq 0$.

a) It follows from equation (1.13) that

$$\begin{aligned} \sum_{p \in \mathcal{P}} (x_p[k+1] - x_p[k]) &= \tau \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{P}} (r_{qp}[k] - r_{pq}[k]) = \\ &= \tau \left(\sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{P}} r_{qp}[k] - \sum_{q \in \mathcal{P}} \sum_{p \in \mathcal{P}} r_{qp}[k] \right) = 0, \end{aligned} \tag{1.31}$$

and, therefore, that $\sum_{p \in \mathcal{P}} x_p[k] = \sum_{p \in \mathcal{P}} x_p[0] = d, \ \forall k \geq 0$.

b) By induction, since $x_p[0] \geq 0,\ \forall p \in \mathcal{P}$, it is proven below that $x_p[k] \geq 0,\ \forall p \in \mathcal{P},\ \forall k \geq 0$. Assuming that $x_p[k] \geq 0,\ \forall p \in \mathcal{P}$, for a given k, it is then sufficient to prove that

$$x_p[k+1] = x_p[k] + \tau \sum_{q \in \mathcal{P}} (r_{qp}[k] - r_{pq}[k]) \geq 0,\ \forall p \in \mathcal{P}. \tag{1.32}$$

Since $x_p[k] \geq 0$, from equation (1.14) it follows that $r_{pq}[k] \geq 0$. Thus, the following inequality holds (in the worst case, no paths migrate part of their population to path p):

$$x_p[k+1] \geq x_p[k] - \tau \sum_{q \in \mathcal{P}} r_{qp}[k],\ \forall p \in \mathcal{P}. \tag{1.33}$$

A sufficient condition for inequality (1.32) to hold is then

$$x_p[k] - \tau \sum_{q \in \mathcal{P}} r_{pq}[k] \geq 0,\ \forall p \in \mathcal{P}. \tag{1.34}$$

Recalling equations (1.14) and (1.20), equation (1.34) can be written as

$$\begin{aligned} &x_p[k] - \tau \sum_{q \in \mathcal{P}} r_{pq}[k] = x_p[k] - \tau \sum_{q \in \mathcal{P}} x_p[k] \sigma_{pq}[k] \mu_{pq}[k] = \\ &= x_p[k] \left(1 - \tau \sum_{q \in \mathcal{P}} \sigma_{pq}[k] \mu_{pq}[k] \right) \geq 0,\ \forall p \in \mathcal{P}. \end{aligned} \tag{1.35}$$

By substituting the expression of $\sigma_{pq}[k]$ from (1.21), we obtain the condition

$$\begin{aligned} &x_p[k] - \tau \sum_{q \in \mathcal{P}} r_{pq}[k] \geq \\ &\geq x_p[k] \left(1 - \frac{\alpha \delta[k]}{\beta_{max} d |\mathcal{P}|} \sum_{q \in \mathcal{P}} \mu_{pq}[k] \right) > \\ &> x_p[k] \left(1 - \frac{\alpha \delta[k]}{\beta_{max} d} \right) \geq 0,\ \forall p \in \mathcal{P}, \end{aligned} \tag{1.36}$$

where the second inequality holds since, considering that $\mu_{qq}[k] = 0,\ \forall k \geq 0$, the inner summation has at most $(|\mathcal{P}| - 1)$ terms equal to 1. If $x_p[k] = 0$, the inequality (1.36) is verified. Instead, if $x_p[k] > 0$, equation (1.36) holds if

$$\frac{\alpha \delta[k]}{\beta_{max} d} \leq 1. \tag{1.37}$$

Given that $0 < \alpha < 1$, and that, by definition (1.20), $\delta[k] \leq \max_{p\in\mathcal{P}} l_p(x_p[k]) \leq l_p(d)$, it follows that

$$\frac{\alpha\delta[k]}{\beta_{max}d} < \frac{l_p(d)}{\beta_{max}d} \leq 1, \tag{1.38}$$

where the last inequality holds since, by the definition of the β_p's, it holds that $l_p(d) \leq \beta_p d \leq \beta_{max} d$.

□

We are now in a position to prove the following theorem.

Theorem 1.3. *Under Assumption 1.1, $\mathbf{x}_W$ is a globally asymptotically stable equilibrium point for the nonlinear discrete-time system (1.13), (1.14), (1.15), (1.19), (1.20), and (1.21), with initial conditions (1.16) and with total traffic demand $d > 0$.*

Proof. The following proof is based on Lyapunov's direct method. Let $\mathcal{L}(x) := \Phi(\mathbf{x}) - \Phi_{min}$ be the candidate Lyapunov function, where $\Phi(\mathbf{x})$ is the potential (1.12) and Φ_{min} is its minimum value, which is unique thanks to Assumption **??**.

Let $\Delta\mathcal{L}(\mathbf{x}[k]) := \mathcal{L}(\mathbf{x}[k+1]) - \mathcal{L}(\mathbf{x}[k])$ denote the difference of the Lyapunov function $\mathcal{L}(\mathbf{x})$ along the solutions of system (1.13), (1.14), (1.15), (1.19), (1.20), and (1.21).

$$\begin{aligned}
&\Delta\mathcal{L}(\mathbf{x}[k+1]) = \mathcal{L}(\mathbf{x}[k+1]) - \mathcal{L}(\mathbf{x}[k]) = \\
&= \sum_{p\in\mathcal{P}} \int_{x_p[k]}^{x_p[k+1]} l_p(\xi)d\xi \leq \\
&\leq \sum_{p\in\mathcal{P}} (x_p[k+1] - x_p[k]) l_p(x_p[k+1]) = \\
&= \sum_{p\in\mathcal{P}} \tau \left(\sum_{q\in\mathcal{P}} r_{qp}[k] - \sum_{q\in\mathcal{P}} r_{pq}[k] \right) l_p(x_p[k+1]) = \\
&= \tau \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{qp}[k] l_p(x_p[k+1]) - \tau \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{pq}[k] l_p(x_p[k+1]) = \\
&= \tau \sum_{q\in\mathcal{P}} \sum_{p\in\mathcal{P}} r_{pq}[k] l_q(x_q[k+1]) - \tau \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{pq}[k] l_p(x_p[k+1]) = \\
&= \tau \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{pq}[k] (l_q(x_p[k+1]) - l_p(x_q[k+1])),
\end{aligned} \tag{1.39}$$

where the inequality holds from geometric considerations, as the latency functions are strictly increasing.

Indeed, if $x_p[k+1] > x_p[k]$, the upper plot of Figure 1.9 shows that the quantity $(x_p[k+1]-x_p[k])l_p(x_p[k+1])$, represented by the area of the rectangle with bold lines, is larger than the integral $\int_{x_p[k]}^{x_p[k+1]} l_p(\xi)d\xi$, represented by the grey area. If $x_p[k+1] < x_p[k]$, the lower plot shows that the quantity $(x_p[k] - x_p[k+1])l_p(x_p[k+1])$, represented by the area of the rectangle with bold lines, is smaller than the integral $\int_{x_p[k+1]}^{x_p[k]} l_p(\xi)d\xi$, represented by the grey area.

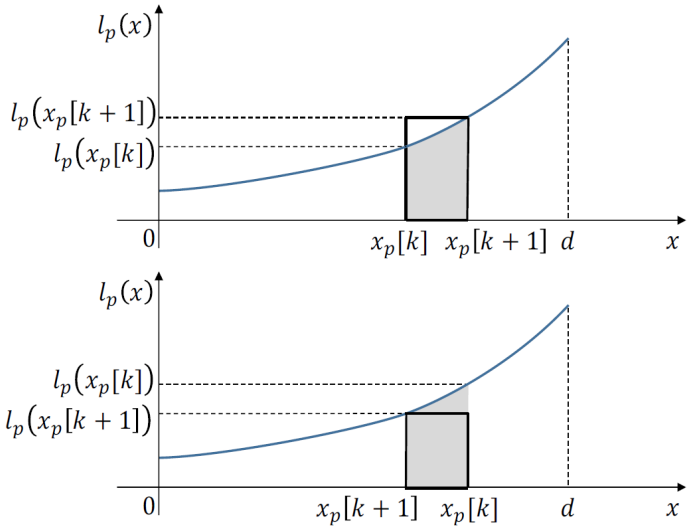


Figure 1.9: Geometrical considerations on the latency functions.

We now prove that the terms $r_{pq}[k](l_q(x_p[k+1]) - l_p(x_q[k+1]))$ of the last summation are either negative or null, for any $p, q \in \mathcal{P}$.

a) If $r_{pq}[k] = 0$, the term is null.

b) It is shown below that, if $r_{pq}[k] > 0$, it holds that $l_p(x_q[k+1]) - l_q(x_p[k+1]) > 0$, i.e., the considered term in (1.39) is negative. By Lemma 1.1, it holds that

$$\begin{aligned} & l_p(x_p[k+1]) - l_q(x_q[k+1]) \geq \\ & \geq \left(l_p(x_p[k]) - \frac{(|\mathcal{P}|-1)}{|\mathcal{P}|}\alpha\delta[k] \right) - \left(l_q(x_q[k]) + \frac{1}{|\mathcal{P}|}\alpha\delta[k] \right) \geq \\ & \geq l_p(x_p[k]) - l_q(x_q[k]) - \alpha\delta[k]. \end{aligned} \tag{1.40}$$

Since we are considering the case $r_{pq}[k] > 0$, it follows from equation (1.19) that

$$l_p(x_p[k]) - l_q(x_q[k]) > \alpha\delta[k], \tag{1.41}$$

which, from equation (1.40), yields that $l_p(x_p[k+1]) - l_q(x_q[k+1]) > 0$.

Now we can prove the asymptotic convergence to $\mathbf{x}_W$ by showing that $\Delta\mathcal{L}(\mathbf{x}[k+1]) = 0$ if and only if $\mathbf{x}[k] = \mathbf{x}_W$.

If $\mathbf{x}[k] = \mathbf{x}_W$, for any couple $p, q \in \mathcal{P}$, it holds either that $l_p(x_p[k]) \leq l_q(x_q[k])$ with $x_p[k] > 0$ or that $x_p[k] = 0$. In the latter case, it follows from equation (1.14) that $r_{pq}[k] = 0$. In the former case, equation (1.19) yields that $\mu_{pq}[k] = 0$ and, therefore, $r_{pq}[k] = 0$. From the discussion above, it follows that $\Delta\mathcal{L}(\mathbf{x}[k+1]) = 0$.

If $\mathbf{x}[k] \neq \mathbf{x}_W$, there exist one or more couples of paths $p, q \in \mathcal{P}$, such that $x_p[k] > 0$ and $l_p(x_p[k]) - l_q(x_q[k]) \geq 0$. Let $p^{'}$ and $q^{'}$ be the paths such that $p^{'} = argmax_{(p\in\mathcal{P} \mid x_p[k]>0)} l_p(x_p[k])$ and $q^{'} = argmin_{q\in\mathcal{P}} l_q(x_q[k])$, respectively, i.e., by definition (1.20), such that $l_{p'}(x_{p'}[k]) - l_{q'}(x_{q'}[k]) = \delta[k]$, with $x_{p'}[k] > 0$. Given that $0 < \alpha < 1$, it holds that $l_{p'}(x_{p'}[k]) - l_{q'}(x_{q'}[k]) > \alpha\delta[k]$ and, from equations (1.14) and ((1.19), that $r_{p'q'}[k] > 0$. From the discussion above, it follows that $\Delta\mathcal{L}(\mathbf{x}[k+1]) < 0$.

□

Remark 1.1. In the control law (1.14), the σ_{pq}'s are interpreted as the control gains. Lemma 1.1 and Theorem 1.3 show that equation (1.21) sets dynamic upper bounds on such control gains, with the twofold objective of keeping the dynamics feasible and of driving the system trajectories towards $\mathbf{x}_W$.

Remark 1.2. Concerning the migration policy in equation (1.19), the key elements for the convergence of the discrete-time case are the facts (i) that, at any time step k, equation (1.19) sets a minimum latency separation $\alpha\delta[k]$ for the migration between two paths, and (ii) that such latency separation vanishes with k, as the latency values converge.

Remark 1.3. As regards the convergence velocity of the considered closed-loop dynamics (1.22), the value chosen for α sets a trade-off between the gain value the upper bound (1.21) on the σ_{pq}'s is proportional to α – and the number of flows selected for migrations – according to equation (1.19), at a given time step k, the μ_{pq}'s are positive only for the path couples whose latencies differ by a quantity larger than $\alpha\delta[k]$. See also the example simulation runs below.

1.2.4 Numerical Simulations

The proposed algorithm is here evaluated via numerical simulations performed with MATLAB. The scenario is composed of $|\mathcal{P}| = 21$ paths, which must support the total traffic $d = 1$. The latency functions are described by the

following equation:

$$l_p(x_p) = \begin{cases} 0.1x_p^2, & p = 1, \dots, 7, \\ 0.2x_p^2, & p = 8, \dots, 14, \\ 0.3x_p^2, & p = 15, \dots, 21. \end{cases} \tag{1.42}$$

Equation (1.42) defines three groups of paths (heterogeneous path case), each one characterized by the same latency function. Accordingly, the local Lipschitz constant is computed as $\beta_p = \max_{\xi \in [0,d]} l_p(\xi)/d\xi = 0.2$, for $p \leq 7$, $\beta_p = 0.4$ for $8 \leq p \leq 14$, $\beta_p = 0.6$ for $p \geq 15$.

With this setup, the value of the latency functions at the Wardrop equilibrium is $l_W = 0.413 \cdot 10^{-3}$ – note that l_W can be analytically computed here only because the latency functions (1.42) are completely known, whereas the proposed algorithm assumes that the functions are unknown. Accordingly, the equilibrium values of the population shares of the paths are $x_p = 0.0642$ for $p \leq 7, x_p = 0.0454$ for $8 \leq p \leq 14, x_p = 0.0371$ for $p \geq 15$. The sampling time is set to $\tau = 0.1$.

The first simulation runs address the tuning of the parameter α of the control action, tehn we show some latency and population dynamics examples as well as a qualitative comparison between the proposed algorithm and the one in [68].

Convergence Time vs. Parameter α.

To evaluate the effect of the parameter α on the convergence time, simulation runs starting from random initial states were simulated for values of α ranging from 0.1 to 0.99. The convergence time is defined as the time step when all the latency values lie close to the value at the equilibrium l_W, within a tolerance $\varepsilon = 0.02$. Fig. 1.10 shows the obtained convergence times, averaged over 20 simulation runs.

In particular, Fig. 1.10 shows that the convergence time is slower for small values of α, decreases with α, due to the fact that the σ_{pq}'s (i.e., the control gains) increase with α, and finally increases again for values of α close to 1, since only the paths with the larger and smaller latency values can exchange their population (see the migration policy in (1.19)). Qualitative examples of the population and latency dynamics are presented.

Examples of population and latency dynamics and comparison with the controller adopted in [68]

We now present an example of simulation run under the proposed algorithm, for different values of α, and under the algorithm in [68]. The latter is a

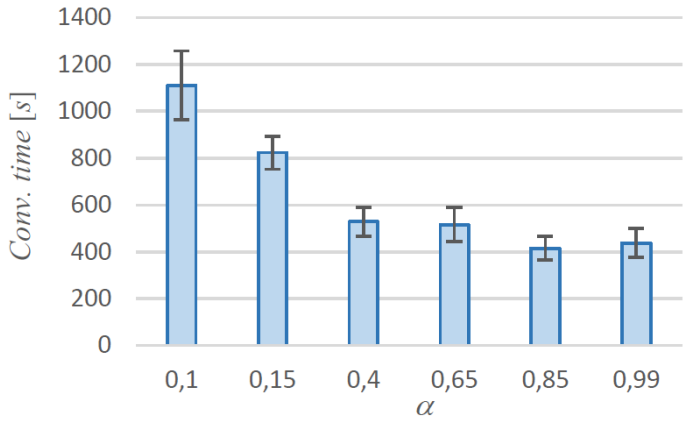


Figure 1.10: Convergence time (mean and std. deviation) vs. α.

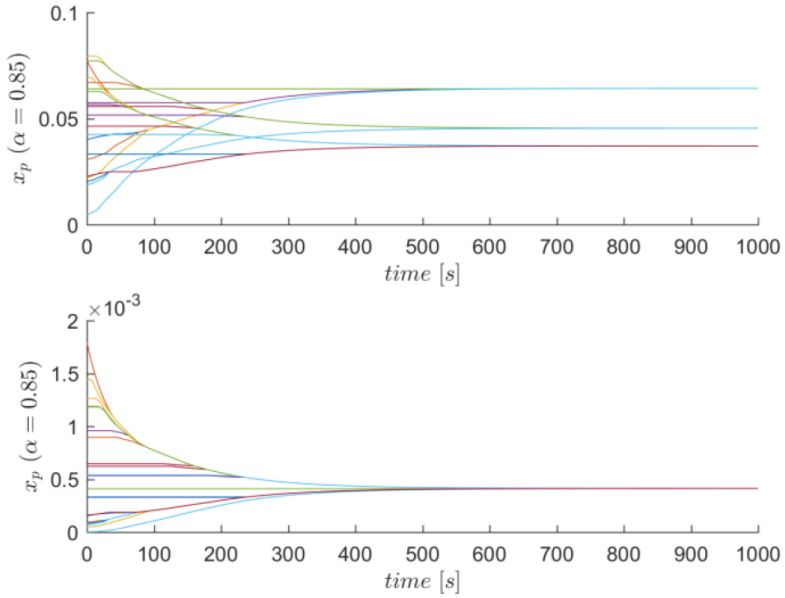


Figure 1.11: Population dynamics (higher plot) and latency dynamics (lower plot) with $\alpha = 0.85$.

round-based algorithm. At every time step k, each path $p \in \mathcal{P}$ is activated with probability γ. Each activated path decides to migrate to a given path

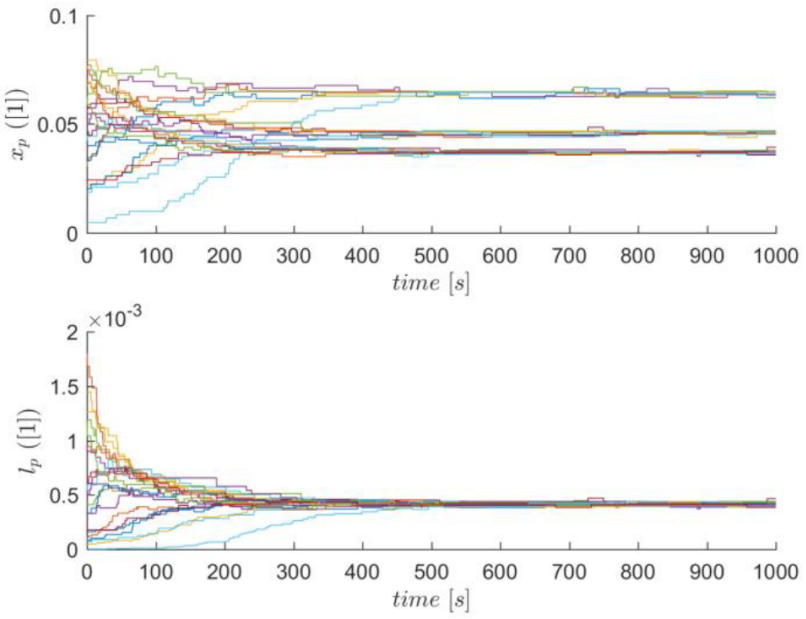


Figure 1.12: Population dynamics (higher plot) and latency dynamics (lower plot) with the algorithm in [68].

$q \in \mathcal{P}$ based on the following rule:

$$Pr\{q \text{ is selected}\} = \begin{cases} \frac{1}{|\mathcal{P}|} & \text{with probability } \beta_F \\ \frac{x_q[k]}{d^i} & \text{with probability } (1-\beta_F), \end{cases} \tag{1.43}$$

where $\beta_F \in (0,1)$ is the probability with which a random path is selected for the migration (i.e., *exploration*) instead of selecting the path proportionally to its population (i.e., *exploitation*). The migration from an activated path to the selected one is still given by equation (1.14), with migration rate $\sigma_p = \frac{1}{|\mathcal{P}|}$ and migration probability $\mu_{pq}[k] = \max\left\{\frac{(l_p[k]-l_q[k])}{\overline{\eta}(l_p[k]+\zeta)}, 0\right\}$, where $\overline{\eta}$ is the maximum value of the elasticity of the latency functions and ζ is a positive value needed to avoid division by 0 if $l_p[k] = 0$. In [68], the elasticity is defined as $\eta(\xi) := \frac{(x \cdot dl(\xi))/d\xi}{l(\xi)}$. The parameter α of the proposed algorithm was set equal to 0.6, whereas the parameters of the algorithm in [68] were set as follows: $\sigma_{pq} = 1/21, \zeta = 0.01, \gamma = 1/32, \beta_F = 0.002, \overline{\eta} = 2$.

Fig. 1.11 and Fig. 1.12 show the dynamics of the latency values anof the population obtained with the proposed algorithm, with $\alpha = 0.85$, and with the algorithm in [68], respectively, starting from the same initial population. The figures highlight that, while the convergence velocities appear similar, the

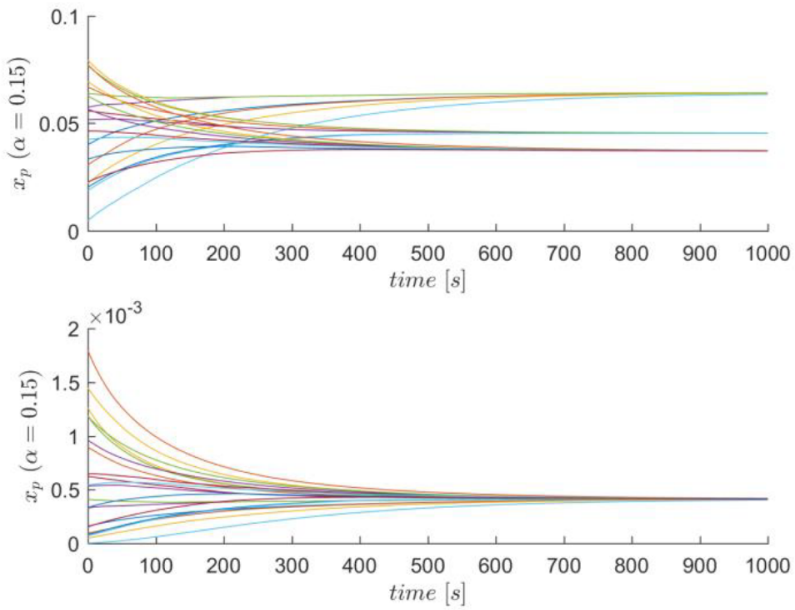


Figure 1.13: Population dynamics (higher plot) and latency dynamics (lower plot) with $\alpha = 0.15$.

dynamics of the proposed algorithm is much smoother, since the algorithm in [68] relies on migration probabilities. For the same reason, even in the long run, the algorithm in [68] produces oscillations around the value at the equilibrium – in fact, it achieves an approximate equilibrium, see [68] for details. The population dynamics (lower plots of the figures) also shows that the population of the paths converges to the equilibrium values of the three path groups defined by equation (1.42).

Fig. 1.13 shows the algorithm behaviour at values of α close to 0, starting from the same initial population of Fig. 1.11 and Fig. 1.12: at each time step, most of the paths exchange their population (according to equation (1.19) but the convergence velocity is slow, given that the σ_{pq}'s are proportional to α. Fig. 1.14 shows the algorithm behavior at values of α close to 1: at each time step, only the paths with the largest and smallest latency values exchange their populations, but the amount of population exchanged is large, given the large value of α and, in turn, of the σ_{pq}'s. Mixed behaviors are obtained along the allowed range of values $(0, 1)$ of α (see, e.g., Fig. 2 for $\alpha = 0.85$).

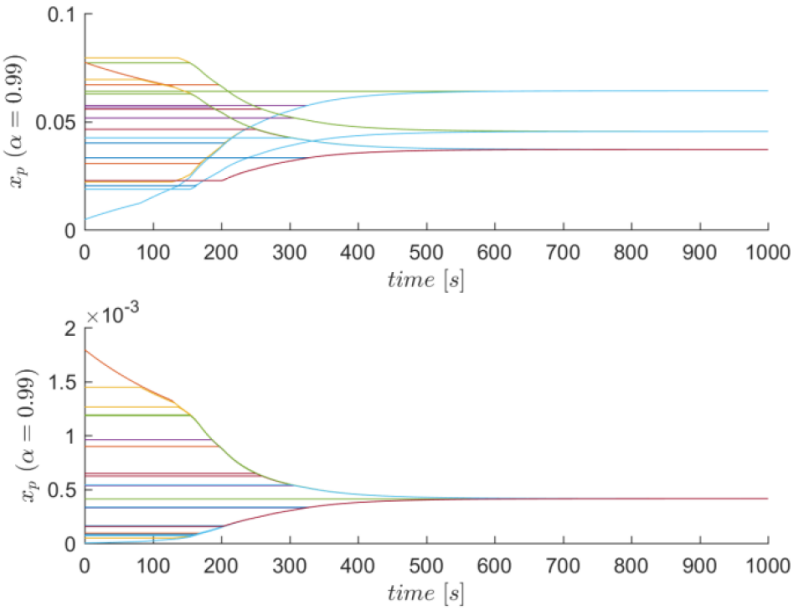


Figure 1.14: Population dynamics (higher plot) and latency dynamics (lower plot) with $\alpha = 0.99$.

1.3 Lyapunov-Based Design of a Distributed Wardrop Load Balancing Algorithm with Application to Software Defined Networking

Thanks to its characteristics, the proposed algorithm is suitable for the *Software Defined Networking* (SDN) scenario, where service requests coming from the network nodes, i.e., the switches, are managed by the so-called *SDN Controllers*, playing the role of providers. The proposed approach is aimed at dynamically balancing the requests of the switches among the SDN Controllers to avoid congestion. We also suggest the adoption of *SDN Proxies* to improve the scalability of the overall SDN paradigm and present an implementation of the algorithm in a proof-of-concept SDN scenario, which shows the effectiveness of the proposed solution with respect to the current approach.

With the advent of cloud services, enterprises and carriers have found themselves in need of evolving their network infrastructures to satisfy new requirements, such as managing a higher demand for bandwidth and for re-

sponsiveness to new data patterns (including machine-to-machine, data-center and mobility traffic), scaling IT resources, sharing the same infrastructure among different logically isolated networks, and applying network-wide policies.

In order to face these new challenges, the current research is focused on the virtualization and elastic provisioning of the network resources within the Software Defined Networking (SDN) paradigm. Such innovation is carried out by different public/private initiatives: among others, the FP7 T-NOVA and FI-Core research projects.

As defined by the *Open Network Foundation* (ONF), SDN is a network architecture where network control is decoupled from forwarding and is directly programmable. While the so-called *forwarding plane* (in charge of physically routing the data packets) resides in the network nodes, i.e., in the switches, all the intelligence related to network control is logically centralized into a single software entity called SDN Controller, responsible for the network behavior. In short, the SDN Controller is in charge of computing the routes of the packets and it is directly programmable from the upper layer (i.e., the *application plane*) through programmable interfaces.

The SDN architecture introduces some issues due to the necessity for the switches to interact with the SDN Controller: any forwarding decision of a given switch consists of a *unit of workload*, or *job*, for the SDN Controller. Thus, the throughput and the latency of the SDN Controller may become the performance bottleneck of the network. Therefore, even if the SDN Controller is logically a single centralized entity, it is actually realized through multiple entities, referred to as SDN Controllers for the sake of simplicity, sharing the overall network load. The problem is then to balance the overall workload of the SDN network among the different SDN Controllers. Currently, proximity-based approaches are proposed, where, simply, each switch (and, consequently, its workload) is associated with the closest SDN Controller. However, this static approach is not effective, especially when the workload is not evenly spatially distributed and/or is dynamic (which is always true in communication networks). Therefore, load balancing algorithms are required to dynamically associate the switches to the SDN Controllers with the overall aim of equalizing the workload of the SDN Controllers.

In this Section, the performance of the SDN Controllers is defined by a latency function, which describes how the response time of the SDN Controller grows with its workload. The objective of the load balancing algorithm is then to direct the requests of the switches to the SDN Controllers in such a way that the values of the latency functions of the SDN Controllers are equalized. The two main problems in the algorithm development are:

- the fact that the latency functions are not known (e.g., the load/delay curve of an SDN Controller depends on its specific hardware and software implementation);
- a distributed approach is needed since a centralized approach would require too much control traffic to exchange information among the SDN Controllers and potentially thousands of switches.

In this Section, a distributed, non-cooperative and dynamic load balancing algorithm is consequently developed on the ground of mean field game theory; specifically, the algorithm considers each request from a switch as an agent (whose decision is to determine the SDN Controller such a request has to be routed to), and is based on the measured response time of the SDN Controllers themselves: the algorithm is such that the agent decisions lead to an equilibrium, known in mean field game theory as Wardrop equilibrium, where the values of the latency functions of the SDN Controllers are equalized.

The main motivations behind this work are then (i) to prove, using Lyapunov arguments, how the difference equation governing the global state of the system (and macroscopically abstracting the microscopic evolution of the single agents involved) converges to an arbitrarily small neighborhood of a Wardrop equilibrium, and (ii) to show the effectiveness of such an approach through its application to a real SDN scenario.

The work presented here was carried out within the FP7 T-NOVA project (www.t-nova.eu), aimed at extending the emerging concept of SDN to the efficient reconfiguration and elastic scaling of virtualized network functionalities. Indeed, the proposed algorithm is embedded in the T-NOVA *SDN Control Plane*, allowing the management of virtual networks over data-center physical infrastructures. However, we further note that, since the algorithm is developed within the research framework of Wardrop load balancing and selfish routing, it can be applied to several problems and scenarios other than the one considered here.

1.3.1 State of the Art and Proposed Innovations with Respect to Software Defined Networks

The main SDN technology is defined by the *OpenFlow* standard [116], which is an open protocol of communication between the forwarding plane (i.e., the switches) and the control plane (i.e., the SDN Controller). The task of a switch is simply to interpret the forwarding rules sent by the SDN Controller, store them as forwarding tables and match incoming packets with the table entries.

The logical network architecture is shown in Fig. 1.15.a, where a single (logical) SDN Controller manages N switches. Since the logic is shifted from the switches to the SDN Controller, any forwarding decision consists of a unit of workload for the SDN Controller. For instance, according to [94], in large-scale scenarios (e.g., in a server cluster of 1.5×10^3 machines) the SDN workload has a mean arrival rate of about 10^5 OpenFlow messages per second. Therefore, an architectural improvement (Fig. 1.15.b) consists in physically distributing the SDN Control Plane among K SDN Controllers, arranged to form a cluster, with a node playing the role of cluster coordinator [188].

A drawback of current SDN networks resides in the fact that the mapping between the forwarding plane and the control plane, i.e., the association of each switch to a given SDN Controller, is static. In fact, in the literature, the problem of associating switches and SDN Controllers is known as "SDN Controller placement problem," and the proposed algorithms are mostly based on the location of the SDN Controllers with respect to the switches see, e.g., [84] presenting a comparison between brute-force and greedy algorithm solutions, [100] presenting a heuristic approach, and [162] solving the problem by adopting operational research theory. Since, in real networks, the network workload varies in time, the main drawback of static mapping is the necessity to find a new solution to the controller placement problem whenever some SDN Controller workload exceeds its processing power (congestion occurrence).

To overcome this problem, a dynamic approach was preliminarily defined in [38], where each switch dynamically decides the SDN Controller to be associated with, based on the response time of the SDN Controllers. This suggested approach is however not compliant with the standard SDN protocols and is not scalable, since each switch is required to explicitly receive information about the current response time of all the SDN Controllers (instead, each switch is able to measure the response time only of the SDN Controller it is associated with). Therefore, in this Section, a new entity is introduced, named SDN Proxy (see Fig. 1.15.c). The switches are statically connected to the nearest SDN Proxy. Each SDN Proxy receives the requests of its switches and has the task of forwarding them to one of the available SDN Controllers, based on a load balancing algorithm as the one proposed below. The introduction of the SDN Proxies is transparent to the OpenFlow standard and improves the SDN scalability since their tasks are quite simpler with respect to the ones of the SDN Controllers.

1.3.2 Proposed Wardrop Load Balancing Algorithm

First, we recall the basic definitions; then, we present the load balancing algorithm and the convergence proof.

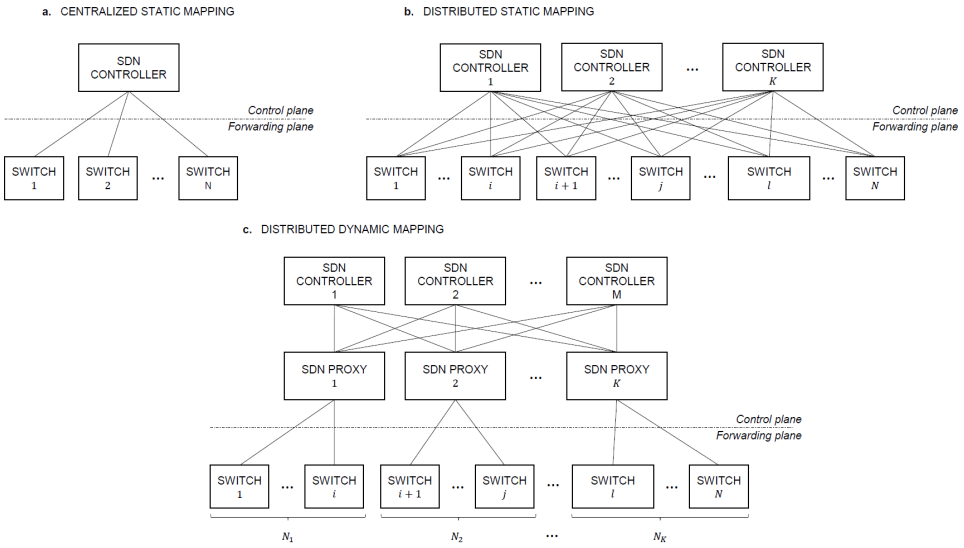


Figure 1.15: SDN architectures: a. centralized static mapping, b. distributed static mapping and c. distributed dynamic mapping.

Preliminaries on Wardrop Equilibrium and on Set Stability

Let $\mathcal{I}$ denote again a set of commodities with flow demands, or rates, $d^i > 0, i \in \mathcal{I}$, generally expressed in jobs per unit of time. For the sake of simplicity, each commodity $i \in \mathcal{I}$ is associated with a (source, destination) couple of nodes, denoted with (s^i, t^i). The d^i's are also such that $d := \sum_{i\in\mathcal{I}} d^i$. Let $\mathcal{P}$ denote again a set of providers, which are used to transmit the flows for every commodity $i \in \mathcal{I}$. All source nodes are connected by the network to the available providers, which, in turn, connect them to the destination nodes. As an example, we may think of the considered model as a description of a network consisting of a set of edges, over which the controllers arrange proper paths to connect source and destination nodes. In the considered scenario, the SDN Controllers are the providers, the SDN Proxies identify the commodities, and the d^i's are their request loads, expressed in requests per unit of time.

Let the vector $\mathbf{x} = (x_p^i)_{(p\in\mathcal{P},i\in\mathcal{I})}$ be again the flow vector (in the literature also referred to as *population share* or *job vector*), describing the overall amount of jobs per unit or time of commodity i. Let the feasible state space $\mathcal{X}$ be the same as (1.11).

Let $x_p := \sum_{i\in\mathcal{I}} x_p^i$ denote the load of provider $p \in \mathcal{P}$, and let each provider be characterized by a continuous cost function, referred to as latency function and denoted with $l_p(\cdot) \ : \ [0, d] \to \mathbb{R}_+$. The latency of a provider p is a

function of its load x_p, i.e., $l_p(x_p)$ is the latency of controller p with load x_p. The following mild assumptions on the l_p's will be considered hereinafter.

Assumption 1.2. (*Latency functions*). The latency functions exhibit the following properties.

(a) $l_p(x)$ is positive and non-decreasing with $x \in [0, d], \forall p \in \mathcal{P}$;

(b) $l_p(x)$ is Lipschitz continuous in $x \in [0, d]$, with Lipschitz constant β_p, $\forall p \in \mathcal{P}$.

□

An instance of the load balancing game is then

$$\Gamma = \left\{ \mathcal{P}, (l_p)_{p \in \mathcal{P}}, (s^i, t^i, d^i)_{i \in \mathcal{I}} \right\}. \tag{1.44}$$

The load balancing problem is formulated below as the problem of determining the strategies which will lead the flow vector to reach an arbitrarily small neighborhood of a Wardrop equilibrium. In Wardrop theory, stable flow assignments are the ones in which no agent (i.e., no small portion of a commodity directed from a source to a destination) can improve its situation by changing its strategy (i.e., the set of used providers) unilaterally. This objective is achieved if all agents reach a Wardrop equilibrium. Let us then recover Assumption 1.1, Definition 1.2, and 1.1.

The following definition and theorem on *set stability* (i.e., on the stability of a set of points in the state space) are also recalled with respect to the nonlinear discrete-time dynamics (1.22).

Definition 1.3. (*Positive definiteness*, [167]). Let $\mathcal{X}$ be an invariant set for system (1.22), let $\mathcal{A}$ be a closed subset of $\mathcal{X}$ and let $d(\mathbf{x}, \mathcal{A}) := \inf_{\mathbf{y} \in \mathcal{A}} |\mathbf{x} - \mathbf{y}|$ be the distance from a point $\mathbf{x} \in \mathcal{X} \setminus \mathcal{A}$. The function $\mathcal{L}(\mathbf{x}) : \mathcal{X} \to \mathbb{R}_+$ is positive definite with respect to the set $\mathcal{A} \subset \mathcal{X}$ if there exists an increasing continuous function $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\psi(0) = \psi_{min}$ and $\psi(d(\mathbf{x}, \mathcal{A})) \leq (x), \forall \mathbf{x} \in \mathcal{X} \setminus \mathcal{A}$.

□

Let $\Delta\mathcal{L}(\mathbf{x}[k]) := \mathcal{L}(\mathbf{x}[k+1]) - \mathcal{L}(\mathbf{x}[k])$ denote the difference of a Lyapunov function $\mathcal{L}(\mathbf{x})$ along the solutions of system (1.22). Lyapunov's second method can be applied to verify if a set is a Globally Asymptotically Stable Set (GASS) as follows [167].

Theorem 1.4. *(Globally Asymptotically Stable Set). Given a closed subset $\mathcal{A} \subset \mathcal{X}$ and a Lyapunov function $\mathcal{L}(\mathbf{x})$ in $\mathcal{X} \setminus \mathcal{A}$, if $\mathcal{L}(\mathbf{x})$ and $-\Delta\mathcal{L}(\mathbf{x}[k])$ are positive definite with respect to $\mathcal{A}$, then $\mathcal{A}$ is a GASS for system (1.22).*

Load Balancing Algorithm and Convergence Proof

Let the system dynamics (1.22) be expressed component-wise by

$$\begin{aligned} &x_p^i[k+1] = x_p^i[k] + \tau \cdot \left(\sum_{q \in \mathcal{P}} r_{qp}^i[k] - \sum_{q \in \mathcal{P}} r_{pq}^i[k] \right), \\ &\forall p \in \mathcal{P},\ \forall i \in \mathcal{I},\ k = 0, 1, \ldots, \end{aligned} \tag{1.45}$$

where δ is the sampling period and $r_{pq}^i[k]$ is the so-called migration rate from provider p to provider q. Inspired by the continuous-time algorithm in [68], the migration rate is defined as:

$$\begin{aligned} &r_{pq}^i[k] = x_p^i[k] \cdot \sigma_{pq}^i[k] \cdot \mu_{pq}^i(l_p(x_p[k]), l_q(x_q([k]))), \\ &\forall p, q \in \mathcal{P},\ \forall i \in \mathcal{I},\ k = 0, 1, \ldots, \end{aligned} \tag{1.46}$$

where $\sigma_{pq}^i[k]$ is the control gain, which sets the rate with which the population share of provider p migrates to provider q, and $\mu_{pq}(l_p, l_q)$ is the so-called migration policy, representing the decision whether (and in which percentage) the population share assigned to provider p migrates to provider q. The proposed migration policy has the following property:

$$\begin{aligned} &\begin{cases} \mu_{pq}^i(l_p, l_q) = 0, & \text{if } l_p \leq l_q + \varepsilon, \\ \mu_{pq}^i(l_p, l_q) \in [\underline{\mu}, \overline{\mu}],\ 0 < \underline{\mu} < \overline{\mu} < +\infty, & \text{otherwise}, \end{cases} \\ &\forall p, q \in \mathcal{P},\ \forall i \in \mathcal{I}, \varepsilon > 0, \end{aligned} \tag{1.47}$$

where ε is a tolerance on the maximum acceptable mismatch between the couples of latency values and $\underline{\mu}$ and $\overline{\mu}$ are positive lower- and upper-bounds, respectively. As shown in the following, the tolerance ε is introduced since the usual migration policies adopted in the continuous-time algorithms (obtained from (1.47) by setting $\varepsilon = 0$) cannot guarantee convergence in the discrete-time case, however small the sampling period (see, e.g., [67]). Let the total migration rate from provider p to provider q be defined as $r_{pq}[k] := \sum_{i \in \mathcal{I}} r_{pq}^i[k]$. For notational simplicity, whenever unambiguous, $\mu_{pq}^i[k]$ will be used in place of $\mu_{pq}^i(l_p(x_p[k]), l_q(x_q([k])))$. Before analysing the algorithm convergence, the following definition of ε-Wardrop equilibrium is introduced.

Definition 1.4. (ε-*Wardrop equilibrium*). A feasible flow vector $\mathbf{x} = (x_p^i)_{p \in \mathcal{P}, i \in \mathcal{I}}$ is defined to be at an ε-Wardrop equilibrium for the instance Γ of the load balancing game if, for each provider $p \in \mathcal{P}$ such that $l_p(x_p) > 0$, the following relation holds:

$$l_p(x_p) \leq l_u(x_u) + \varepsilon, \forall u \in \mathcal{P}, \text{for } \ 0 < \varepsilon < \overline{l} - \underline{l} \tag{1.48}$$

where $\bar{l} := \max_{p\in\mathcal{P}} l_p(d)$ and $\underline{l} := \min_{p\in\mathcal{P}} l_p(0)$ are the maximum and minimum latency values, respectively. The set of all ε-Wardrop equilibria is the following closed subset of $\mathcal{X}$:

$$\begin{aligned}\mathcal{X}_{W,\varepsilon} :=&\{\mathbf{x}\in\mathcal{X}, \varepsilon > 0 \mid l_p(x_p) \leq l_j(x_j) + \varepsilon, l_p(x_p) > 0,\\ &\forall j \in \mathcal{P},\ \forall p \in \mathcal{P},\ \forall i \in \mathcal{I},\ 0 < \varepsilon < \bar{l} - \underline{l}|\}.\end{aligned} \tag{1.49}$$

□

In practice, at an ε-Wardrop equilibrium, the latencies of all the loaded providers have the same value up to the tolerance ε. Hereafter, the following assumptions on the latency functions and on the migration policy (1.47) will be considered.

Assumption 1.3. The latency functions, the migration policy (1.47) and the control gain exhibit the following properties.

(a) $l_p(x)$ is increasing with $x \in [0, d], \forall p \in \mathcal{P}$;

(b) $l_p(x)$ is Lipschitz continuous in $x \in [0, d]$, with Lipschitz constant $\beta_p,\ \forall p \in \mathcal{P}$;

(c) $l_p(0) = l,\ \forall p \in \mathcal{P}$;

(d) $\mu^i_{pq}(l_p, l_q)$ is Lipschitz-continuous, $\forall l_p, l_q \in [\underline{l}, \bar{l}],\ \forall i \in \mathcal{I}$;

(e) $\sigma^i_{pq}[k]$ is constant and equal to $\sigma = \frac{\varepsilon}{|\mathcal{P}|\cdot d\cdot\beta_{max}\cdot\bar{\mu}\cdot\tau}$, where $\overline{\eta} := \max_{p\in V}\eta_p$;

(f) $\varepsilon < \bar{l} - \underline{l}$.

□

By imposing, via Assumption 1.3, that the latency functions are strictly increasing, the Wardrop equilibrium and the corresponding flow vector, denoted with l_W and $\mathbf{x}_W$, respectively, are unique.

Definition 1.5. (*Distance*). Let the norm of a state be defined as $\|\mathbf{x}\| := \max_{p\in\mathcal{P}}(l_p(x_p) - l_W)$, and let the distance between a state $\mathbf{x} \in \mathcal{X} \setminus \mathcal{X}_{W,\varepsilon}$ and the set $\mathcal{X}_{W,\varepsilon}$ be defined as $d(\mathbf{x}, \mathcal{X}_{W,\varepsilon}) := \|\mathbf{x}\| - \max_{\mathbf{y}\in\mathcal{X}_{W,\varepsilon}}\|\mathbf{y}\| > 0$.

□

Under Assumption 1.3, the set of Wardrop and ε-Wardrop equilibria can be written as:

$$\mathcal{X}_W := \{\mathbf{x}\in\mathcal{X} \mid \|\mathbf{x}\| = 0\} = \{\mathbf{x}_W\}; \tag{1.50}$$

$$\mathcal{X}_{W,\varepsilon} := \{\mathbf{x}\in\mathcal{X}, \varepsilon > 0 \mid \|\mathbf{x}\| \leq \varepsilon\}. \tag{1.51}$$

and the following properties hold.

Property 1.2. The feasible set $\mathcal{X}$, and the set of the ε-Wardrop equilibria $\mathcal{X}_{W,\varepsilon}$ are such that:

(P1) $\mathcal{X}_{W,\varepsilon} = \mathcal{X}$ if $\varepsilon \geq \bar{l} - \underline{l}$;

(P2) $\mathcal{X} \supset \mathcal{X}_{W,\varepsilon_2} \supset \mathcal{X}_{W,\varepsilon_1} \supset \{\mathbf{x}_W\},\ \forall \varepsilon_1, \varepsilon_2 \mid 0 < \varepsilon_1 < \varepsilon_2 < \bar{l} - \underline{l}$;

(P3) $\mathcal{X}_{W,\varepsilon} \to \{\mathbf{x}_W\}$ as $\varepsilon \to 0$.

□The set convergence to an arbitrarily small neighbourhood of the Wardrop equilibrium is proven by using the Beckmann, McGuire, and Winsten potential (1.12) to build a candidate Lyapunov function. The following lemma demonstrates some properties of the potential which will be used in the convergence proof of the subsequent Theorem 1.6.

Lemma 1.5. *(Properties of the potential). Under Assumption 1.3, the following properties hold for the nonlinear discrete-time system (1.45), (1.46), (1.47), with total flow rate $d > 0$:*

(L1) $\Phi(\mathbf{x}) > \Phi_{min},\ \forall \mathbf{x} \in \mathcal{X} \setminus \{\mathbf{x}_W\},\ \Phi(\mathbf{x}_W) = \Phi_{min}$;

(L2) $\Phi(\mathbf{x}[k+1]) - \Phi(\mathbf{x}[k]) < 0, \forall \mathbf{x}[k] \in \mathcal{X} \setminus \mathcal{X}_{W,\varepsilon}$;

(L3) $\Phi(\mathbf{x}[k+1]) - \Phi(\mathbf{x}[k]) = 0, \forall \mathbf{x}[k] \in \mathcal{X}_{W,\varepsilon}$.

Proof. . Conditions (L1) hold thanks to Property 1.1. To verify condition (L2), the variation of the Lyapunov function along any trajectory of the considered system is written as follows:

$$
\begin{aligned}
\Delta\Phi(\mathbf{x}[k]) = \Phi(\mathbf{x}[k+1]) - \Phi(\mathbf{x}[k]) &= \sum_{p\in\mathcal{P}} \int_{x_p[k]}^{x_p[k+1]} l_p(s)ds \leq \\
&\leq \sum_{p\in\mathcal{P}} (x_p[k+1] - x_p[k]) l_p(x_p[k+1]) = \\
&= \sum_{p\in\mathcal{P}} (\sum_{q\in\mathcal{P}} r_{qp}[k] - \sum_{q} r_{pq}[k]) \cdot \tau \cdot l_p(x_p[k+1]) = \\
&= \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{qp}[k] \cdot \tau \cdot l_p(x_p[k+1]) - \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{pq}[k] \cdot \tau \cdot l_p(x_p[k+1]) = \\
&= \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{pq}[k] \cdot \tau \cdot l_q(x_q[k+1]) - \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{pq}[k] \cdot \tau \cdot l_p(x_p[k+1]) = \\
&= \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} r_{pq}[k] \cdot \tau \cdot [l_q(x_q[k+1]) - l_p(x_p[k+1])],
\end{aligned}
\tag{1.52}
$$

where the inequality holds from geometrical considerations.

Indeed, the quantity $(x_2 - x_1) \cdot l(x_2)$, represented by the area of the rectangle with bold lines in Fig. 1.16, is larger than the integral $\int_{x_1}^{x_2} l(s)ds$, represented by the grey area.

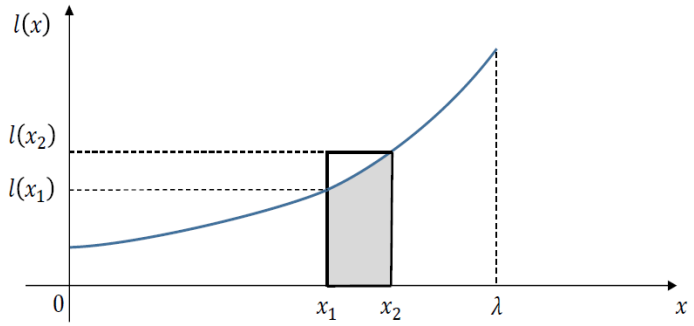


Figure 1.16: Geometrical considerations proving inequality (1.52).

The following shows that (i), if $l_p(x_p[k]) - l_q(x_q[k]) > \varepsilon$, the corresponding term of the summation in the last row of (1.52) is negative, whereas (ii), if $l_p(x_p[k]) - l_q(x_q[k]) \leq \varepsilon$, then the term is null.

(i) If $l_p(x_p[k]) - l_q(x_q[k]) > \varepsilon$, from Assumptions 1.3.a)-1.3.c) it follows that $x_p[k] > 0$ and that $l_p(x_p[k]) > 0$. Moreover $\mu^i_{pq}[k] > 0$ from equation (1.47) and, thus, $r^i_{pq}[k] > 0$, $\forall i \in \mathcal{I}$. Now we need to show that $l_p(x_p[k+1]) - l_q(x_q[k+1]) > 0$. From equation (1.45) the following inequality holds:

$$\begin{aligned} & l_p(x_p[k+1]) - l_q(x_q[k+1]) \geq \\ & \geq l_p\left(x_p[k] - \sum_{q\in\mathcal{P}} r_{pq}[k]\cdot\tau\right) - l_q\left(x_q[k] + \sum_{p\in\mathcal{P}} r_{pq}[k]\cdot\tau\right), \end{aligned}$$

$$\forall p, q \in \mathcal{P},\ k = 0, 1, \ldots \tag{1.53}$$

In equation (1.53), the worst-case system dynamics over δ is considered, in which no commodities migrate part of their population to provider p and from provider q. From Assumption 1.3, since β_{max} is an upper-bound for the largest derivative of the l_p's, it holds that

$$\begin{aligned} l_p(x_p[k]) - l_p\left(x_p[k] - \sum_{q\in\mathcal{P}} r_{pq}[k]\cdot\tau\right) &\leq \beta_{max}\cdot\sum_{q\in\mathcal{P}} r_{pq}[k]\cdot\tau; \\ l_q(x_q[k] + \sum_{p\in\mathcal{P}} r_{pq}[k]\cdot\tau) - l_q(x_q[k]) &\leq \beta_{max}\cdot\sum_{p\in\mathcal{P}} r_{pq}[k]\cdot\tau. \end{aligned} \tag{1.54}$$

Since we are analysing the $l_p(x_p[k]) - l_q(x_q[k]) > \varepsilon$ case, the following inequality holds:

$$l_p(x_p[k+1]) - l_q(x_q[k+1])\varepsilon - \beta_{max} \cdot \tau \cdot \left(\sum_{q \in \mathcal{P}} r_{pq}[k] + \sum_{p \in \mathcal{P}} r_{pq}[k] \right). \quad (1.55)$$

From equation (1.46) and Assumption 1.3.a), and considering that $x_p^i[k] \leq d^i, \ \forall i \in \mathcal{I}$, the following upper-bound holds:

$$r_{pq}^i[k] = x_p^i[k] \cdot \sigma_{pq}^i[k] \cdot \mu_{pq}^i[k] \leq d^i \cdot \sigma \cdot \beta_{max}, \ \forall p, q \in \mathcal{P}, \ \forall i \in \mathcal{I}. \quad (1.56)$$

Then, considering that there are at most $(|\mathcal{P}| - 1)$ terms in the first summation of the second term of (1.54), it is upper-bounded by

$$\sum_{q \in \mathcal{P}} r_{pq}[k] = \sum_{i \in \mathcal{I}} \sum_{q \in \mathcal{P}} r_{pq}^i[k] \leq \sigma \cdot \overline{\mu} \cdot (|\mathcal{P}| - 1)d. \quad (1.57)$$

Also, the second summation of the second term of (1.54) is upper-bounded by:

$$\sum_{p \in \mathcal{P}} r_{pq}[k] = \sum_{i \in \mathcal{I}} \sum_{p \in \mathcal{P}} r_{pq}^i[k] \leq \sigma \cdot \overline{\mu} \cdot \sum_{i \in \mathcal{I}} \sum_{p \in \mathcal{P}} x_p^i[k] \leq \sigma \cdot \overline{\mu} \cdot d. \quad (1.58)$$

From equations (1.56) and (1.57), we obtain that a sufficient condition for the right-hand side of equation (1.54) to be non-negative is the following:

$$\varepsilon \geq |\mathcal{P}| \cdot \sigma \cdot d \cdot \overline{\mu} \cdot \beta_{max} \cdot \sigma, \quad (1.59)$$

which holds by Assumption 1.3.e).

(ii) If $l_p(x_p[k]) - l_q(x_q[k]) \leq \varepsilon, r_{pq}[k] = 0$ by equations (1.46) and (1.47), and the corresponding term of the summation in the last row of (1.52) is null.

From (i) and (ii) it follows that property (L2) holds since, if $\mathbf{x}[k] \notin \mathcal{X}_{W,\varepsilon}$ (i.e., there exists at least one couple $(p, q) \in \mathcal{P}^2$ such that $l_p(x_p[k]) - l_q(x_q[k]) > \varepsilon$, which, in turn, entails that $x_p[k] > 0$), at least one term of equation (1.52) is negative; property (L3) holds since, if $\mathbf{x}[k] \in \mathcal{X}_{W,\varepsilon}$ (i.e., for all couples $(p, q) \in \mathcal{P}^2$ with $l_p(x_p[k]) > 0$ we have that $l_p(x_p[k]) - l_q(x_q[k]) \leq \varepsilon$), all the terms of equation (1.52) are null. □

Theorem 1.6. *(ε-Wardrop equilibrium set as a GASS). Under Assumption 1.3, $\mathcal{X}_{W,\varepsilon}$ is a GASS for the nonlinear discrete-time system (1.45), (1.46), (1.47), with total flow rate $d > 0$.*

Proof. The proof is structured as follows: first, it is shown that the feasible state space is a positively invariant set (A); secondly, the asymptotic set stability is proven (B).

(A) Feasibility.

It is shown in the following that, since the initial job vector is feasible (i.e., from Definition 1.11, $\sum_{p\in\mathcal{P}} x_p^i[0] = d^i$ and $x_p^i[0] \geq 0, \forall p \in \mathcal{P},\ \forall i \in \mathcal{I}$), the job vector is feasible during the entire system dynamics. In fact, it follows from equation (1.45) that:

$$\sum_{p\in\mathcal{P}}(x_p^i[k+1] - x_p^i[k]) = \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{P}}(r_{qp}^i[k] - r_{pq}^i[k])\cdot\tau = \sum_{p\in\mathcal{P}}\sum_{q\in\mathcal{P}} r_{qp}^i[k]\cdot\tau - \\ - \sum_{q\in\mathcal{P}}\sum_{p\in\mathcal{P}} r_{qp}^i[k]\cdot\tau = 0, \tag{1.60}$$

and, therefore, that $\sum_{p\in\mathcal{P}} x_p^i[k] = \sum_{p\in\mathcal{P}} x_p^i[0] = d^i,\ \forall k \geq 0$. By induction, since $x_p^i[0] \geq 0,\ \forall p \in \mathcal{P}$, and given equation (1.45), in order to prove that $x_p^i[k] \geq 0,\ \forall k \geq 0$, it is sufficient to assume that $x_p^i[k] \geq 0,\ \forall p \in \mathcal{P},\ \forall i \in \mathcal{I}$, for a given k, and to prove that

$$x_p^i[k+1] = x_p^i[k] + \sum_{q\in\mathcal{P}} r_{qp}^i[k] - r_{pq}^i[k])\cdot\tau \geq 0, \forall p \in \mathcal{P},\ \forall i \in \mathcal{I}. \tag{1.61}$$

In this respect, it can be observed that the following inequality holds (considering that, in the worst-case, no commodities migrate part of their population to provider p):

$$x_p^i[k+1] \geq x_p^i[k] - \sum_{q\in\mathcal{P}} r_{pq}^i[k]\cdot\tau,\ \forall p \in \mathcal{P},\ \forall i \in \mathcal{I}. \tag{1.62}$$

From definition (1.47) it follows that $r_{pp}^i[k] = 0$, so there are at most $(|\mathcal{P}| - 1)$ terms in the summation of equation (1.61). Thus, considering that $r_{pq}^i[k] \leq x_p^i[k]\cdot\sigma\cdot\overline{\mu}$, the condition in (1.61) is met if the following inequality holds:

$$x_p^i[k] - x_p^i[k]\cdot(|\mathcal{P}| - 1)\cdot\sigma\cdot\overline{\mu}\cdot\tau \geq 0,\ \forall p \in \mathcal{P}, \forall i \in \mathcal{I}. \tag{1.63}$$

If $x_p[k] = 0$, inequality (1.62) is verified. If $x_p[k] > 0$, inequality (1.62) is verified provided that:

$$\sigma \leq \frac{1}{((|\mathcal{P}| - 1)\cdot\tau\cdot\overline{\mu})}, \tag{1.64}$$

which holds by Assumption 1.3.e), considering that $\frac{\varepsilon}{\beta_{max}\cdot\lambda} < 1$ (in fact, by the definitions of β_{max} and $\overline{l}$, it holds that $\beta_{max}\cdot\lambda \geq \overline{l}$, and, by Assumption 1.3.f), it holds that $\overline{l} > \varepsilon$).

(B) Global asymptotic set stability.

Let $\mathcal{L}(\mathbf{x}) := \Phi(\mathbf{x}) - \Phi_{min}$ be the candidate Lyapunov function, where $\Phi(\mathbf{x})$ is the potential (1.12) and Φ_{min} is its minimum value, which is unique thanks to Assumption 1.3. If $\mathbf{x} \in \mathcal{X}_{W,\varepsilon}$, from Lemma 1.5 it follows that $\mathcal{L}(\mathbf{x})$ is positive definite and that $\Delta\mathcal{L}(\mathbf{x}[k]) = 0$. If $\mathbf{x} \in \mathcal{X} \setminus \mathcal{X}_{W,\varepsilon}$, it is shown below that $\mathcal{L}(\mathbf{x})$ and $-\Delta\mathcal{L}(\mathbf{x}[k])$ are positive definite with respect to the closed set $\mathcal{X}_{W,\varepsilon}$.

B1. $\mathcal{L}(\mathbf{x})$ *is positive definite with respect to* $\mathcal{X}_{W,\varepsilon}$.

Let $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ be defined as follows: $\psi(d(\mathbf{x}, \mathcal{X}_{W,\varepsilon})) := \gamma_1 d(\mathbf{x}, \mathcal{X}_{W,\varepsilon})$, with $\gamma_1 > 0$. By definition, we have that $\psi(0) = 0$ and that $\psi(d(\mathbf{x}, \mathcal{X}_{W,\varepsilon}))$ is increasing with $d(\mathbf{x}, \mathcal{X}_{W,\varepsilon})$. We have to show that $\psi(d(\mathbf{x}, \mathcal{X}_{W,\varepsilon})) = \gamma_1 d(\mathbf{x}, \mathcal{X}_{W,\varepsilon}) \leq \mathcal{L}(\mathbf{x}) = \Phi(\mathbf{x}) - \Phi_{min},\ \forall \mathbf{x} \in \mathcal{X} \setminus \mathcal{X}_{W,\varepsilon}$, i.e., that a value for γ_1 exists such that the following inequality holds:

$$\gamma_1 \leq \frac{(\Phi(\mathbf{x}[k]) - \Phi_{min})}{d(\mathbf{x}[k], \mathcal{X}_{W,\varepsilon})},\ \forall k = 0, 1, 2, \ldots \tag{1.65}$$

Since $x \in \mathcal{X} \setminus \mathcal{X}_{W,\varepsilon}$, by definition we have that $d(\mathbf{x}, \mathcal{X}_{W,\varepsilon}) > 0$; moreover, $d(\mathbf{x}, \mathcal{X}_{W,\varepsilon})$ is upper-bounded by $(\bar{l} - \underline{l})$ (by Assumption 1.3.d)). By geometrical considerations, it turns out that $\Phi(\mathbf{x} - \Phi_{min} > \frac{\varepsilon^2}{4\bar{\eta}} > 0$ for all $x \in \mathcal{X} \setminus \mathcal{X}_{W,\varepsilon}$.

Indeed, let $\mathbf{x}_W$ be the flow vector at the Wardrop equilibrium, i.e., $l_p(x_{p,W}) = l_W,\ \forall p \in \mathcal{P}$, and let $\mathbf{x} \in X \setminus X_{W,\varepsilon}$ be defined by the vector $(x_p)_{p\in\mathcal{P}} = (x_{p,W} + \Delta x_p)_{p\in\mathcal{P}}$, where Δx_p is the difference (positive, negative or null) between x_p and $x_{p,W}$. The difference between the potential in $\mathbf{x}$ and the potential (1.12) at the equilibrium $\Phi_{min} = \Phi(\mathbf{x}_W)$

is then:

$$\begin{aligned}\Phi(\mathbf{x}) - \Phi(\mathbf{x}_W) = \Bigg(& \sum_{p\in\mathcal{P} \mid \Delta x_p>0} \int_0^{x_{p,W}+\Delta x_p} l_p(s)ds + \\ & + \sum_{q\in\mathcal{P} \mid \Delta x_q<0} \int_0^{x_{q,W}+\Delta x_q} l_q(s)ds + \\ & + \sum_{m\in\mathcal{P} \mid \Delta x_m=0} \int_0^{x_{m,W}} l_m(s)ds) - \\ & - \sum_{n\in\mathcal{P}} \int_0^{x_{n,W})} l_n(s)ds \Bigg) = \\ & = \sum_{p\in\mathcal{P} \mid \Delta x_p>0} \int_{x_{p,W}}^{x_{p,W}+\Delta x_p} l_p(s)ds - \\ & - \sum_{q\in\mathcal{P} \mid \Delta x_q<0} \int_{x_{q,W}+\Delta x_q}^{x_{q,W}} l_q(s)ds.\end{aligned} \tag{1.66}$$

Considering the areas B_q and A_q depicted in Fig. 1.17, we can write:

$$\begin{aligned}\Phi(\mathbf{x}) - \Phi_{min} &= \sum_{p\in\mathcal{P} \mid \Delta x_p>0} (\Delta x_p l_W + A_p) - \sum_{q\in\mathcal{P} \mid \Delta x_q<0} (\Delta x_q l_W - B_q) = \\ &= \sum_{p\in\mathcal{P} \mid x_p>0} A_p + \sum_{q\in\mathcal{P} \mid \Delta x_q<0} B_q \geq \\ &\geq \sum_{p\in\mathcal{P} \mid \Delta x_p>0} \frac{(\varepsilon_p^2)}{(2\beta_{max})} + \sum_{q\in\mathcal{P} \mid \Delta x_q<0} \frac{(\varepsilon_q^2)}{(2\beta_{max})},\end{aligned} \tag{1.67}$$

where the last equality holds since $A_p \geq \frac{(\varepsilon_p^2)}{(2\beta_{max})}$ and $B_q \geq \frac{(\varepsilon_q^2)}{(2\beta_{max})}$ and where $\varepsilon_p := l_p(x_p) - l_W$ and $\varepsilon_q := l_W - l_q(x_q)$. Since $\mathbf{x} \in \mathcal{X} \setminus \mathcal{X}_{W,\varepsilon}$, there exists at least a couple $p^{'}, q^{'} \in \mathcal{P}$ such that $\Delta x_{p'} > 0, \Delta x_{q'} < 0$ and $\varepsilon_{p'} + \varepsilon_{q'} > \varepsilon$. It follows that:

$$\Phi(\mathbf{x}) - \Phi_{min} \frac{(\varepsilon_{p'}^2)}{(2\beta_{max})} + \frac{(\varepsilon_{q'}^2)}{(2\beta_{max})} \geq \frac{(\varepsilon_{p'} + \varepsilon_{q'})^2}{(4\beta_{max})} > \frac{\varepsilon^2}{(4\beta_{max})}. \tag{1.68}$$

Therefore, according to such geometrical considerations, a suitable choice for γ_1 is $\gamma_1 = \frac{\varepsilon^2}{(4\beta_{max}(\bar{l}-\underline{l}))}$.

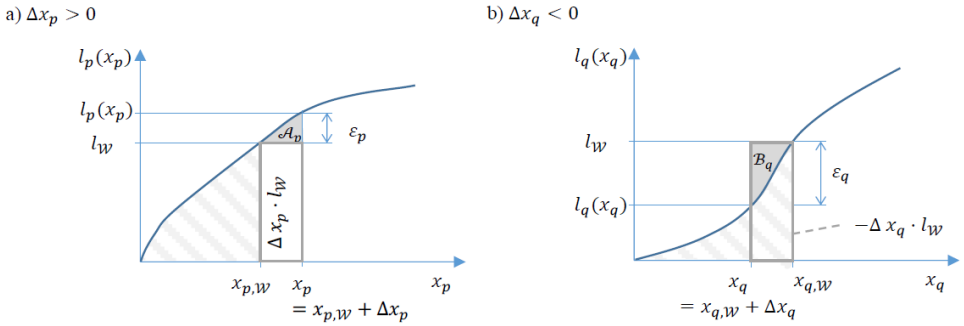


Figure 1.17: Geometrical considerations proving argument B1.

B2. $-\Delta\mathcal{L}(\mathbf{x}[k])$ *is positive definite with respect to* $\mathcal{X}_{W,\varepsilon}$.

Let $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ be defined as $\psi(d(\mathbf{x}, \mathcal{X}_{W,\varepsilon})) := \gamma_2 d(\mathbf{x}, \mathcal{X}_{W,\varepsilon})$, with $\gamma_2 > 0$. Also, consider that $\Delta\mathcal{L}(\mathbf{x}[k]) = \Delta\Phi(\mathbf{x}[k])$ and that, from Lemma 1.5, $\Delta\Phi(\mathbf{x}[k]) < 0$, $\forall \mathbf{x}[k] \in \mathcal{X} \setminus \mathbf{x}_W$. We have to show that $\psi(d(\mathbf{x}[k], \mathcal{X}_{W,\varepsilon})) = \gamma_2 d(\mathbf{x}, \mathcal{X}_{W,\varepsilon}) \leq -\Delta\mathcal{L}(\mathbf{x}[k]) = -\Delta\Phi(\mathbf{x}[k])$, $\forall \mathbf{x}[k] \in \mathcal{X} \setminus \mathcal{X}_{W,\varepsilon}$, i.e., that there exists a value for γ_2 such that the following inequality holds:

$$\gamma_2 \leq \frac{(-\Delta\Phi(\mathbf{x}[k]))}{d(\mathbf{x}[k], \mathcal{X}_{W,\varepsilon})}, \ \forall k = 0, 1, 2, \ldots \tag{1.69}$$

For the numerator of equation (1.64), the following inequality holds (see Lemma 1.5, equation (1.52)):

$$-\Delta\Phi(\mathbf{x}[k]) = -\sum_{p\in\mathcal{P}} \int_{x_p[k]}^{x_p[k+1]} l_p(s)ds \geq$$
$$\geq \sum_{p\in\mathcal{P}} \sum_{q\in\mathcal{P}} \tau \cdot r_{pq}[k](l_p(x_p[k+1]) - l_q(x_q[k+1])), \quad \forall k = 1, 2, \ldots, \tag{1.70}$$

where the terms of the last summation are either null or positive. From equations (1.54) and (1.58) it follows that $l_p(x_p[k+1]) - l_q(x_q[k+1]) \geq 0$. Let us consider the provider p^* which has the maximum latency value at time k, i.e., $p^* = argmax_{p\in\mathcal{P}}\ l_p(x_p[k])$ We thus write from equations (1.65), (1.46) and (1.47):

$$-\Delta\Phi(\mathbf{x}[k]) \geq \tau \cdot x_{p^*}[k] \cdot \sigma \cdot \underline{\mu}. \tag{1.71}$$

Since $\mathbf{x}[k] \in \mathcal{X} \setminus \mathcal{X}_{W,\varepsilon}$, we know that $l_{p^*}(x_{p^*}[k]) - l_q(x_q[k]) > \varepsilon$, $\forall q \in \mathcal{P}$, yielding $l_{p^*}(x_{p^*}[k]) > \varepsilon$; since β_{max} is the upper-bound for the Lipschitz

constants of the l_p's, we have that $\beta_{max} \cdot x_{p^*}[k] \geq l_{p^*}(x_{p^*}[k]) > \varepsilon$. Thus, recalling that $d(\mathbf{x}, \mathcal{X}_{W,\varepsilon}) \leq (\bar{l} - \underline{l})$, the following choice for γ_2 lets inequality (1.64) hold for all $k = 0, 1, \ldots : \gamma_2 = \frac{(\tau \cdot \varepsilon \cdot \sigma \cdot \mu)}{(\beta_{max} \cdot (\bar{l} - \underline{l})}$.

□

Remark 1.4. In the control law (1.46), $\sigma^i_{pq}[k]$ can be interpreted as the control gain, and the interpretation of Theorem 1.6 is that it sets an upper-bound on the control gain, with the twofold objective of keeping the dynamics feasible and of driving the system trajectories towards a neighbourhood of $\mathbf{x}_W$.

1.3.3 Proof-of-Concept Application to Software Defined Networking

As discussed above, in order to manage the SDN Control Plane traffic in large-scale networks, the currently adopted solutions rely on a Control Plane adopting a logically centralized but physically distributed architectural model. This model considers the Control Plane as distributed across a cluster of multiple SDN Controllers. The OpenDaylight SDN Controller [117], an open-source project of the Linux Foundation sponsored by a large consortium of networking companies, introduced a cluster-based implementation: it runs on a cluster of machines which share a distributed data store to maintain the global view of the network.

At the same time, the OpenFlow protocol, since version 1.3, has regulated its architectural model by defining the concept of SDN Controller for a switch. This enabled two modes of operation when multiple SDN Controllers exist in a network: *master/slave interaction* and *equal interaction*. In the master/slave interaction each switch can be associated with all the SDN Controllers but managed by only one (the master), responsible for all the events corresponding to that switch, whereas the others (slaves) are used as backup controllers. On the contrary, in equal interaction, each switch can be associated with multiple SDN Controllers and have more than one master association, that is, several equal associations.

We now show how the presented algorithm has been applied to enforce the discussed load balancing mechanisms onto the distributed virtualized SDN Controllers developed within the T-NOVA project. In particular, we present the results obtained in a scenario where the considered SDN works in equal interaction mode across a cluster of three OpenDaylight SDN Controllers (*Beryllium* release). A set of virtual switches deployed with the *Mininet* framework is connected to the set of SDN Controllers by means of SDN Proxies. The switches are statically connected to the SDN Proxies. Each

SDN Proxy is connected to all the SDN Controllers and can measure their response time. Each SDN Proxy is a network proxy for the OpenFlow traffic, developed in Python, and embeds the proposed load balancing algorithm.

Following the presented approach, each SDN Proxy decides to forward each request, received from its switches, to a given SDN Controller, based on the current response time of the SDN Controllers; the decisions are taken without any communications among the switches and/or the controllers themselves.

In this case study, the latency associated with an SDN Controller is its average response time. The response time grows with the controller load and thus (i) it is a reliable indicator of the controller congestion status, (ii) it is a non-decreasing function of the request rate and therefore a suitable latency function, and (iii) it can be easily computed by the SDN Proxies, as explained below.

The time-scale is divided into rounds of duration τ. At every round, the latency is evaluated as follows.

- The SDN Proxy collects the last measurements of the response time of all the SDN Controllers by measuring the delay between the transmission of a request to an SDN Controller and its answer. Let l_p^{meas} denote the last measure of the response time of the SDN Controller p. If an SDN Proxy has not sent a request to a given SDN Controller, say SDN Controller p, in the last n rounds, it sends a fake request to the SDN Controller p in order to update the response time measure.

- The value of the latency function is updated by a simple exponential averaging approach: $l_p \leftarrow \alpha l_p + (1-\alpha) l_p^{meas}$.

At round k, based on the updated latency values, the i-th SDN Proxy computes the rates $x_p^i[k]$ to all the SDN Controllers p by means of the proposed algorithm. Then, the i-th SDN Proxy sends each one of the requests received during round k to one of the SDN Controllers following a weighted round-robin scheduling, with weights proportional to the rates $x_p^i[k]$. In the implementation, the maximum latency value $\bar{l}$ coincides with the maximum tolerated response time. In practice, the implementation of the control law (1.46)-(1.47) is straightforward, except for the determination of the maximum latency value $\bar{l}$ and of the maximum Lipschitz constant β_{max}.

In most cases, the latency value represents an actual measure of the performance of the providers and can be upper-bounded according to realistic considerations. The latency value is then set equal to $\bar{l}$ whenever the value of the latency computed from the provider performance measures is larger than $\bar{l}$. For instance, in the described scenario, the latency represents the delay of

the controller responses; in practice, there are Quality-of-Service constraints which should be met by the provider, in terms of maximum response time, defining the upper-bound $\bar{l}$.

Table I: Implementation Parameters	
Number of SDN Controllers	$\lvert\mathcal{P}\rvert = 3$
Number of SDN Proxies	$\lvert\mathcal{I}\rvert = 2$
Overall SDN Controller load	$d = 6 \left[\frac{req}{s}\right]$
Maximum latency value	$\bar{l} = 4\ [s]$
Minimum latency value	$\underline{l} = 0\ [s]$
Maximum Lipschitz constant of the latency functions	$\beta_{max} = 2 \left[\frac{s^2}{req}\right]$
Latency tolerance	$\varepsilon = 0.25\ [s]$
Normalization parameter of the migration policy	$c_\mu = 0.25\ [s^{-1}]$
Maximum value of the migration policy	$\overline{\mu} = 1$
Sampling time	$\tau = 1\ [s]$
Number of sampling times to send a fake request	$\eta = 1$
Averaging constant	$\alpha = 0.95$

The Lipschitz constant of the latency functions, instead, must be estimated on-line based on the actual controller responses. In real applications, the latency derivative grows with the provider load (e.g., the typical load-delay curve is often modelled in the theory of M/M/1 queues as $l(x) = \frac{1}{d-x}$ or $l(x) = \frac{x}{d-x}$. Therefore, by limiting the maximum value of the latency functions according to practical consideration (as described above), we also limit the Lipschitz constants: the gain $\sigma^i_{pq}[k]$ is then increased and, in turn, the system dynamics is made faster. To simulate the SDN controller load, OpenFlow control traffic was generated on the forwarding plane side at different request rates. The three deployed SDN Controllers were assumed to be heterogeneous, i.e., they were set up with different hardware specifications (respectively, 3/3/6 GB of RAM and 2/4/4 vCPUs) and therefore different processing capabilities. Consequently, as shown in Fig. 1.18, their load-response time curves are different. The maximum Lipschitz constant was estimated as $\beta_{max} := \max_{p\in\mathcal{P}} \beta_p \approx 2\ [s^2/req]$. To validate the effectiveness of the load balancing algorithm, different tests with generated OpenFlow control traffic have been performed. The requests were generated according to an exponential distribution with mean $d = 6\ [req/s]$ on the forwarding plane and then sent to the SDN Proxies which implement the proposed algorithm, with migration

policy defined as:

$$\mu^i_{pq}(l_p, l_q) = \mu(l_p, l_q) = \begin{cases} 0 \quad \text{if} \;\; l_p \leq l_q + \varepsilon, \\ c_\mu(l_p - l_q) \quad \text{otherwise}, \end{cases} \qquad \forall p, q \in \mathcal{P}, \forall i \in \mathcal{I}, \; \varepsilon > 0, \tag{1.72}$$

where c_μ is a normalization constant. For comparison purposes, the same test was run without SDN Proxies (i.e., without load balancing), and assuming that the load is equally distributed among the SDN Controllers, i.e., each SDN Controller receives, on average, 2 requests per second. The authors have considered an implementation scenario with parameters as reported in Table I.

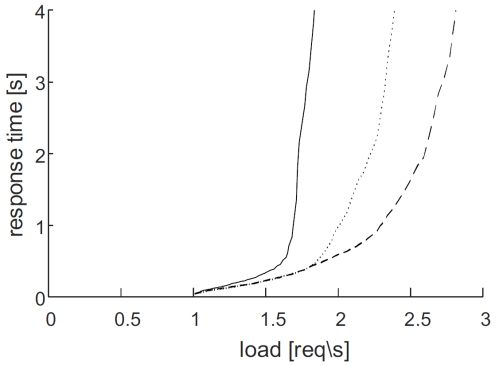


Figure 1.18: Load-response time curves of the implemented SDN Controllers. Solid line: SDN Controller 1; dashed line: SDN Controller 2; dotted line: SDN Controller 3 .

Figs. 1.19-1.22 show the test results. Fig. 1.19 compares the response times of the SDN Controllers in the test without load balancing (upper plot) to the ones in the test with load balancing (lower plot): in the former case, one of the SDN Controllers is congested and its response time rapidly grows over 4 s, whereas, if load balancing is implemented, the response time of all the SDN Controllers is kept well below 1 s. Correspondingly, Fig. 1.20 shows that the latency values (which are simply exponential averages of the response times) grow over the maximum tolerated latency $\bar{l} = 4$ s if no load balancing is implemented, whereas the latency of all the SDN Controllers is kept at about 0.2 s if load balancing is implemented. Fig. 1.21 and Fig. 1.22 show how the result is obtained. Fig. 1.21 shows that the load balancer is capable of distributing the overall load among the SDN Controllers; in particular, it feeds the worst performing SDN Controller (i.e., SDN Controller 1) with less requests and the best performing SDN Controller (i.e., SDN Controller 3)

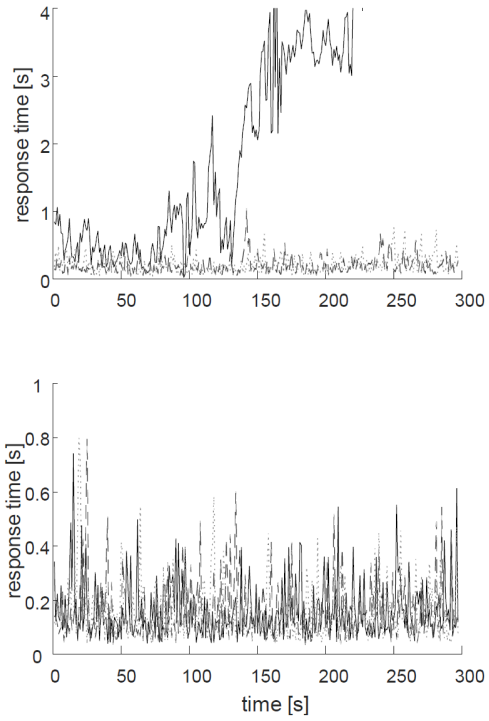


Figure 1.19: Response time without load balancing (upper plot) and with load balancing (lower plot). Solid line: SDN Controller 1; dashed line: SDN Controller 2; dotted line: SDN Controller 3.

with more requests. Fig. 1.22 shows the shares of requests sent by the SDN Proxies to the SDN Controllers.

We have proposed a Wardrop load balancing algorithm for SDN networks, and introduced two innovations. From the methodological viewpoint, a distributed, non-cooperative discrete-time load balancing algorithm, based on mean-field game theory, was presented and proved to converge to an arbitrarily small neighborhood of a Wardrop equilibrium. From an architectural point of view, SDN Proxies for the OpenFlow traffic were introduced to improve the scalability of SDN networks by dynamically dispatching the control workload across the available SDN Controllers. To evaluate the effectiveness of the proposed approach, a proof-of-concept implementation on a real SDN network was carried out and the related performance test results were reported. The proposed approach is scalable, since no communications among the switches is needed and no centralized load balancing algorithm must be executed by the SDN Controllers.

Future work is aimed at validating the algorithm on larger use cases.

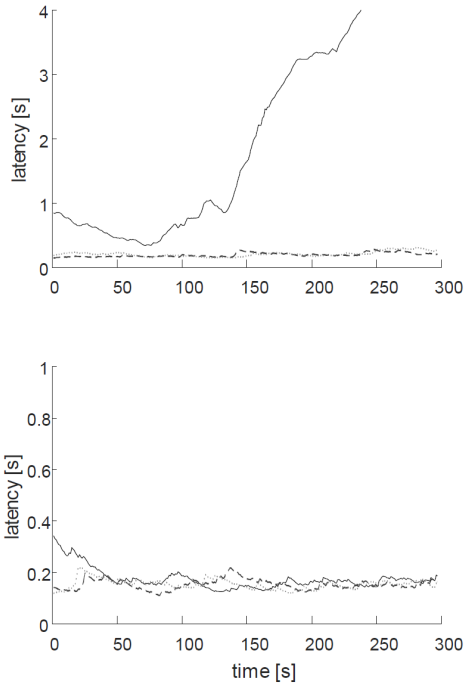


Figure 1.20: Latency values without load balancing (upper plot) and with load balancing (lower plot). Solid line: SDN Controller 1; dashed line: SDN Controller 2; dotted line: SDN Controller 3.

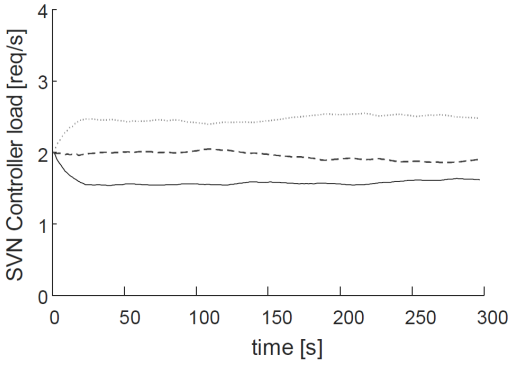


Figure 1.21: Loads of the SDN Controllers (averaged with an exponential average for clarity of the presentation). Solid line: SDN Controller 1; dashed line: SDN Controller 2; dotted line: SDN Controller 3.

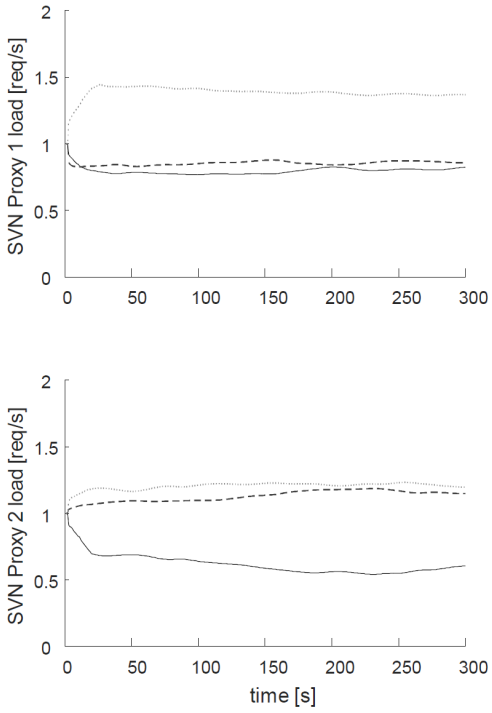


Figure 1.22: Distribution of the loads of the SDN Proxies among the SDN Controllers (averaged with an exponential average for clarity of the presentation). Solid line: SDN Controller 1; dashed line: SDN Controller 2; dotted line: SDN Controller 3.

1.4 Optimization of Energy Storage System Placement for Protecting Power Transmission Grids Against Dynamic Load Altering Attacks

Over the last years, the need for securing power grids against the danger of cyber-physical attacks has been increasingly encouraging the development of distributed intelligence technologies accompanied by appropriate security enforcements. In particular, cyber-physical attacks have been targeting all sectors of power systems, i.e., generation, distribution and control, and consumption. In this respect, a suitable classification with meaningful examples is given in [144]. More specifically, as concerns cyber-physical attacks targeting the generation sector, the interested reader is referred to [19] and [171];

as concerns, instead, cyber-physical attacks targeting the distribution and control sector, the reader is referred to [107] and [180].

This Section is focused on cyber-physical attacks targeting the consumption sector. In particular, we are concerned with Load Altering Attacks (LAAs) whose aim is to maliciously alter a group of remotely accessible yet unsecured controllable loads, thus artificially creating power imbalances in the power network responsible for frequency and load angle instability, and consequently network blackout through sequential generator tripping.

LAAs can be classified into *static* ones, which abruptly modify the volume of certain vulnerable loads *una tantum*, and*dynamic* ones (hereafter referred to as D-LAAs), which not only determine the volume of the change enforced onto the compromised load, but also establish the load trajectory over time. D-LAAs can either be *open-loop* – such that the attacker is not capable of monitoring the power grid in real-time and therefore assigns a pre-programmed trajectory to the compromised load based on some available historical data – or *closed-loop*. Whenever a closed-loop D-LAA is struck against a power grid, the attacker continuously monitors the grid conditions through his own installed sensors or by hacking into an existing monitoring infrastructure, and consequently uses the feedback from the power grid frequency to alter the victim load buses. Moreover, we distinguish between *single-point* closed-loop D-LAAs, which compromise only the vulnerable load at one victim load bus, and *multi-point* ones, which compromise the vulnerable loads at several victim load buses in a coordinated fashion in order to maximize the attack impact [5].

In this Section, based on the IEEE 39-bus test system, we design a protection scheme against closed-loop single-point and multi-point D-LAAs by formulating and solving a non-convex optimization problem subject to a Lyapunov stability constraint. We take into account the most relevant power system dynamics, and feedback control theory is here used as a tool to model and build a remedy action against the attack: this adds to the already existing results on the control-theoretic study of cyber-physical systems [142] [143]. The proposed protection scheme relies upon the proper installation of suitably-sized Energy Storage Systems (ESSs) [61] [10] in order to mitigate the effects of the ongoing D-LAA and preserve the power systems stability. In this regard, ESS technology has significantly improved over the last years, with possible applications starting to be investigated at transmission [74] [56], distribution [130] [57], microgrid [102] and consumer [55] level. The presented setup is also of practical interest due to its link to the concept of frequency-responsive loads [121] [186], which are expected to support traditional power plants in the provisioning of frequency regulation services.

In particular, this study has been carried out within the framework of

the H2020 ATENA project, which is aimed at developing ICT networked components for the detection of and reaction to adverse events in the context of cyber-physical security for Critical Infrastructures (CI), where it is crucial to prevent the propagation of damage to other CIs interdependent with the power grid (see also the FP7 projects MICIE and CockpitCI as well as the SHIELD framework and the related publications [47]).

This Section is organized as follows: Subsection 2.1 describes the mathematical model of the considered power transmission network subject to a D-LAA; Subsection 2.2 proposes the adopted problem formulation for the optimization of ESS placement; eventually Subsection 1.4.3 reports the obtained simulation results, showing the effectiveness of the proposed approach.

1.4.1 Mathematical Model of the IEEE 39-Bus Test System Under a D-LAA

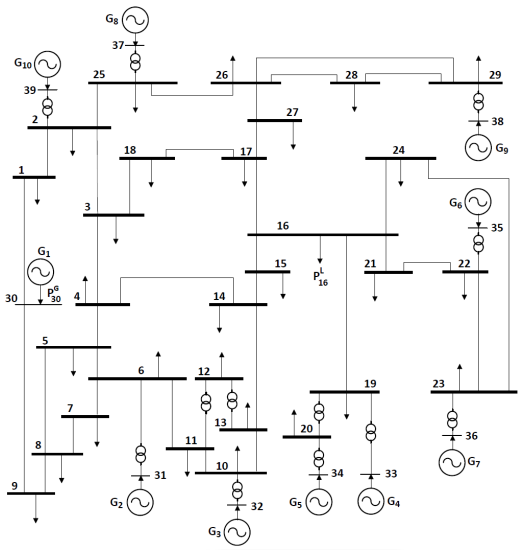


Figure 1.23: The IEEE 39-bus test system.

We now present the mathematical model for the IEEE 39-bus test system based on the 10-machine New-England power network and depicted in Fig. 1.23: we will use this model for the design, relying upon ESSs, of a protection scheme against D-LAAs. Let $\mathcal{G}$ and $\mathcal{L}$ represent the sets of generator buses and load buses, respectively, across the grid. More in detail, the IEEE 39-bus test system is made of 10 generator buses and 29 load buses, so we assume that $\mathcal{L} = \{1, \ldots, 29\}$ and $\mathcal{G} = \{30, \ldots, 39\}$. Let then $\mathcal{N} = \mathcal{G} \cup \mathcal{L}$ represent

the set of all buses across the grid. For a generic bus $i \in \mathcal{N}$, the total amount of power delivered can be separated into generator and load terms [144]. Namely, power flow equations can be written distinguishing the power amount P_i^G injected into the grid by each generator $i \in \mathcal{G}$ and the total power P_i^L absorbed by each load bus $i \in \mathcal{L}$. By defining δ_i as the voltage phase angle of the i-th generator bus, θ_i as the voltage phase angle of the i-th load bus and H_{ij} as the admittance value between the generic i-th and j-th buses, it follows that

$$\begin{aligned} P_i^G &= \sum_{j \in \mathcal{G}} H_{ij}(\delta_i - \delta_j) + \sum_{j \in \mathcal{L}} H_{ij}(\delta_i - \theta_j), \quad \forall i \in \mathcal{G} \\ -P_i^L &= \sum_{j \in \mathcal{G}} H_{ij}(\theta_i - \delta_j) + \sum_{j \in \mathcal{L}} H_{ij}(\theta_i - \theta_j), \quad \forall i \in \mathcal{L}. \end{aligned} \tag{1.73}$$

As regards the generator buses, the swing equations are adopted to model the dynamic behavior of each generator $i \in \mathcal{G}$, i.e.,

$$\begin{aligned} \dot{\delta}_i &= \omega_i \\ M_i(\omega_i) &= P_i^M - P_i^G - D_i^G \omega_i, \end{aligned} \tag{1.74}$$

where ω_i is the rotor frequency deviation at the i-th generator bus, M_i is the rotor inertia associated with the i-th generator, P_i^M is the mechanical power input and $D_i^G \omega_i$ is the damping term, proportional to the frequency deviation, $\forall i \in \mathcal{G}$. We assume that the inertia M_i and the damping coefficient D_i^G are strictly positive In particular, according to [76], it is possible to combine a turbine-governor control action with a load-frequency one into a proportional-integral (PI) controller, aimed at keeping the rotor frequency at its nominal level by pushing the frequency deviation ω_i back to zero. Said PI controller is represented by

$$P_i^M = -\left(K_i^P \omega_i + K_i^I \int_0^t \omega_i \right), \quad K_i^P, K_i^I > 0. \tag{1.75}$$

Consequently, the rotor frequency dynamics in equation (1.74) can be rewritten by expressing the mechanical power P_i^M for each generator in terms of frequency deviation ω_i, as defined in (1.75). It follows that

$$M_i \dot{\omega}_i = -\left(K_i^P \omega_i + K_i^I \int_0^t \omega_i \right) - P_i^G - D_i^G \omega_i, \tag{1.76}$$

and, since the power P_i^G injected by the generating unit is defined according to (1.73) and the integral of the frequency deviation is equal to the voltage

phase angle of the generator, we obtain, $\forall i \in \mathcal{G}$,

$$\begin{aligned} M_i\dot{\omega}_i = &-K_i^P\omega_i - K_i^I\delta_i - \\ &- \sum_{j\in\mathcal{G}} H_{ij}(\delta_i - \delta_j) - \sum_{j\in\mathcal{L}} H_{ij}(\delta_i - \theta_j) - D_i^G\omega_i. \end{aligned} \tag{1.77}$$

After some manipulations, we have

$$-M_i\dot{\omega}_i = (K_i^P + D_i^G)\omega_i + K_i^I\delta_i + \sum_{j\in\mathcal{G}} H_{ij}(\delta_i - \delta_j) + \sum_{j\in\mathcal{L}} H_{ij}(\delta_i - \theta_j), \quad \forall i \in \mathcal{G}. \tag{1.78}$$

As regards the load buses, instead, following [186] we use P_i^L to define the aggregate power consumption of (i) uncontrollable loads as well as of (ii) controllable but frequency-insensitive ones. On the other hand, (iii) controllable and frequency-sensitive loads can be assumed to increase linearly with the frequency deviation at the load buses: it follows that the related power consumption can be modeled by $D_i^L\phi_i$, where D_i^L is the strictly positive damping term of the i-th load bus and $\phi_i = -\dot{\theta}_i$ is the frequency deviation at each bus $i \in \mathcal{L}$. We can rewrite (1.73), $\forall i \in \mathcal{L}$, as follows,

$$\begin{aligned} &\dot{\theta}_i = -\phi_i \\ &- D_i^L\phi_i - P_i^L = \sum_{j\in\mathcal{G}} H_ij(\theta_i - \delta_j) + \sum_{j\in\mathcal{L}} H_ij(\theta_i - \theta_j). \end{aligned} \tag{1.79}$$

Equations (1.74), (1.78), and (1.79) define the complete dynamical model of the IEEE 39-bus test system depicted in Fig. 1.23. The power grid can now be represented in the form of a linear state-space descriptor model. First of all, we need to arrange the admittance values, appearing in equations (1.78), and (1.79), into four different matrices, that is, (i) H^{GG}, containing the admittance values associated with the lines connecting buses in $\mathcal{G}$; (ii) H^{GL}, containing the admittance values associated with the lines between generator and load buses; (iii) $H^{LG} = (H_{GL})^T$; (iv) H^{LL}, containing the admittance values associated with the lines connecting the buses in $\mathcal{L}$. Therefore, the complete admittance matrix of the power system is

$$H = \begin{pmatrix} H^{GG} & H^{GL} \\ H^{LG} & H^{LL} \end{pmatrix}. \tag{1.80}$$

Moreover, the inertia and damping values (M_i and D_i^G, respectively) in (1.78), as well as the damping terms D_i^L in (1.79), can be collected into properly-dimensioned diagonal matrices, namely M, D^G, and D^L. The same considerations apply to the proportional and integral values K_i^P and K_i^I

as well as to the load power consumptions P_i^L. Eventually, by defining $\delta = \begin{pmatrix}\delta_1 & \dots & \delta_{10}\end{pmatrix}^T$ as the vector of the voltage phase angles associated with the generators, $\theta = \begin{pmatrix}\theta_1 & \dots & \theta_{29}\end{pmatrix}^T$ as the vector of the voltage phase angles associated with the load buses, $\omega = \begin{pmatrix}\omega_1 & \dots & \omega_{10}\end{pmatrix}^T$ as the vector of the frequency deviations of the generators, and $\phi = \begin{pmatrix}\phi_1 & \dots & \phi_{29}\end{pmatrix}^T$ as the vector of the load frequency deviations, and considering δ, θ, ω, and ϕ as state variables, the complete linear state-space descriptor model for the IEEE 39-bus test system is

$$\begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & -M & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\delta} \\ \dot{\theta} \\ \dot{\omega} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & -I \\ K^I + H^{GG} & H^{GL} & K^P + D^G & 0 \\ H^{LG} & H^{LL} & 0 & -D^L \end{pmatrix} \begin{pmatrix} \delta \\ \theta \\ \omega \\ \phi \end{pmatrix} + \\ + \begin{pmatrix} 0 \\ 0 \\ 0 \\ I \end{pmatrix} P^L, \tag{1.81}$$

where the I's are properly-dimensioned identity matrices.

Let us now plug a D-LAA into the system reported above. By definition, a D-LAA is aimed at compromising a certain amount of vulnerable load in specific grid areas and at controlling its evolution over time so that the overall interconnected system is considerably altered and damaged. Therefore, in line with [5], we regard power consumption at the load buses, i.e., P^L, as the sum of two contributions: part of the load consumption is identified as a protected portion P^{LS}, while P^{LV} denotes the vulnerable unprotected portion of the load:

$$P^L = P^{LS} + P^{LV}. \tag{1.82}$$

Let $\mathcal{V} \subseteq \mathcal{L}$ be the set of victim load buses and let $\mathcal{S} \subseteq \mathcal{N}$ be the set of the positions of sensors which are capable of attack detection. Accordingly, let $K_{vs}^{LG} \geq 0$ denote the attack gain at victim bus $v \in \mathcal{V}$ if the sensor bus s is a generator bus (belonging to $\mathcal{G}$), and $K_{vs'}^{LL} \geq 0$ denote the control gain of the attacker at bus $v \in \mathcal{V}$ if the sensor bus s' is a load bus (belonging to $\mathcal{L}$). A D-LAA against the power grid can then be modelled by the proportional controller

$$P_v^{LV} = -K_{vs}^{LG}\omega_s - K_{vs'}^{LL}\phi_{s'}, \tag{1.83}$$

where ω_s is the generator frequency deviation measured by a sensor bus $s \in \mathcal{G}$, and $\phi_{s'}$ is the frequency deviation of the load buses measured by a sensor bus

$s^{'} \in \mathcal{L}$. In particular, the D-LAA is such that the update of P^{LV} is inversely proportional to frequency deviation: namely, if ω_s decreases (increases), then the amount of vulnerable load increases (decreases), and the same holds with respect to $\phi_{s'}$. Hence, equation (1.83) is a proportional controller modelling a D-LAA against the power grid. By the way, note that other choices (such as PID or PD controllers) are also possible to model such attacks. On this basis, the power grid under attack is modelled by substituting (1.83) into (1.82), and then into (1.81), thus obtaining

$$\begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & -M & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\delta} \\ \dot{\theta} \\ \dot{\omega} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & -I \\ K^I + H^{GG} & H^{GL} & K^P + D^G & 0 \\ H^{LG} & H^{LL} & -K^{LG} & -D^L - K^{LL} \end{pmatrix} \begin{pmatrix} \delta \\ \theta \\ \omega \\ \phi \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ I \end{pmatrix} P^L. \tag{1.84}$$

When the system is under attack, the attacker can compromise the grid stability by properly modifying the controller gains, and, subsequently, the amount of vulnerable unprotected load P^{LV}. Formally, from a control-theoretic point of view, the closed-loop system above becomes unstable if controller gains K^{LG} and K^{LL} are capable of moving the system poles to the right-hand side of the complex plane, that is, to the unstable region for continuous-time linear systems.

1.4.2 Optimization of ESS Placement

As in [5], the idea is to exploit the notion of Lyapunov stability in combination with an optimization criterion so as to guarantee power grid security in the presence of a D-LAA characterized as in (1.83). More specifically, in this Section it is proposed to solve the following problem: given a power grid whose load buses are assumed to be potential victims to a D-LAA, determine the minimum number of ESSs (with fixed size) and their exact locations in order to protect the system against the ongoing D-LAA. In this respect, a proper optimization problem can be defined where ESSs are modelled based on feedback from the frequency deviations detected all across the power grid. Let us assume that the term P^{LS}, that is, the protected portion of the power consumption P^L at the load buses, be the power provided by a certain number of ESSs at different locations in the power grid.

Let us suppose that the sensor bus s is necessarily a generator bus, i.e., $s \in \mathcal{S} \subseteq \mathcal{G}$, and consequently K^{LL} is set to zero. The power provided by an

ESS placed at the victim load bus $v \in \mathcal{V}$ can be modelled by a proportional controller in the form

$$P_v s^{LS} = K_{vs}^{LS} \omega_s, \tag{1.85}$$

where $K_{vs}^{LS} \geq 0$ denotes the storage gain at each victim load bus v when the sensor is located at generator bus s and ω_s is the frequency deviation measured at bus s. In other words, we assume that the ESS operating conditions are strictly related to the power grid state and, therefore, to the frequency deviations that occur as a result of the D-LAA being struck against the power grid itself.

Neglecting the $K^{LL}\phi$ term due to the assumption on the sensor bus, the power consumption P^L in (1.82) can be then rewritten as

$$P^L = (K^{LS} - K^{LG})\omega. \tag{1.86}$$

The resulting closed-loop system dynamics – modelling the power grid subject to the D-LAA and to ESS control for attack mitigation – is obtained by substituting (1.86) into (1.81) so as to have

$$\begin{pmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & -M & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\delta} \\ \dot{\theta} \\ \dot{\omega} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & -I \\ K^I + H^{GG} & H^{GL} & K^P + D^G & 0 \\ H^{LG} & H^{LL} & K^{LS} - K^{LG} & -D^L \end{pmatrix} \begin{pmatrix} \delta \\ \theta \\ \omega \\ \phi \end{pmatrix}. \tag{1.87}$$

The last row of the descriptor system above can be solved with respect to ϕ and properly substituted in order to obtain an equivalent linear state-space model, i.e.,

$$\begin{pmatrix} \dot{\delta} \\ \dot{\theta} \\ \dot{\omega} \end{pmatrix} = (A - BK) \begin{pmatrix} \delta \\ \theta \\ \omega \end{pmatrix}, \tag{1.88}$$

where

$$\begin{aligned} A &= \begin{pmatrix} 0 & 0 & I \\ -(D^L)^{-1}H^{LG} & -(D^L)^{-1}H^{LL} & 0 \\ -M^{-1}(K^I + H^{GG}) & -M^{-1}(H^{GL}) & -M^{-1}(K^P + D^G) \end{pmatrix}, \\ B &= \begin{pmatrix} 0 & (D^L)^{-1} & 0 \end{pmatrix}^T, \\ K &= \begin{pmatrix} 0 & 0 & K^{LS} - K^{LG} \end{pmatrix}. \end{aligned} \tag{1.89}$$

At this point, we can formulate the optimization problem. In particular, according to Lyapunov's stability theorem for linear systems, the system poles are required to be kept inside the left-hand side of the complex plane. In

this respect, the following linear matrix inequality has to hold if we want to ensure Lyapunov stability, i.e.,

$$(A - BK)^T X + X(A - BK) < 0, \tag{1.90}$$

with K as in (1.89), thus implying that the stability of the overall system is strictly related to (i) the entity of the D-LAA against the power grid, and to (ii) the ESS size.

Before formulating the optimization problem, a feasibility constraint on the entity of the D-LAA has to be formulated. Namely, we assume that the attack intensity cannot be greater than the difference between the total vulnerable load at victim load bus v (P_v^L) and the power provided by the corresponding ESS. In other words, the more power the ESSs provide, the less effective the D-LAA against the power grid is.

$$K_{vs}^{LG}\omega_s^{max} \leq \frac{(P_v^L - P_v^{LS})}{2} = \frac{(P_v^L - K_{vs}^{LS}\omega_s)}{2}, \tag{1.91}$$

where ω_s^{max} denotes the maximum admissible frequency deviation for generator s before its over or under frequency relays trip [5]. Another constraint to be enforced can be expressed in terms of the ESS size. Namely, the storage control gain is limited according to the following relation:

$$K_{vs}^{LS}\omega_s^{max} \leq P_v^{LS,max}, \tag{1.92}$$

where $P_v^{LS,max}$ is the maximum power provided by the ESS, expressed in p.u. Under these constraints, the optimization problem can be formulated as follows.

Problem 1.1. (*Optimization of number and location of ESSs protecting the power grid against a D-LAA*). Given the total vulnerable load P^L at victim load bus $v \in \mathcal{V}$ and given a proper ESS size, determine the minimum number and the exact location of ESSs so that the power grid is asymptotically stable, that is,

$$\min \|K^{LS}\|_0 \tag{1.93}$$

subject to

$$\begin{aligned} &X \geqslant 0, \\ &X = X^T, \\ &\text{Eqs. (1.90), (1.91), and (1.92), } \forall v \in \mathcal{V}. \end{aligned} \tag{1.94}$$

By minimizing the l_0-norm of vector K^{LS} (i.e., the vector listing all energy storage control gains K_{vs}^{LS} at $v \in \mathcal{V}$ and $s \in \mathcal{S}$), it is possible to determine the

minimum number and the optimal location of the ESSs to be installed in the power grid in order to prevent a D-LAA in the form (1.83) from compromising the overall system stability. However, note that a solution to this problem is not easily found, because solving a cardinality minimization problem is NP-hard [131], and due to the presence of the non-convex quadratic constraint defined by (1.91). For the former problem, an approximation is needed to reduce the computational complexity. A common choice is the minimization of the l_1 norm, characterized by sparse feasible solutions (i.e., solutions which have null elements) [160]. Generally, a non-convex optimization problem may have multiple solutions, it may be infeasible or it can take exponential time to determine the global minimum across all admissible solution regions. In order to overtake non-convexity, we exploit a two-step solution approach, adapted from [5] and inspired by the coordinate descent method whose convergence is guaranteed [16]. First, note that inequality (1.91) has to turn into an equality when attempting to solve Problem 1.1. In fact, if (1.91) holds as a strict inequality, when the optimal solution is found, one could think of reducing the value of P^{LS} and consequently lower the objective function, thus contradicting the optimality status. It then follows that the constraint in (1.91) should be rewritten as an equality, making K_{vs}^{LG} act as a slack variable, i.e.,

$$K_{vs}^{LG}\omega_s^{max} = \frac{(P_v^L - K_{vs}^{LS}\omega_s)}{2}. \tag{1.95}$$

This way, we reduce the decision variables of the optimization problem to K^{LS} and X, since K_{vs}^{LG} is now univocally defined by the vulnerable loads and the power injected by the ESSs. Nevertheless, these two variable sets are still coupled through the attack control gain K_{vs}^{LG} and the non-convex constraint defined by equation (1.95). To this end, the problem is split up into the two following coupled subproblems.

- Step (1). Initially, the storage control gain vector K^{LS} is assumed to be constant, thus easily determining the attack control gain K_{vs}^{LG} according to constraint (1.95). This way, we can solve a feasibility problem over variable X, i.e.,

  $$\min \|K^{LS}\|_1 \tag{1.96}$$

  subject to

  $$\begin{aligned} &X \geqslant 0, \\ &X = X^T, \\ &\text{Eqs. (1.90) and (1.95)}, \forall v \in \mathcal{V}, \end{aligned} \tag{1.97}$$

  where the decision variables are the entries of matrix X. Such a feasibility problem can also be classified as a semi-definite program [20].

- Step (2). Next, we take the solution X of the feasibility problem above as a constant and we solve Problem 1.1 over K^{LS} only, i.e.,

$$\min \|K^L S\|_1 \tag{1.98}$$

subject to

$$\text{Eqs. (1.90), (1.92), and (1.95), } \forall v \in \mathcal{V}, \tag{1.99}$$

where the decision variables are the entries of K^{LS}.

These two steps are iterated until convergence is reached. In particular, note that the ESS number and placement is assessed, as a result of the optimization procedure: the non-zero elements of the resulting K^{LS} vector identify the optimal number and location of the ESSs to be deployed.

1.4.3 Simulation Results

The simulations presented in this section have been carried out using MATLAB: in particular, the authors relied upon the CVX package for determining a numerical solution to Problem 1.1 according to the two-step iterative procedure explained above. As regards the values of the parameters of the transmission lines, of the inertia (i.e., M) and damping coefficients (i.e., D^G) of generators, of the generator controller gains (i.e., the K_i^Ps) and of the damping coefficients for each dynamic load (i.e., the D_i^Ls), such values are chosen as in [5]. In particular, the controller parameters are set in order to keep the overall system stable during normal operations, i.e., in the absence of an attack. The nominal system frequency is 60 Hz. We assume that the over-frequency relays of the generators trip at 62 Hz, whereas the under-frequency relays trip at 58 Hz. Consequently, i.e., $\omega_s^{max} = \frac{2}{60}$. The vulnerable loads at each load bus are reported in Table 4.3. Note that, unlike [5] we are assuming the power loads reported in Table I to be entirely vulnerable. Therefore, in our scenario, the way chosen to protect them is by relying on the power provision allowed by suitably-deployed ESSs.

First Attack Scenario

With respect to the 10-machine New-England power network depicted in Fig. 1.23, in the first attack scenario we assume that only a subset of vulnerable loads can be regarded as potential victims to a D-LAA. Let us consider as potential victims only the load buses identified by $\mathcal{V} = \{6, 16, 19, 23, 29\}$ and let us assume that the sensor capable of detecting the ongoing attack is located at generator bus $s = 33 \subset \mathcal{G}$. Let us also assume that the vulnerable

Table 1.1: Vulnerable Loads at each Load Bus (P^L).

Load Bus v	P_v^L (p.u.)	Load Bus v	P_v^L (p.u.)	Load Bus v	P_v^L (p.u.)
1	4	11	4	21	6.7
2	4	12	4.1	22	4
3	7.2	13	4	23	9.8
4	9	14	4	24	7
5	4	15	7.2	25	6.2
6	7	16	10.9	26	5.4
7	6.3	17	4	27	6.8
8	9.2	18	5.6	28	6.1
9	4	19	5.6	29	15.1
10	4	20	10.3	-	-

loads at the victim load buses are $P^L = \begin{pmatrix} 7 & 10.9 & 5.6 & 9.8 & 15.1 \end{pmatrix}^T$ and let the ESS size be equal to the available load at the victim load buses. This last assumption implies that the initial values of the storage control gains are set to $\frac{P^L}{2\omega_s^{max}}$. Starting from control gains initialized to the maximum admissible values, the iterative algorithm discussed above is run so as to solve this instance of Problem 1.1. Since we intend to determine the minimum number of ESSs and their exact location in the power grid, the obtained simulation results claim that, by introducing one ESS located at load bus no. 19 with storage capacity equal to 5.6 p.u., the power grid remains stable under the considered D-LAA.

Second Attack Scenario

In the second attack scenario, we still assume that the sensor detecting the D-LAA is located at generator bus $s = 33$ and that the victim load buses are $\mathcal{V} = \{6, 16, 19, 23, 29\}$. This time, however, we intend to analyze the impact of the ESS size on the optimization problem solution: by contrast with the previous scenario, where the ESS size is fixed and initialized to $\frac{P^L}{2\omega_s^{max}}$, we now consider different sizes and assess how the power provided by the ESSs influences the feasibility of the optimization problem. To this end, we assume that the power provided by the ESSs starts from 1 p.u. The iterative algorithm is run for each different size in order to determine whether the corresponding problem for the determination of the optimal ESS location is solvable. Starting from ESSs with unit size, the problem is solved; then, by increasing the size by one unit at a time, the problem is solved again. Such

a procedure is repeated until convergence to a constant number of ESSs is obtained, with no further increase in the number of ESSs as the size grows. In particular, the optimal ESS placement problem turns out to be infeasible for ESS sizes equal to 1 p.u., 2 p.u., and 3 p.u.: indeed, for such values, the iterative algorithm proposed above is not able to determine an admissible solution such that the overall system stays stable under attack. Note that, instead, at 4 p.u., it is possible to determine an exact number of ESSs (i.e., 2) such that the overall system stability is ensured. For sizes of 5 p.u. or greater, just one ESS is sufficient to guarantee stability under the considered D-LAA.

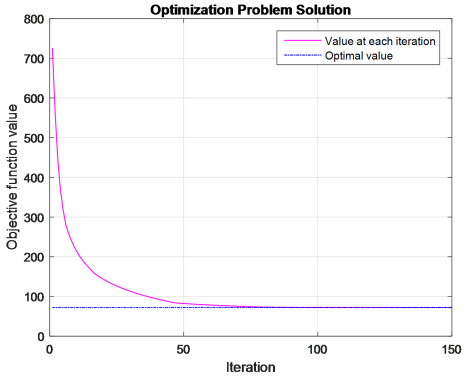


Figure 1.24: Convergence result of the iterative algorithm solving Problem 1 in the first attack scenario.

In this Section a protection scheme making use of energy storage systems for improving power system reaction to closed-loop dynamic load altering attacks is presented. The problem is formulated as a non-convex optimization problem subject to a Lyapunov stability constraint for the autonomous representation of the power system obtained after linearization and application of the attack and frequency control laws. The reported results show how the proposed two-step iterative algorithm allows to determine a solution to the problem of optimizing the number and location of energy storage systems, ensuring grid stability. Yet, the deployability of the resulting solution depends on the availability on the market of suitably-sized energy storage systems. Future work will be aimed at tackling the presented problem by means of a greedy method for finding a sparse solution, namely the so-called *matching pursuit* one, with the aim of comparing the related results with those obtained by minimizing the l_1 norm of the storage control gain vector. Moreover, we will consider the placement of energy storage systems for reducing the possibility of designing undetectable attacks as well as for their usage in support of

secondary regulation services.

Chapter 2

State of the Art on Distributed Coordination of Multi-Agent Systems

This Chapter discusses some of the most interesting results in the state of the art on distributed coordination of multi-agent systems. In particular, it gives an overview stretching from first-order systems [45] [33] [176] [139] [129] [41] to second-order systems [153] [152] [173] [106], through Lyapunov-based analysis and control, also with specific focus on networked Lagrangian systems [161] [36] [37] [118], on the one hand, and on sampled-data control [82] [72] [73] [184], on the other hand.

More specifically, this Chapter is organized as follows: Section 2.1 discusses discrete-time and continuous-time consensus in first-order systems; Section 2.2 discusses how consensus is achieved with respect to second-order systems and in particular to mass-spring-damper systems; Section 2.3 discusses consensus in 2D with rotation matrices; then Section 2.4 proposes a Lyapunov-based approach to the analysis and control of nonlinear multi-agent models; eventually, Sections 2.5 and 2.6 provide an in-depth discussion of some interesting results (with the related simulations) concerning distributed leaderless coordination of networked Lagrangian systems and sampled-data coordinated tracking for single-integrator multi-agent dynamics, respectively. The investigation of these results has laid the foundation for the innovative work discussed in the subsequent Chapter 3.

2.1 First-Order Systems

2.1.1 Discrete-Time Consensus in Linear Systems

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ characterized by $n := |\mathcal{V}|$ nodes, playing the role of agents, we can represent, in discrete time, the dynamics of each agent as follows [176] [129]:

$$x_i[k+1] = \sum_{j \in \mathcal{N}_i[k] \cup \{i\}} \beta_{ij}[k] x_j[k], \quad \forall v_i \in \mathcal{V}. \tag{2.1}$$

The β_{ij}'s represent the weights of the communication links and $\mathcal{N}_i$ is the neighbour set of agent v_i, $\forall i \in \mathcal{V}$. In short, the discrete-time dynamics is computed as the weighted average of the neighbours' current states at time k.

In particular, the following condition holds

$$\sum_{j \in \mathcal{N}_i[k] \cup \{i\}} \beta_{ij}[k] = 1, \ \beta_{ij} \geq 0. \tag{2.2}$$

According to (2.1) and (2.2), let us consider, for instance, the discrete-time system dynamics:

$$\begin{aligned}
x_1[k+1] &= \frac{1}{4}x_1[k] + \frac{1}{4}x_2[k] + \frac{1}{4}x_4[k] + \frac{1}{4}x_5[k] \\
x_2[k+1] &= \frac{1}{3}x_1[k] + \frac{1}{3}x_2[k] + \frac{1}{3}x_5[k] \\
x_3[k+1] &= \frac{1}{2}x_2[k] + \frac{1}{2}x_3[k] \\
x_4[k+1] &= \frac{1}{3}x_1[k] + \frac{1}{3}x_4[k] + \frac{1}{3}x_5[k] \\
x_5[k+1] &= \frac{1}{4}x_1[k] + \frac{1}{4}x_2[k] + \frac{1}{4}x_4[k] + \frac{1}{4}x_5[k],
\end{aligned} \tag{2.3}$$

which, can be expressed, in vector form, as

$$\begin{gathered}
\mathbf{x}[k+1] = \mathcal{D}\mathbf{x}[k], \quad \text{i.e.,} \\
\begin{pmatrix} x_1[k+1] \\ x_2[k+1] \\ x_3[k+1] \\ x_4[k+1] \\ x_5[k+1] \end{pmatrix} = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} \end{pmatrix} \begin{pmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \\ x_4[k] \\ x_5[k] \end{pmatrix},
\end{gathered} \tag{2.4}$$

where $\mathbf{x}[k] = col\begin{pmatrix} x_1[k] & \dots & x_5[k] \end{pmatrix}$. This matrix $\mathcal{D}$ is very similar to the adjacency matrix in continuous-time linear multi-agent systems, except for

the main diagonal. The reason why the elements in the diagonal are nonzero is that each node x_i at time $k+1$ uses also the information related to its own previous state $x_i[k]$.

Namely, the main characteristic of matrix $\mathcal{D}$ is that it is a *stochastic* matrix, i.e.,

$$\mathcal{D}\begin{pmatrix}1\\1\\\vdots\\1\\1\end{pmatrix} = 1\begin{pmatrix}1\\1\\\vdots\\1\\1\end{pmatrix} \tag{2.5}$$

This is actually a special adjacency matrix such that 1 is an eigenvalue and $\mathbf{1} = \begin{pmatrix}1 & 1 & \dots & 1 & 1\end{pmatrix}^T$ is the corresponding eigenvector, thus ensuring that the row sum of $\mathcal{D}$ equals 1. Indeed, a right stochastic matrix is a real square matrix, with each row summing to 1.

We intend to reach consensus on this dynamics, i.e., for any initial conditions $x_i[0]$ we want to show that $x_i[k] \to x_j[k], \forall v_i \in \mathcal{V}$ and for some $v_j \in \mathcal{V}$ such that $v_j \neq v_i$.

In particular, the solution to the discrete-time system is given by:

$$\begin{pmatrix}x_1[k]\\x_2[k]\\\vdots\\x_5[k]\end{pmatrix} = \mathcal{D}^k \begin{pmatrix}x_1[0]\\x_2[0]\\\vdots\\x_5[0]\end{pmatrix}. \tag{2.6}$$

So, if we make $\mathcal{D}^k$ converge to identical rows, we can eventually show a consensus result.

The proof of the consensus result relies on Gershgorin's disk theorem [119]: if we know the entries of a matrix, we can roughly estimate where the eigenvalues are located. Given, for instance, matrix $\mathcal{D} = [d_{ij}]$, (i) for each row we remove the diagonal entries, then (ii) we take the off-diagonal entries only and we add them up. Eventually, (iii) we use the diagonal entries as a center and we draw n circles with radii:

$$R_i(\mathcal{D}) = \sum_{j=1, j\neq i} |d_{ij}|, \quad i = 1, \dots, n. \tag{2.7}$$

Then, the union of these circles, i.e.,

$$\cup_{i=1}^{n} \{z \in \mathbb{C} : |z - d_{ii}| \leq R_i(\mathcal{D})\} \equiv G(\mathcal{D}), \tag{2.8}$$

tells us where the eigenvalues of $\mathcal{D}$ are located. In other words, the eigenvalues of $\mathcal{D}$ will not fall outside the circle $G(\mathcal{D})$, which, in turn, as shown in Fig. 4.8,

stays entirely within the unit circle, assuming the entries of $\mathcal{D}$ to be those specified by (2.4).

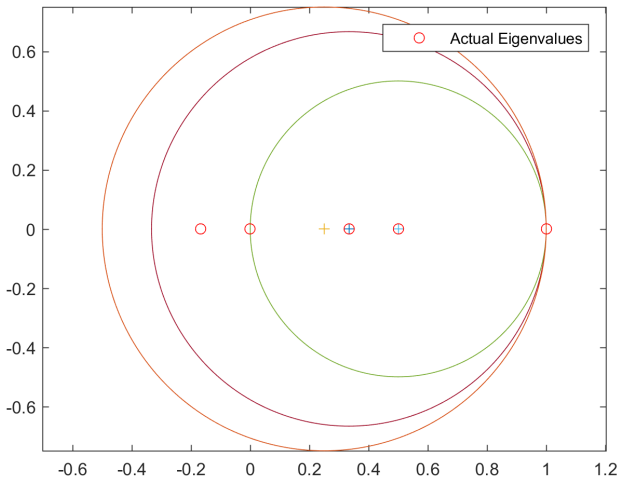


Figure 2.1: Circles with radius $R_i(\mathcal{D}),\ i = 1, \ldots, n$ for matrix $\mathcal{D}$ in equation (2.4) according to Gershgorin's disk theorem.

Note that, in the considered case, we have only one eigenvalue equal to 1. Therefore, it happens that

$$\lim_{k\to\infty} \mathcal{D}^k = \mathbf{p}\mathbf{q}^T \longrightarrow \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \mathbf{q}^T \tag{2.9}$$

where $\mathbf{p}$ is the right eigenvector associated with the eigenvalue in 1 and $\mathbf{q}$ is the left eigenvector associated with the eigenvalue in 1.

Proposition 2.1. *The presence of a spanning tree in the graph $\mathcal{G}$ and the fact that there is at least one strictly positive diagonal entry in $\mathcal{D}$ are necessary and sufficient conditions to have a unique eigenvalue in* 1.

Proof. A *spanning tree* of an undirected graph is a subgraph that is a tree – i.e., an undirected subgraph in which any two nodes are connected by exactly one path – including all the nodes of $\mathcal{G}$, with the minimum possible number of edges. Given this definition, Prop. 2.1 is proven in [119]. □

However, by Gershgorin's disk theorem, we know that 1 is the largest eigenvalue in $\mathcal{D}$. The corresponding right eigenvector is

$$\mathbf{p} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \tag{2.10}$$

whereas the corresponding left eigenvector $\mathbf{q}$ must be such that the sum of its entries equals 1. So, we have

$$\lim_{k\to\infty} \begin{pmatrix} x_1[k] \\ x_2[k] \\ \vdots \\ x_5[k] \end{pmatrix} = \lim_{k\to\infty} \mathcal{D}^k \begin{pmatrix} x_1[0] \\ x_2[0] \\ \vdots \\ x_5[0] \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \mathbf{q}^T \begin{pmatrix} x_1[0] \\ x_2[0] \\ \vdots \\ x_5[0] \end{pmatrix}. \tag{2.11}$$

In this last equation, the product $\mathbf{q}^T\mathbf{x}[0]$ will essentially tell us the final values which the system converges to.

The reason the weighted average works is because $\mathcal{D}$ is a stochastic matrix. If, instead, we zero the diagonal elements of $\mathcal{D}$ (that is, we assume that each agent is not using its own information any more) and we make sure that all elements on the rows add up to 1, we obtain a special stochastic matrix. However, in this case, there might be two eigenvalues equal to 1 with just one independent eigenvector, and the network dynamics would not converge.

Let us consider the undirected graph depicted in Fig. 2.2 in order to show another interesting result concerning where the system converges to. Once again, the link weights are set equal to 1.

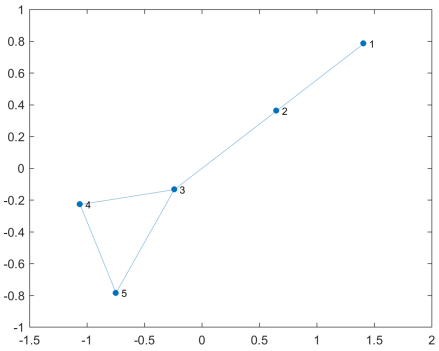


Figure 2.2: An example of undirected graph.

Its $\mathcal{D}$ matrix is the following:

$$\mathcal{D} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}. \tag{2.12}$$

Let us make this a symmetric matrix by adjusting the weights. Under the dynamics (2.4, if $\mathcal{D}$ is symmetric, $\mathcal{D}^k$ for a high value of k (say, e.g., 100) has all identical rows, which means that we have reached consensus among all agents. Let us now write down the matrix after the adjustment to simmetry.

$$\mathcal{D}' = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{pmatrix}. \tag{2.13}$$

Indeed, this time $\mathbf{q}$ is identical to $\mathbf{p}$, i.e.,

$$\mathbf{p}\mathbf{q}^T = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}. \tag{2.14}$$

Instead, if we cut one edge from the graph in Fig. 2.2, i.e., if we consider the graph depicted in Fig. 2.3, $\mathcal{D}^k$ does not converge because the graph does not contain a spanning tree any more.

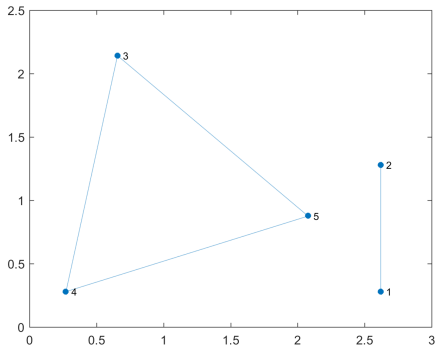


Figure 2.3: The undirected graph in Fig. 2.2 after cutting one edge.

What if the original $\mathcal{D}$ is not such that the row sum is equal to 1? Let us assume that $\mathcal{D}$ is such that the row sum is not equal to 1, then we will have instability for $\mathcal{D}^k$ because the eigenvalues of $\mathcal{D}$ are all outside the unit circle.

We can also consider a directed graph, such as the one in Fig. 2.4.

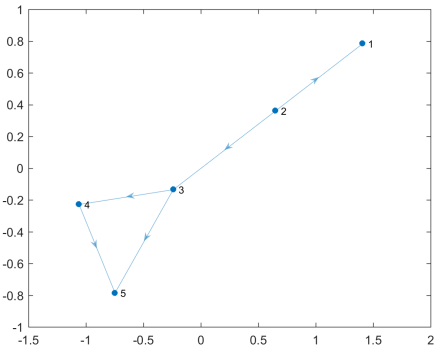


Figure 2.4: A directed graph with a spanning tree rooted in agent 2.

Will the agents achieve consensus in this case? This graph contains a tree. In fact, agent 2 is the only agent which passes information to any other agent: hence, agent 2 dominates and consensus will be achieved at a final value of $x_2[0]$, i.e.,

$$x_i[k] \to x_2[0], \quad \forall i \in \{1, 2, \ldots, 5\}, \tag{2.15}$$

because agent 2 is the root of the spanning tree.

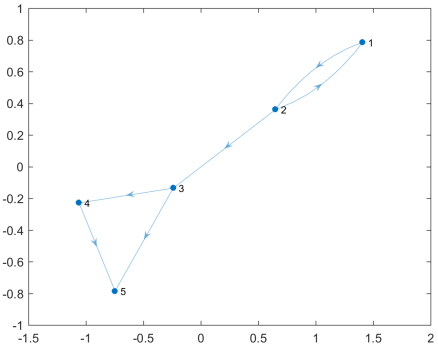


Figure 2.5: Directed graph obtained from the graph in Fig. 2.4 by adding an edge from agent 1 to agent 2.

If we add an edge from agent 1 to agent 2 in the graph shown in Fig. 2.4, we obtain the graph shown in Fig. 2.5, that is, we still reach consensus but this time towards $\frac{1}{2}x_1[0] + \frac{1}{2}x_2[0]$, since agents 1 and 2 contribute to the final consensus value, both of them having paths to any other node in the graph.

So, in the directed graph case, the presence of a spanning tree is a necessary condition for consensus. Strong connection – i.e., the condition when it is possible to reach any node starting from any other node by traversing edges in the directions in which they point – is sufficient but not necessary.

2.1.2 Continuous-Time Consensus in Linear Systems

When dealing with continuous-time systems such as

$$\dot{x} = u, \tag{2.16}$$

if we need to track a constant reference r, we can manage to do it by resorting to a proportional controller $u = -K(x - r)$. The tracking error will then be

$$e = x - r, \tag{2.17}$$

and the related error dynamics will be

$$\dot{e} = -Ke. \tag{2.18}$$

If the reference signal is not constant, the proportional controller $u = -K(x - r)$ will not work any more, as there will be an additional term in the error dynamics, i.e.,

$$\dot{e} = \dot{x} - \dot{r} = -Ke - \dot{r}. \tag{2.19}$$

The only way the proportional controller works is if r is constant.

Based on this, given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ characterized by $n := |\mathcal{V}|$ nodes playing the role of agents, let us consider the multi-agent system

$$\dot{x}_i = u_i, \forall v_i \in \mathcal{V}, \tag{2.20}$$

where x_i is the position of the i-th agent and u_i is the velocity command of the i-th agent. If we want to achieve consensus, we need to design u_i using local information to drive all the x_i's to be identical.

Let us now recall the proportional feedback case. We want the agents' states to converge to the same final value. Yet, we do not know the convergence point *a priori*, otherwise we would set $u_i = -K(x_i - r)$, with r being the desired convergence value. Therefore, we can enforce the following control law:

$$u_i = -\sum_{j \in \mathcal{N}_i} (x_i - x_j) = -\sum_{j \in \mathcal{N}_i} \left(x_i - \frac{\sum_{j \in \mathcal{N}_i \cup \{i\}} x_j}{|\mathcal{N}_i| + 1} \right), \quad \forall v_i \in \mathcal{V}. \tag{2.21}$$

This way, each agent is pushing itself towards one of his neighbours until at some point the agents will stop. It is the same as if each agent pushed itself towards the center between itself and its neighbours. We can also put a weight $\alpha_{ij}(t)$ in the control law, i.e.,

$$\dot{x}_i(t) = -\sum_{j \in \mathcal{N}_i(t)} \alpha_{ij}(t)(x_i(t) - x_j(t)), \quad \forall v_i \in \mathcal{V}. \tag{2.22}$$

By closing the loop and writing this in vector form, we obtain

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{pmatrix} = -L \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{2.23}$$

The $-L$ matrix is the negative Laplacian matrix associated with the underlying graph $\mathcal{G}$: the control law is based on negative feedback, thus the negative Laplacian matrix appears in the closed-loop system dynamics. In compact form, we have:

$$\dot{\mathbf{x}} = -L\mathbf{x}. \tag{2.24}$$

The weights are not necessarily unit values, yet they have to be positive values. Moreover, the row sum in the Laplacian always equals zero. The Laplacian matrix is such that 0 is an eigenvalue with an eigenvector of all ones,

$$L \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = 0 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{2.25}$$

By Gershgorin's disk theorem, all the eigenvalues of the Laplacian matrix fall within the right hand-side of the complex plane. So, $-L$ is such that all its eigenvalues are with non-positive real parts.

The presence of one eigenvalue in zero implies marginal stability. However, if there are two eigenvalues in zero, we cannot say anything in terms of stability.

We now generalize the notion of spanning tree, adopted in undirected graphs according to the matrix-tree theorem (see p. 28 in [119]), to the context of directed graphs, introducing the notion of rooted out-branching.

Definition 2.1. A directed graph $\mathcal{G}$ contains a *rooted out-branching* as a subgraph if it does not contain a directed cycle and if it has a node v_r (i.e., the *root node*) such that for every other node $v \in \mathcal{V}$ there exists a directed path from v_r to v.

□

Hence, the following property holds.

Proposition 2.2. *The graph contains a rooted out-branching if and only if L has a simple zero eigenvalue.*

Proof. Refer to pp. 51-53 in [119] and p. 7 in [155]. □

So, if the system is marginally stable (e.g., L has a simple zero eigenvalue), where does the system converge? We know from linear systems theory that if a generic linear system $\dot{x} = Ax$ is asymptotically stable, then $x(t) \to 0$. Instead, if $\dot{x} = Ax$ is marginally stable, then $x(t) \to \ker\ A$.

The subspace ker A in this case is $span\{\begin{pmatrix}1 & \dots & 1\end{pmatrix}^T\}$. Hence, the system dynamics stops at a set spanned by $\begin{pmatrix}1 & \dots & 1\end{pmatrix}^T$. For ker A to be spanned by $\begin{pmatrix}1 & \dots & 1\end{pmatrix}^T$ only, we need L to have a simple zero eigenvalue, i.e., no more than one eigenvalue in zero, or, equivalently, we need the directed graph to contain a rooted out-branching.

Remark 2.1. If the directed graph is strongly connected, then *a fortiori* we certainly must have L with a simple zero eigenvalue. Hence, the presence of a rooted out-branching is a necessary and sufficient condition for the zero eigenvalue to be simple, whereas strong connection is just a sufficient condition for that to happen.

Let us consider a weighted graph represented by the following Laplacian matrix:

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1.5 & 1.5 & 0 & 0 \\ 0 & -2 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \tag{2.26}$$

Incidentally, this is a lower-triangular matrix. Moreover, the addition/removal of one edge to/from the graph does not change the number of zero eigenvalues.

Hence, summing up, as concerns continuous-time consensus algorithms, the fact that L has a unique eigenvalue in 0 is a necessary and sufficient condition for the presence of a rooted out-branching and for the achievement of consensus.

Instead, as concerns discrete-time consensus algorithms, the fact that $\mathcal{D}$ in (2.4) has a unique eigenvalue in 1 is a necessary and sufficient condition for the presence of a rooted out-branching and for the achievement of consensus.

In particular, the consensus equilibrium value is the following:

$$\lim_{t\to\infty} x_i(t) = \sum_{j=1}^{n} (c_j x_j(0)). \tag{2.27}$$

This is nothing but the weighted average of the initial conditions of the different agents. Strong connection means that everybody has a path to everybody else, thus implying that all agents contribute to the consensus condition with different weights, yielding *weighted average consensus*.

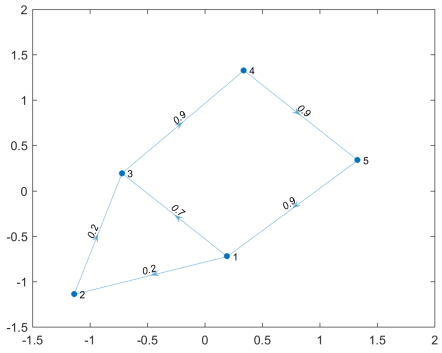


Figure 2.6: A weighted directed graph.

If the graph not only exhibits strong connection but is also balanced, i.e., for every node the total incoming weight is the same as the outcoming weight, then all nodes contribute to the consensus achievement with the same weight, thus allowing the system to reach *average consensus*. Note that an undirected graph is balanced already by default (see, e.g., Fig. 4.2).

As a result of the considerations above, if we have just a rooted out-branching, then only the root node of the rooted out-branching will contribute to the final consensus value!

Definition 2.2. A digraph is said to be *weakly connected* if its disoriented version is connected, that is, if its disoriented version is such that there always exists a path between every pair of nodes and there are no unreachable nodes.

□

Definition 2.3. A digraph is said to be *balanced* if, for every node, the *in-degree* (i.e., the number of head ends adjacent to the considered node) and the *out-degree* (i.e., the number of tail ends adjacent to the considered node) are equal. □

Based on these two definitions, we can state Prop. 2.3.

Proposition 2.3. *For every initial condition* $\mathbf{x}_0$*, the diffusively coupled multi-agent system (2.24) reaches the so-called* average consensus

$$\begin{aligned}\mathbf{x'}_e &= \lim_{t\to\infty}\mathbf{x}(t) = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\mathbf{x}_0 = \\ &= \left(\sum_{i=1}^n \frac{x_{0,i}}{n} \quad \sum_{i=1}^n \frac{x_{0,i}}{n} \quad \cdots \quad \sum_{i=1}^n \frac{x_{0,i}}{n}\right)^T\end{aligned} \tag{2.28}$$

if and only if it is weakly connected and balanced.

Proof. See Theorem 3.17 at p. 56 in [119]. □

Instead, should $\mu(\lambda_1) > 1$, then the multi-agent system (2.24) would converge to a different GAS equilibrium state than (2.28).

Proposition 2.4. *Weak connectedness of the digraph* $\mathcal{G}$ *and the absence of a rooted out-branching in* $\mathcal{G}$ *are sufficient conditions for the algebraic multiplicity of the zero eigenvalue to grow above* 1*, i.e.,* $\mu(\lambda_1) = k > 1$*, yielding* $rank(L) = n - k$*. In such a case, the GAS equilibrium state the diffusively coupled multi-agent system (3.1) converges to is given by*

$$\mathbf{x''}_e = \lim_{t\to\infty}\mathbf{x}(t) = \mathbf{u}_1\mathbf{v}_1^T\mathbf{x}_0 + \mathbf{u}_2\mathbf{v}_2^T\mathbf{x}_0 + \ldots + \mathbf{u}_k\mathbf{v}_k^T\mathbf{x}_0, \tag{2.29}$$

where $\mathbf{u}_1, \ldots, \mathbf{u}_k$ *are* k *distinct and linearly independent eigenvectors associated with the zero eigenvalue of* L *(i.e.,* $L\mathbf{u}_i = 0,\ i = 1, \ldots, k$*) such that* $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ *is a basis of ker* $L := \mathcal{U}$*.*

Note that the GAS equilibrium condition $\mathbf{x''}_e$ does not imply that the same value is reached for all the components of the state vector $\mathbf{x}$, as it occurs, instead, in (3.14) and (2.28). In the following Section, we will therefore investigate how to lead suitable groups (more precisely, cells) of agents to converge each to the same final value, thus yielding multi-consensus.

In particular, the solution to (2.24) is given by:

$$\mathbf{x}(t) = e^{-Lt}\mathbf{x}(0). \tag{2.30}$$

Let us consider a directed graph represented by the following Laplacian matrix:

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 2 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \tag{2.31}$$

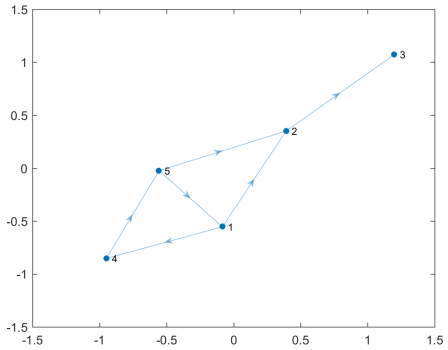


Figure 2.7: Directed graph with Laplacian matrix (2.31)

Note that e^{-Lt} is a stochastic matrix. The columns of L denote the outgoing edges and they determine who contributes to the consensus. Moreover, if L is symmetric, then it means that the graph is undirected.

In particular, the graph represented by the Laplacian matrix (2.31) is shown in Fig. 2.7.

Namely, the graph shown in Fig. 2.7 is such that it has three rooted out-branchings with roots in nodes 1, 4, and 5, respectively.

If we add an edge from node 3 to node 2 to the graph in Fig. 2.7 (as shown in Fig. 2.8), we still do not have a rooted out-branching starting from node 3, so node 3 will not contribute to the final consensus value.

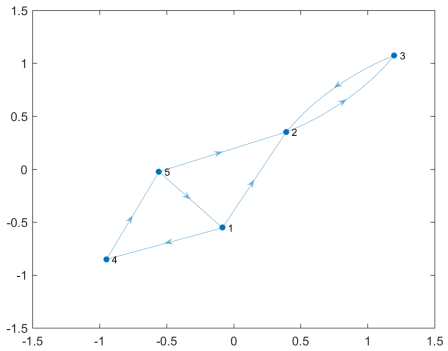


Figure 2.8: Directed graph with Laplacian matrix (2.31) when adding an edge from node 3 to node 2

2.1.3 A Few Hints about Switching Topologies Interconnecting First-Order Systems

As regards continuous time, let $A(t) \in \mathbb{R}^{n\times n}$ and $L(t) \in \mathbb{R}^{n\times n}$ be, respectively, the adjacency matrix and the nonsymmetric Laplacian matrix associated with the directed graph $\mathcal{G}(t) := [\mathcal{V}(t), \mathcal{E}(t)]$. Suppose that $A(t)$ is piecewise-continuous and such that its positive entries are both uniformly lower and upper-bounded (i.e., $a_{ij}(t) \in [\underline{a}, \overline{a}]$, where $0 < \underline{a} < \overline{a}$, if $(j,i) \in \mathcal{E}(t)$ and $a_{ij}(t) = 0$ otherwise). Let $t_0, t_1, \ldots$ be the time sequence corresponding to the times when $A(t)$ switches, assuming that $t_i - t_{i-1} \geq t_L,\ \forall i = 1, 2, \ldots$ with t_L a positive constant. Consensus is reached for the closed-loop system $\dot{x}_i = -\sum_{j=1}^{n} a_{ij}(t)(x_i - x_j)$ if there exists an infinite sequence of contiguous, nonempty, uniformly bounded time intervals $[t_{i_j}, t_{i_{j+1}},\ j = 1, 2, \ldots$, starting at $t_{i_1} = t_0$, with the property that the union of $\mathcal{G}(t)$ across each such interval has a rooted out-branching [155].

As regards discrete time, instead, the interested reader is referred to [104] and [85].

2.2 Second-Order Systems

So far, we have considered a first-order model for agent i, i.e., $\dot{x}_i = u_i$, where x_i is the state and u_i is the control input. We solved the problem of the distributed coordination of the overall multi-agent system by relying on proportional negative feedback control,

$$u_i = -\sum_{j\in\mathcal{N}_i} \alpha_{ij}(x_i - x_j), \quad \forall v_i \in \mathcal{V}, \tag{2.32}$$

implying that, by making each agent push itself towards the center of it and its neighbours, consensus is eventually achieved.

Let us now consider double-integrator systems. The corresponding model is given by:

$$\ddot{x}_i = u_i, \tag{2.33}$$

i.e.,

$$\begin{cases} \dot{x}_i = v_i \\ v_i = u_i. \end{cases} \tag{2.34}$$

In this respect, a differential-drive robot in the inertial frame is shown in Fig. 2.9.

A body frame is then attached to such a differential-drive robot in Fig. 2.10.

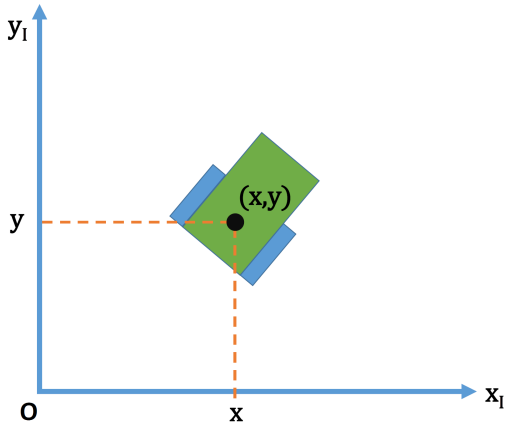


Figure 2.9: A differential-drive robot in the inertial frame.

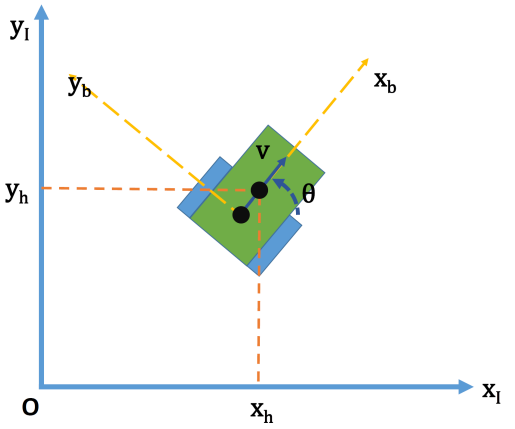


Figure 2.10: A differential-drive robot in its body frame.

The position of the robot in the inertial frame is (x, y). Its orientation is represented by the variable θ. We assume the presence of two control inputs, i.e., the linear speed v and the angular speed ω. In particular, when $v = 1$ and $\omega = 0$, the robot goes straight. Instead, when $v = 0$ and $\omega = const.$, then the robot spins. The situation when $v = const.$ and $\omega = const.$ implies that the robot draws a circle. Moreover, from the point of view of implementation, robots convert linear speed and angular velocity into the velocity of the left wheel on the one side and of the right wheel on the other side, which are then converted into the related motor voltages.

The kinematic model of the considered robot is given by the following

equations:

$$\begin{cases} \dot{x} &= v\cos\theta, \\ \dot{y} &= v\sin\theta, \\ \dot{\theta} &= \omega. \end{cases} \tag{2.35}$$

This is clearly a nonlinear system. We have two control inputs and the system is nonholonomic because there is a nonholonomic constraint. Such a nonholonomic constraint is $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$. This constraint is always true, since a mobile robot cannot move sideways (e.g., a car cannot park by moving sideways)! This means that:

$$\begin{cases} \theta = 0 &\implies \dot{y} = 0, \\ \theta = \frac{\pi}{2} &\implies \dot{x} = 0. \end{cases} \tag{2.36}$$

This causes difficulties in controlling the considered mechanical system. If we want to stabilize such a system, there does not exist a continuous controller driving the state of the system to zero, but we need to use either a discontinuous controller or a time-varying one [22]. It is then difficult to design v and ω so that $x \to 0$ and $y \to 0$. A possible solution consists in controlling the point (x_h, y_h) rather than the center (x, y). In other words, the same system expressed in the coordinates (x_h, y_h) admits a stabilizing continuous controller.

Hence, we change coordinates in the following way:

$$\begin{cases} x_h = x + L\cos\theta, \\ y_h = y + L\sin\theta, \end{cases} \tag{2.37}$$

where L is the axial distance between the two wheels. Such a change of coordinates leads to the following two-dimensional linear system:

$$\begin{cases} \dot{x}_h = \dot{x} + L(-\sin\theta)\dot{\theta} = v\cos\theta - L\omega\cdot\sin\theta, \\ \dot{y}_h = \dot{y} + L(\cos\theta)\dot{\theta} = v\sin\theta + L\omega\cdot\cos\theta, \end{cases} \tag{2.38}$$

which, in compact form, becomes

$$\begin{pmatrix} \dot{x}_h \\ \dot{y}_h \end{pmatrix} = \begin{pmatrix} \cos\theta & -L\sin\theta \\ \sin\theta & L\cos\theta \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \tag{2.39}$$

where we define $\begin{pmatrix} \cos\theta & -L\sin\theta \\ \sin\theta & L\cos\theta \end{pmatrix}$ as $T(\theta)$, which is an invertible matrix. $T(\theta)$ is invertible if $L \neq 0$ because $det(T(\theta)) \neq 0$. Moreover, the following relation holds

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = T^{-1}(\theta) \begin{pmatrix} u_x \\ u_y \end{pmatrix}, \tag{2.40}$$

where u_x and u_y are new inputs to be designed. Hence, we have

$$\begin{pmatrix} \dot{x}_h \\ \dot{y}_h \end{pmatrix} = \begin{pmatrix} u_x \\ u_y \end{pmatrix}. \tag{2.41}$$

This is a simple linear system with a two-dimensional state. Thus, we find ourselves in the same modeling framework as $\dot{x} = u$, except for the fact that x is now two-dimensional.

At this point, our design objective is to drive (x_h, y_h) to the origin by means of proportional feedback, i.e., by relying on

$$\begin{cases} u_x &= -Kx_h \\ u_y &= -Ky_h. \end{cases} \tag{2.42}$$

Alternatively, we may be interested in driving (x_h, y_h) to track (x_r, y_r), i.e., a time-varying reference trajectory, by relying on the control law

$$\begin{cases} u_x &= \dot{x}_r - K(x_h - x_r) \\ u_y &= \dot{y}_r - K(y_h - y_r). \end{cases} \tag{2.43}$$

This last design strategy requires us to know the changing rate of the reference signal because $r(t)$ is assumed to represent a time-varying trajectory. By means of controller (2.43), the (x_h, y_h) point of the robot is guaranteed to follow the specified trajectory $r(t)$ (see Fig. 2.11).

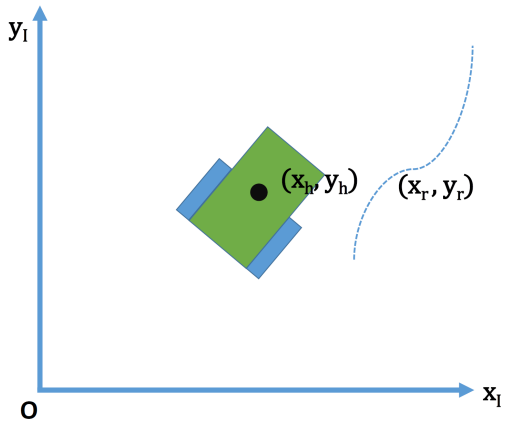


Figure 2.11: A differential-drive robot, under controller (2.43), manages to track the reference trajectory $r(t) = \begin{pmatrix} x_r(t) & y_r(t) \end{pmatrix}^T$.

Bearing this in mind, we can now apply the control law (2.32) to multi-robot systems. By performing the change of coordinates (3.18), we convert a

nonlinear model into a linear model without any approximation. Yet, there are two drawbacks: through (3.18), (i) we actually lose the control orientation, i.e., θ, and (ii) we are considering a position that is slightly off the center. Moreover, through this approach, (iii) we can make the error small, but we still need to check that the inputs do not become extremely big.

The main problem lies in the unobservability of θ, as a result of the transformations made to get from (2.35) to (2.38). So, the dynamics of θ represents the zero dynamics of the entire system. Yet, this zero dynamics has been shown to be stable – even if not asymptotically stable [168] – and this is why the presented design strategy works.

In particular, if $v = const.$ and $w = const.$, the open-loop robot (nonlinear single robot) moves in circle. The design objective is to control (x_h, y_h) in (2.38) so as to drive it according to a trajectory. However, when there is a sharp turn, the limits of the model are shown because the vehicle can go backwards and the controller can only guarantee that (x_h, y_h) tracks correctly.

In order to get from (2.35) to (2.38), we need to apply feedback linearization but it creates the above-mentioned problem from the point of view of implementation, even though it is sure that (x_h, y_h) tracks correctly.

Given these limitations, let the control laws below be applied onto system (2.38), i.e.,

$$\begin{cases} u_{x_i} &= -\sum g_{x_{ij}} k_{ij} (x_{h_i} - x_{j_i}) \\ u_{y_i} &= -\sum g_{y_{ij}} k_{ij} (y_{h_i} - y_{j_i}), \quad \forall v_i \in \mathcal{V}, \end{cases} \tag{2.44}$$

then we have consensus for the considered multi-agent system of robots.

When the *formation control* problem is addressed, instead, we are interested in ensuring that a set of final desired destinations are eventually reached, e.g., $(r_1^d, r_2^d, r_3^d, r_4^d)$ when considering a group of four agents.

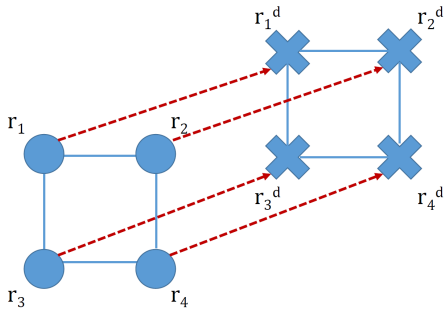


Figure 2.12: An example of formation control.

For instance, with respect to Fig. 2.12, we may need to make the depicted

four agents (i) maintain a square shape, and simultaneously (ii) reach their destination.

Let us assume that the robots start from arbitrary initial positions. Let the robot positions be denoted with the vector

$$r_i = \begin{pmatrix} x_{h_i} \\ y_{h_i} \end{pmatrix}, \quad \forall v_i \in \mathcal{V}. \tag{2.45}$$

Therefore, the four error vectors, measuring the distance of the robot positions from the final desired destinations are represented by $(r_1^d - r_1), (r_2^d - r_2), (r_3^d - r_3), (r_4^d - r_4)$. If these four error vectors,

$$\begin{cases} r_1^d - r_1, \\ r_2^d - r_2, \\ r_3^d - r_3, \\ r_4^d - r_4, \end{cases} \tag{2.46}$$

are kept the same, then the directions are parallel (since the robots start from different initial conditions) and thus the robots maintain the desired formation.

So, the corresponding mathematical model becomes:

$$\dot{r}_i = u_i, \quad \forall v_i \in \mathcal{V}. \tag{2.47}$$

In this respect, if we choose $u_i = -\sum_{j \in \mathcal{N}_i} \alpha_{ij}(r_i - r_j)$, then the agents will rendez-vous, i.e., $r_i \to r_j, \forall i \neq j$. If instead we consider the following design, the differences $(r_i^d - r_i)$ will be pushed towards each other, thus ensuring formation control, i.e.,

$$u_i = -\sum_{j \in \mathcal{N}_i} \alpha_{ij}[r_i - r_i^d - (r_j - r_j^d)] = -\sum_{j \in \mathcal{N}_i} a_{ij}[(r_i - r_j) - (r_i^d - r_j^d)]. \tag{2.48}$$

This way, the agents maintain the desired formation. We can think of this as a consensus algorithm pushing $(r_i - r_i^d)$ to be the same as $(r_j - r_j^d)$ or, equivalently $(r_i - r_j)$ to be the same as $(r_i^d - r_j^d)$, thus maintaining the desired shape.

Nonetheless, in order to get to the desired final destination, we still need to introduce an additional term $-K(r_i - r_i^d)$ for each agent, i.e., to design the following control law:

$$\begin{aligned} u_i &= -\sum_{j \in \mathcal{N}_i} \alpha_{ij}[(r_i - r_i^d) - (r_j - r_j^d)] - K(r_i - r_i^d) = \\ &= -\sum_{j \in \mathcal{N}_i} \alpha_{ij}[(r_i - r_j) - (r_i^d - r_j^d)] - K(r_i - r_i^d), \end{aligned} \tag{2.49}$$

where the first term is in charge of preserving formation and the second term is in charge of ensuring that the final destination is reached. If we care more about the shape, we just have to make K smaller and increase α_{ij}.

Let us now outline the relation between $\begin{pmatrix} v & \omega \end{pmatrix}^T$ and the speed of the wheels, i.e., there is a linear relationship holding in the following form:

$$v = \frac{v_l + v_r}{2}, \tag{2.50}$$

where v is the overall linear velocity, v_l is the linear velocity of the left wheel and v_r is the linear velocity of the right wheel. Instead,

$$\omega = \frac{v_r - v_l}{L}, \tag{2.51}$$

where L stands for the axial distance between the two wheels. In particular, v_l and v_r convert to proper voltages for low-level control.

We can now consider the double-integrator dynamics:

$$\begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = u_i, \end{cases} \tag{2.52}$$

where x_i is the state, and u_i is the acceleration command.

In general, the dynamics of a mobile robot can be represented by the following equation:

$$m\ddot{x} = f_{total}, \tag{2.53}$$

which, more specifically, becomes:

$$\begin{cases} M\dot{v} = F - F_{friction} \\ J\dot{\omega} = \tau - \tau_{friction}. \end{cases} \tag{2.54}$$

For the differential-drive robot, the overall dynamic model is the following:

$$\begin{cases} \dot{x} = v\cos\theta, \\ \dot{y} = v\sin\theta, \\ \dot{\theta} = \omega, \\ M\dot{v} = F - F_{friction}, \\ J\dot{\omega} = \tau - \tau_{friction}. \end{cases} \tag{2.55}$$

This is a sufficiently accurate dynamic model and it is such that the state variables are x, y, θ, v and ω, while the control inputs are F and τ, i.e., forces and torques.

In general, with reference to second-order systems, if we take the kinematic model (2.35) and we perform feedback linearization onto it, we obtain a single integrator in the form (2.41).

Instead, if we take the dynamic model (2.55) and we perform feedback linearization onto it, we obtain a double-integrator structure. The control inputs are F and τ.

We now choose again x_h, y_h in the following way:

$$\begin{cases} x_h = x + L\cos\theta, \\ y_h = y + L\sin\theta. \end{cases} \tag{2.56}$$

By differentiating (2.55) twice (thus calculating $\begin{pmatrix} \ddot{x_h} & \ddot{y_h} \end{pmatrix}^T$), and then by choosing F and τ so as to cancel the nonlinear terms (i.e., performing feedback linearization) in the following way

$$\begin{pmatrix} \frac{F}{m} \\ \frac{L\tau}{J} \end{pmatrix} = \begin{pmatrix} \frac{F_{friction}}{m} + L\omega^2 \\ \frac{L\tau_{friction}}{J} - v\omega \end{pmatrix} + \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix}, \tag{2.57}$$

we get the double-integrator structure:

$$\begin{cases} \ddot{x}_h = u_x \\ \ddot{y}_h = u_y. \end{cases} \tag{2.58}$$

Note that L cannot be zero, otherwise we cannot recover τ. Moreover, the zero dynamics here is proven to be stable [151].

Incidentally, let us now design a consensus algorithm for double integrators. Let us consider a single system, i.e.,

$$\ddot{x} = u, \tag{2.59}$$

or, equivalently,

$$\begin{cases} \dot{x} = v \\ \dot{v} = u. \end{cases} \tag{2.60}$$

If we want to stabilize this system, we can use a PD controller

$$u = -K_1 x - K_2 \dot{x}, \tag{2.61}$$

where K_2 represents the damping term.

The resulting closed-loop system will be

$$\ddot{x} = -K_1 x - K_2 \dot{x}, \tag{2.62}$$

i.e.,

$$\ddot{x} + K_2\dot{x} + K_1 x = 0. \tag{2.63}$$

The corresponding characteristic polynomial is

$$s^2 + K_2 s + K_1 = 0 \tag{2.64}$$

with $K_1 > 0$, $K_2 > 0$, which is necessary according to the Routh criterion. Recall that $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$ is such that the $2\zeta\omega_n s$ term introduces damping. Actually, if we want to stabilize a second-order system, we need some damping term K_2.

Hence, if we apply $u_i = -\sum_{j\in\mathcal{N}_i} \alpha_{ij}(x_i - x_j)$ in a multi-agent second-order system, this will not work because there is no damping. So, we need to introduce an additional damping term

$$u_i = -\sum_{j\in\mathcal{N}_i} \alpha_{ij}(x_i - x_j) - Kv_i, \tag{2.65}$$

where this $-Kv_i$ term represents self-damping negative feedback.

If we have such a damping term, the result is that

$$\begin{cases} x_i \to x_j \text{ (constant)}, \\ v_i \to 0, \end{cases} \tag{2.66}$$

provided that the graph contains a spanning tree and K is large enough. Hence, damping is necessary and should be strong enough.

What if, instead, we use the following control law,

$$u_i = -\sum_{j\in\mathcal{N}_i} \alpha_{ij}(x_i - x_j) - K\sum_{j\in\mathcal{N}_i} \alpha_{ij}(v_i - v_j), \tag{2.67}$$

that is, if we use relative damping? Under this control, we will have

$$\begin{cases} x_i \to x_j, \\ v_i \to v_j \text{ (constant)}, \end{cases} \tag{2.68}$$

provided that the graph contains a spanning tree and that K is large enough. This way, there is damping, but we are pushing one agent's velocity towards his neighbours' velocity. Put another way, the position becomes the same and the velocity becomes the same for all agents.

Let us now outline the difference between the two control laws: under (2.65), the robots meet at the same point and stop (i.e., $v_i = 0$). Under (2.67), the robots meet at the same point and end up moving at the same velocity but may not stop (i.e., $v_i \to v_j$).

Let us carry out some analysis on the multi-agent system subject to (2.65): we plug such a controller into the original model and we take a look at the closed-loop system. We obtain

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \\ \hline \dot{v}_1 \\ \dot{v}_2 \\ \vdots \\ \dot{v}_n \end{pmatrix} = \left(\begin{array}{c|c} 0 & I \\ \hline -L & -KI \end{array}\right) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ \hline v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}, \tag{2.69}$$

i.e.,

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \\ \dot{v}_1 \\ \dot{v}_2 \\ \vdots \\ \dot{v}_n \end{pmatrix} = A \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}, \tag{2.70}$$

with $A = \begin{pmatrix} 0 & I \\ -L & -KI \end{pmatrix}$.

What about the eigenvalues and eigenvectors of A?

$$\begin{pmatrix} 0 & I \\ -L & -KI \end{pmatrix} \begin{pmatrix} w \\ z \end{pmatrix} = \lambda \begin{pmatrix} w \\ z \end{pmatrix} \tag{2.71}$$

We need to solve these two equations:

$$\begin{cases} z = \lambda w \\ -Lw - Kz = \lambda z \end{cases} \tag{2.72}$$

It turns out that the generic eigenvalue of A, λ, is related to the eigenvalues of L. If we replace $z = \lambda w$ in $-Lw - Kz = \lambda z$, we get

$$-Lw - K\lambda w = \lambda^2 w \implies -Lw = (\lambda^2 + K\lambda)w. \tag{2.73}$$

The quantity $(\lambda^2 + K\lambda)$ is therefore an eigenvalue of $-L$ with w as the eigenvector. So, if $\mu_i,\ i \in \{1, \ldots, n\}$ is an eigenvalue of L, then $\lambda^2 + K\lambda = -\mu_i$.

This is a second-order equation, thus implying that every eigenvalue of L (i.e., μ_i) corresponds to two eigenvalues of A, i.e., λ_1 and λ_2. In other words, given $\lambda^2 + K\lambda = -\mu_i$, we solve for λ and find the eigenvalues of A. Each eigenvalue of L corresponds to two eigenvalues of A.

Since L is such that $\mu_1 = 0$ and the others $\mu_i : Re[\mu_i] < 0,\ \forall i \in \{2, \ldots, n\}$, then the zero eigenvalue of L corresponds to two eigenvalues of A: $\lambda_{1+} = 0, \lambda_{1-} = -K$, because $\lambda^2 + K\lambda = 0$ yields $\lambda(\lambda + K) = 0$.

When all eigenvalues have no complex part, if $\mu_i > 0$, we deduce that

$$\lambda^2 + K\lambda + \mu_i = 0 \tag{2.74}$$

is such that all $\lambda_i : Re[\lambda_i] > 0$. So all the eigenvalues of the A matrix have negative real parts, except for the only eigenvalue in the origin, thus implying that A is marginally stable: so, defining $\mathbf{x}(t) = col(x_1(t), x_2(t), \ldots, x_n(t), v_1(t), v_2(t), \ldots, v_n(t))$, we have that $\mathbf{x}(t) \to \ker A$.

Let, instead, the solutions of (2.74) be equal to an imaginary number a_j: in this case, we need $K > K^*$ (for some computable K^*) to make sure that the corresponding eigenvalue falls in the left hand-side of the complex plane. So, if μ_i is complex, then we need a large enough K. If the graph is a directed tree, all the eigenvalues are real and K might be any number (there is no need to make it large enough). Also, if the graph is undirected, then the Laplacian matrix is symmetric and a symmetric matrix has all real eigenvalues, so A is easy. L can therefore have complex eigenvalues in the case of a directed graph only.

When $\mu_i = 0$, then $\lambda_{1+} = 0, \lambda_{1-} = -K$. So, let us now look at the corresponding eigenvector. If $\lambda = 0$, then $-Lw = (\lambda^2 + K\lambda)w = 0w$, thus implying that

$$w = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \tag{2.75}$$

and

$$z = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{2.76}$$

So, the corresponding eigenvector of A is

$$\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{2.77}$$

From linear systems theory, $\begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} \rightarrow Im \begin{pmatrix} \mathbf{1} \\ \hline 0 \end{pmatrix}$. So $x_i \rightarrow x_j$ and all the v_i's will converge to 0. The x's will be the same, the v's will be 0.

Instead, if $K < K^*$, then A will have eigenvalues in the right hand-side of the complex plane and the control strategy (2.65) will not work.

Let us now consider the case of relative damping represented by (2.67): in this case, the dynamic matrix is more complicated.

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \\ \hline \dot{v}_1 \\ \dot{v}_2 \\ \vdots \\ \dot{v}_n \end{pmatrix} = \left(\begin{array}{c|c} 0 & I \\ \hline -L & -KL \end{array}\right) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ \hline v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}, \tag{2.78}$$

i.e.,

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \\ \dot{v}_1 \\ \dot{v}_2 \\ \vdots \\ \dot{v}_n \end{pmatrix} = A' \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}, \tag{2.79}$$

with $A' = \begin{pmatrix} 0 & I \\ -L & -KL \end{pmatrix}$. What about the eigenvalues and eigenvectors of

A'? This time, we need to solve the following equations:

$$\begin{pmatrix} 0 & I \\ -L & -KL \end{pmatrix} \begin{pmatrix} w \\ z \end{pmatrix} = \lambda \begin{pmatrix} w \\ z \end{pmatrix} \implies \tag{2.80}$$

$$\begin{cases} z = \lambda w, \\ -Lw - KLz = \lambda z \implies -Lw - K\lambda Lw = \lambda^2 w. \\ \implies -L(1 + K\lambda)w = \lambda^2 w. \end{cases} \tag{2.81}$$

This implies that

$$-Lw = \frac{\lambda^2}{1 + K\lambda} w. \tag{2.82}$$

Let us assume μ_i as an eigenvalue of L, then $-(1 + K\lambda)\mu_i$ should be equal to λ^2,

$$-(1 + K\lambda)\mu_i = \lambda^2. \tag{2.83}$$

If μ_i is an eigenvalue of L, then $-(1 + K\lambda)\mu_i$ is an eigenvalue of $-L(1 + K\lambda)$. Then

$$\lambda^2 + K\mu_i\lambda + \mu_i = 0 \implies \tag{2.84}$$

$$\lambda = \frac{-K\mu_i \pm \sqrt{(K\mu_i)^2 - 4K\mu_i^2}}{2}. \tag{2.85}$$

This is the relationship between the eigenvalues of A' and those of L. In particular, from this equation we have

$$\mu_1 = 0 \implies \lambda_\pm = 0. \tag{2.86}$$

If μ_1 is 0, there are two corresponding eigenvalues in 0 for A' and they are repeated. The matrix A (under control law (2.65)) had one zero eigenvalue. The matrix A', instead, (under control law (2.67)) has two zero eigenvalues with one independent eigenvector (so, this time the dynamic matrix of the system is unstable). So the system is unstable, but, under (2.67), the relative positions become the same, so the consensus is still reached.

If the eigenvalues of L are positive, then $Re[\lambda] < 0$ where the λ's are the eigenvalues of A' and $K > 0$. If the graph is a tree or undirected, any positive K works. If the graph is arbitrary and $\lambda_\pm = a_j$ with a_j an imaginary number, then we need to set $\lambda_\pm$ and identify the lower bound K^*, above which the algorithm works (i.e., for $K > K^*$).

The overall system is unstable, but the error dynamics (group dynamics) is stable.

So the solution to this linear differential equation is

$$\begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \\ v_1(t) \\ \vdots \\ v_n(t) \end{pmatrix} e^{A't} \begin{pmatrix} x_1(0) \\ \vdots \\ x_n(0) \\ v_1(0) \\ \vdots \\ v_n(0) \end{pmatrix}. \tag{2.87}$$

Provided that (i) the graph contains a spanning tree and (ii) the gain K is large enough, then A' has two zero eigenvalues and all the others are on the left hand-side of the complex plane.

First, $e^{A't}$ can be rewritten in the following way:

$$e^{A't} = Pe^{Jt}P^{-1}, \tag{2.88}$$

where

$$P = \left(\text{generalized eigenvectors arranged by columns}\right), \tag{2.89}$$

$$J = \begin{pmatrix} 0 & 1 & \dots & \dots \\ 0 & 0 & \dots & \dots \\ \dots & \dots & \lambda_3 & \dots \\ \dots & \dots & \dots & \lambda_4 \end{pmatrix}. \tag{2.90}$$

This J matrix has clearly two zero eigenvalues and is such that $Re[\lambda_i] < 0$. At steady state, the blocks associated with the λ_i's vanish. Only the blocks associated with the zero eigenvalues are left. Since

$$\lim_{t\to\infty} e^{Jt} = \begin{pmatrix} 1 & t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{2.91}$$

then

$$\lim_{t\to\infty} e^{A't} = \lim_{t\to\infty} Pe^{Jt}P^{-1} =$$

$$\lim_{t\to\infty} \big(\text{generalized eigenvectors arranged by columns}\big) \begin{pmatrix} 1 & t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times$$

$$\times \big(\text{generalized eigenvectors arranged by rows}\big). \tag{2.92}$$

We only care about the first two columns and rows, since everything else goes to zero. Namely, the first two columns of P are

$$col_1(P) = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{2.93}$$

$$col_2(P) = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \tag{2.94}$$

while the first two rows of P are

$$row_1(P) = \begin{pmatrix} q_1 & \dots & q_n & 0 & \dots & 0 \end{pmatrix}, \tag{2.95}$$

$$row_2(P) = \begin{pmatrix} 0 & \dots & 0 & q_1 & \dots & q_n \end{pmatrix}, \tag{2.96}$$

where $\begin{pmatrix} q_1 & \dots & q_n \end{pmatrix}$ is the left eigenvector of L associated with the zero eigenvalue and $\sum_{i=1}^{n} q_i = 1,\ q_i \geq 0$.

We now recall that

$$L \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = 0 \tag{2.97}$$

where $\begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$ is the right eigenvector of L. Instead,

$$\begin{pmatrix} q_1 & \dots & q_n \end{pmatrix} L = 0 \tag{2.98}$$

and

$$\begin{pmatrix} q_1 & \dots & q_n \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = 1 \tag{2.99}$$

Indeed, with respect to single integrators, the left eigenvector of L determines the final equilibrium point.

We also recall that

$$\lim_{t \to \infty} (\lambda^{-1} A')^m = \mathbf{p}\mathbf{q}^T, \tag{2.100}$$

where $\mathbf{q}$ is the left eigenvector. The left eigenvector of L indeed determines the way we converge to the consensus condition.

It turns out that $row_1(\mathrm{P})$ and $row_2(P)$ are the left generalized eigenvectors of the two zero eigenvalues of A'. Hence, (2.92) converges to

$$\begin{bmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \mathbf{q}^T & t \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \mathbf{q}^T \\ 0 & \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \mathbf{q}^T \end{bmatrix}. \tag{2.101}$$

In the single integrator case, it was

$$\lim_{t\to\infty} e^{-Lt} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \mathbf{q}^T \implies x_i(t) = \sum_{i=1}^{n} q_i x_i(0), \quad \forall v_i \in \mathcal{V}. \tag{2.102}$$

The weights were determined by the graph (i.e., by the eigenvectors of L).

Instead, what will then the final result be in the considered case?

$$\begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \\ v_1(t) \\ \vdots \\ v_n(t) \end{pmatrix} = e^{A't} \begin{pmatrix} x_1(0) \\ \vdots \\ x_n(0) \\ v_1(0) \\ \vdots \\ v_n(0) \end{pmatrix} \implies \tag{2.103}$$

$$\begin{cases} x_i(t) \to \sum_{i=1}^{n} q_i x_i(0) + t \sum q_i v_i(0), \\ v_i(t) \to \sum_{i=1}^{n} q_i v_i(0). \end{cases} \tag{2.104}$$

The agents converge and then they maintain speed according to the Jordan decomposition. The speed they maintain is weighted by the left generalized eigenvectors of L. This is what happens in the case of relative damping.

So far, we have considered fixed graphs only. What happens in the case when the graph is changing? The absolute damping algorithm becomes

$$\begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = -\sum_{j\in\mathcal{N}_i(t)} \alpha_{ij}(x_i - x_j) - K v_i, \end{cases} \tag{2.105}$$

whereas the relative damping algorithm becomes:

$$\begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = -\sum_{j\in\mathcal{N}_i(t)} a_{ij}(x_i - x_j) - K \sum_{j\in\mathcal{N}_i(t)} a_{ij}(v_i - v_j). \end{cases} \tag{2.106}$$

So, when the graph is changing over time, if the union of the graphs (over some time interval) contains a spanning tree, then the absolute damping algorithm (2.65) converges. In such a case, we would have time-varying matrices, i.e., given $L(t)$, we would have that

$$A(t) = \begin{bmatrix} 0 & I \\ -L(t) & -KL(t) \end{bmatrix}, \tag{2.107}$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \vdots \\ \dot{x}_n(t) \\ \dot{v}_1(t) \\ \vdots \\ \dot{v}_n(t) \end{bmatrix} = A(t) \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \\ v_1(t) \\ \vdots \\ v_n(t) \end{bmatrix} . \tag{2.108}$$

The idea behind this approach is that, if the union of the graphs contains a spanning tree sufficiently often, then (2.65) works. Instead, this is not true for (2.67).

For instance, let us consider the time-varying graph shown in Fig. 2.13. We need to check the presence of a spanning tree in the union graph. This

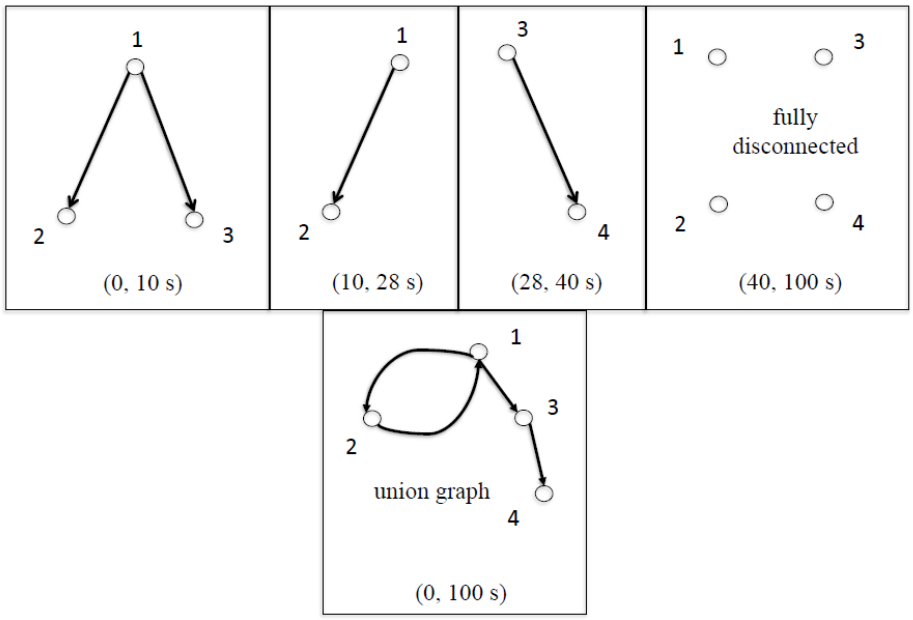


Figure 2.13: An example of time-varying graph. The union graph is reported in the bottom part of the figure.

has to be checked repeated times, on each interval.

If, for any t, there exists a $T > 0$ such that

$$\int_t^{t+T} L(s)ds \tag{2.109}$$

contains a spanning tree, then convergence is preserved. The integral (2.109) represents the union graph analytically. This is the analytical sufficient

condition in continuous time. In discrete time, this is also a necessary condition.

In general, we have verified that, if an agent has no neighbour, then in first-order consensus it stops: it is stable under the considered consensus protocol. Instead, in second-order consensus, if an agent has no neighbour, then it stops under absolute damping, while it does not stop but but keeps moving under relative damping. This is why the condition on the union graph does not work under (2.67).

Let us now consider the case of a switching graph, i.e., of a networked system

$$\dot{x} = A(t)x \tag{2.110}$$

such that $A(t)$ switches within the topologies represented by the set $\{A_1, A_2, \ldots, A_m\}$. Even if each A_i is stable, the whole system

$$\dot{x} = A_i x, \quad i = 1, \ldots, n, \tag{2.111}$$

can still be unstable. This is why there is a lot of ongoing research in the field switching systems. In particular, by enforcing a condition such that the switching time $t_{i+1} - t_i$ is greated than a lower bound – in other words, if we do not jump too fast from one graph to another – we can preserve stability.

Also in the case of a second-order algorithm, we can switch from one graph to another. Let us then define our new states to be the relative errors,

$$\tilde{x} = \begin{bmatrix} x_1 - x_2 \\ x_2 - x_3 \\ \vdots \\ x_{n-1} - x_n \\ v_1 - v_2 \\ \vdots \\ v_{n-1} - v_n \end{bmatrix}, \tag{2.112}$$

thus yielding

$$\dot{\tilde{x}} = \tilde{A}(t)\tilde{x}. \tag{2.113}$$

If we switch from one graph to another without going too fast, we can have stability, i.e., by ensuring that $t_{i+1} - t_i$ stays greater than a suitable lower bound. This is a very conservative result. For this case, however, we require that the graph contains a spanning tree for each time instant, at each switch. We are switching in this case from one graph which can guarantee consensus to another graph which can guarantee consensus. Hence, we just have to make sure that we do not switch extremely fast.

2.2.1 Interconnection of n Mass-Spring-Damper Systems

Let us now consider how a system of n mass-spring-dampers behaves under absolute damping and under relative damping, respectively.

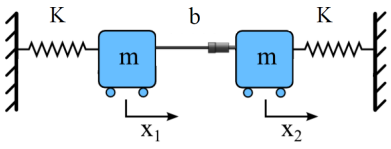


Figure 2.14: A system of $n = 2$ mass-spring-dampers.

What if we push the second mass towards the first mass in Fig. 2.14? The variables x_1 and x_2 are the related coordinates. If we use the absolute position and the relative velocity, we obtain:

$$\begin{cases} m\ddot{x}_1 = -kx_1 - b(\dot{x}_1 - \dot{x}_2) \\ m\ddot{x}_2 = -kx_2 - b(\dot{x}_2 - \dot{x}_1). \end{cases} \tag{2.114}$$

Let $e = x_1 - x_2$, then we have:

$$m\ddot{e} = -ke - 2b\dot{e} \implies m\ddot{e} + 2b\dot{e} + ke = 0. \tag{2.115}$$

This is a classical second-order system and hence we have a second-order polynomial, so that $m, b, k > 0 \implies e \to 0$. Thus, the error converges to 0 and, consequently, the displacement $x_1 - x_2$ will get to 0. Namely, $x_1(t)$ and $x_2(t)$ will have oscillating trajectories. These are indeed couples of harmonic oscillators.

Instead of having just two pieces connected with each other, let us assume that we have n. In such a case, the corresponding model becomes:

$$m\ddot{x}_i = -kx_i - b\sum_{j\in\mathcal{N}_i}(\dot{x}_i - \dot{x}_j), \quad \forall i \in 1, \dots, n \implies \tag{2.116}$$

The total force is the elastic force associated with the spring plus the damping effect relative to the neighbours.

$$\implies \ddot{x}_i = \frac{k}{m}x_i - \frac{b}{m}\sum(\dot{x}_i - \dot{x}_j) \tag{2.117}$$

Thus, comparing with the previous strategy, we are using here absolute position and relative velocity, i.e.,

$$\begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = -\alpha x_i - \sum_{j \in \mathcal{N}_i} \beta_{ij}(v_i - v_j) \end{cases} \tag{2.118}$$

This way, the agents are going to meet (yielding consensus) and oscillate, generating some motion patterns.

Let us assume that x_i is in 2D. If the state is (x, y) and the y-variable is separated from the x-variable by 90 degrees, then each agent oscillates describing a circle. The system dynamics can therefore be written in the following way:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \\ \dot{v}_1 \\ \vdots \\ \dot{v}_n \end{bmatrix} = A \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ v_1 \\ \vdots \\ v_n \end{bmatrix} \tag{2.119}$$

where

$$A = \begin{bmatrix} 0 & I \\ -\alpha I & -L \end{bmatrix}. \tag{2.120}$$

In order to compute the eigenvalues of matrix A, we write:

$$A \begin{bmatrix} w \\ z \end{bmatrix} = \lambda \begin{bmatrix} w \\ z \end{bmatrix} \implies$$

$$\begin{bmatrix} 0 & I \\ -\alpha I & -L \end{bmatrix} \begin{bmatrix} w \\ z \end{bmatrix} = \lambda \begin{bmatrix} w \\ z \end{bmatrix} \implies$$

$$\begin{cases} z = \lambda w \\ -\alpha w - Lz = \lambda z \implies -\alpha w - \lambda L w = \lambda^2 w \implies -\lambda L w = (\lambda^2 + \alpha) w. \end{cases} \tag{2.121}$$

Hence, the eigenvalue for $(-\lambda L)$ is just $(\lambda^2 + \alpha)$. So, defining μ_i as the eigenvalue of L, we have:

$$-\lambda \mu_i = \lambda^2 + \alpha. \tag{2.122}$$

Then,

$$\mu_i = 0 \implies \lambda_{\pm} = \pm \sqrt{\alpha} j, \tag{2.123}$$

where j is the imaginary unit. Namely, the zero eigenvalue maps into two eigenvalues on the imaginary axis. But all the other eigenvalues are mapped

into the left hand-side of the complex plane. So, the system is marginally stable.

Furthermore, the integral of (2.119) can be written as:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \\ v_1(t) \\ v_2(t) \\ \vdots v_n(t) \end{bmatrix} = e^{At} \begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_n(0) \\ v_1(0) \\ v_2(0) \\ \vdots \\ v_n(0) \end{bmatrix} \implies \tag{2.124}$$

$$\implies \begin{bmatrix} \cos(\sqrt{a}t) \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \mathbf{q}^T & -\frac{1}{\sqrt{a}} \sin(\sqrt{a}t) \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \mathbf{q}^T \\ -\sqrt{a} \sin(\sqrt{a}t) \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \mathbf{q}^T & \cos(\sqrt{a}t) \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \mathbf{q}^T \end{bmatrix} \cdot \begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_n(0) \\ v_1(0) \\ v_2(0) \\ \vdots \\ v_n(0) \end{bmatrix}, \tag{2.125}$$

where $\mathbf{q} = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix}$, with $q_i \geq 0, \sum_{i=1}^n q_i = 1$, is once again the left eigenvector of L associated with the zero eigenvalue. Hence, $\forall v_i \in \mathcal{V}$,

$$x_i(t) \to \cos(\sqrt{a}t) \sum_{j=1}^n q_j x_j(0) - \frac{1}{\sqrt{\alpha}} \sin(\sqrt{\alpha}t) \sum_{j=1}^n q_j v_j(0), \tag{2.126}$$

$$v_i(t) \to -\sqrt{\alpha} \sin(\sqrt{\alpha}t) \sum_{j=1}^n q_j x_j(0) + \cos(\sqrt{\alpha}t) \sum_{j=1}^n q_j v_j(0). \tag{2.127}$$

So, the solutions agree both on the position and on the velocity. The agents reach consensus but they do not stop, instead they oscillate – due to the presence of cosines and sines, recall the fundamental relation $a\cos(\omega t) + b\sin(\omega t) = \sqrt{a^2+b^2}\cos(\omega t + \phi)$. Namely, $\sqrt{\alpha}$ tells us how fast the whole system is oscillating. The magniture of the oscillations is instead given by $\sqrt{a^2+b^2}$ which in (2.127) corresponds to:

$$\sqrt{\left(\sum_{j=1}^n q_j x_j(0)\right)^2 + \left(\frac{1}{\sqrt{\alpha}} \sum_{j=1}^n q_j v_j(0)\right)^2}. \tag{2.128}$$

Let us now consider the case of leader-following coordination of a multi-agent system. Let us assume that we have a leader agent characterized by some dynamics, e.g., $(\dot{x}_0 = v_0,\ \dot{v}_0)$. The leader agent (i.e., agent 0) does not receive any information, but he just gives information out, as it is assumed to be the only root node in the considered graph. In this case, all the agents

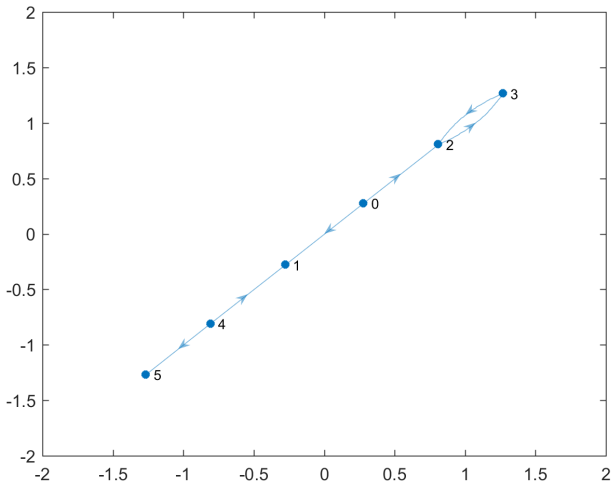


Figure 2.15: An example of leader-following topology.

will converge to the path dictated by agent 0, because the only spanning tree in this graph has root in node 0. Let the dynamics of agent 0 will be

$$\begin{cases} \dot{x}_0 = v_0 \\ \dot{v}_0 = -\alpha x_0, \end{cases} \tag{2.129}$$

with initial conditions $x_0(0)$ and $v_0(0)$. The initial trajectory can be specified so that the x-component is $\cos(t)$ and the y-component is $\sin(t)$. The solution for agent 0 is then:

$$\begin{aligned} x_0(t) &= \cos(\sqrt{\alpha}t)x_0(0) + \frac{1}{\alpha}\sin(\sqrt{\alpha}t)v_0(0) \\ v_0(t) &= -\sqrt{\alpha}\sin(\sqrt{\alpha}t)x_0(0) + \cos(\sqrt{\alpha}t)v_0(0). \end{aligned} \tag{2.130}$$

With such a topology, everybody will follow this leader. By tuning $x_0(0)$ and $v_0(0)$, we specify the trajectory. The agents follow the circles in synchrony.

What if the graph is changing? In the case of switching graphs, we need to check that each graph contains a spanning tree. If this is true, then the

algorithm works. If the graphs are strongly connected and balanced and α is fixed, we do not even care about how fast we switch. All these algorithms can also be applied to differential-drive robots after feedback linearization.

2.3 Consensus in 2D with Rotation Matrices

Let us recall consensus algorithms for single-integrator systems. We consider:

$$\begin{aligned} \dot{x}_i &= u_i \\ u_i &= -\sum_{j\in\mathcal{N}_i} a_{ij}(x_i - x_j), \end{aligned} \tag{2.131}$$

where x_i can be a scalar, or it can belong to $\mathbb{R}^2$, etc. depending on the application. Hence, $x_i \in \mathbb{R}$, or $x_i \in \mathbb{R}^2$, or $x_i \in \mathbb{R}^3$. etc. We now choose x_i to be a vector. For instance, let x_i represents the position in 2D:

$$x_i = \begin{pmatrix} r_{x_i} \\ r_{y_i} \end{pmatrix}. \tag{2.132}$$

In this case, we have an x-coordinate and a y-coordinate.

Let us now change controller (2.32) (adopted above for single-integrator systems) into

$$u_i = -R(\theta)\sum_{j\in\mathcal{N}_i} a_{ij}(x_i - x_j). \tag{2.133}$$

where $R(\theta)$ is a rotation matrix premultiplying the sum of the position differences. In 2D,

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{2.134}$$

Given a generic vector, if we multiply it by $R(\theta)$ we are rotating it clockwise. The properties of rotation matrices are (i) $det(R(\theta)) = 1$ and (ii) there are two complex conjugate eigenvalues, $e^{j\theta}, e^{-j\theta}$. When we multiply by a rotation matrix, we are rotating, we are changing direction a little bit. Without $R(\theta)$, we will always move towards the center of our neighbours without rotating (i.e., algorithm (2.32)). If we multiply by $R(\theta)$, the agents will continuously adjust their angle, they will not go straight towards the consensus condition but they will converge with a spiral movement (i.e., algorithm (2.133)), yet only provided that θ is kept small enough. If θ increases, the agents will move around periodic orbits without converging. In other words, if θ goes above some critical value, then the agents may spiral out, causing instability.

Incidentally, in 3D, we have a rotation matrix $R(a, \theta)$ where a is the axis around which the rotation occurs (axis and angle determine this 3×3 matrix).

If $a = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T$, we will be rotating about the z-axis. The inverse of R is equal to the transpose of R. We will rotate vector v about axis a with angle θ when we multiply v by $R(a, \theta)$. Moreover, $R(a, \theta)$ in 3D has three eigenvalues: $1, e^{j\theta}, e^{-j\theta}$.

Before applying the control algorithm (2.133), let us introduce the Kronecker product, denoted with the $\otimes$ symbol: given

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & \dots & a_{33} \end{bmatrix}, \tag{2.135}$$

and

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, \tag{2.136}$$

the Kronecker product consists in multiplying each entry of A by the entire B matrix, so we have:

$$A \otimes B = \begin{bmatrix} a_{11} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} & a_{12} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} & a_{13} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \\ \dots & \dots & \dots \\ \dots & \dots & a_{33} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \end{bmatrix}. \tag{2.137}$$

Namely, the eigenvalues of $A \otimes B$ are equal to the eigenvalues of A times the eigenvalues of B, i.e., $\lambda_i(A) \cdot \lambda_j(B), \forall i, j$: hence, each eigenvalue of A gets multiplied by each eigenvalue of B. All these will be the eigenvalues of the Kronecker product. E.g., if the eigenvalues of A and B are

$$\sigma(A) = \{1, 2\}, \tag{2.138}$$

$$\sigma(B) = \{3, 4\}, \tag{2.139}$$

then the eigenvalues of $A \otimes B$ are

$$\sigma(A \otimes B) = \{1 \cdot 3, 1 \cdot 4, 2 \cdot 3, 2 \cdot 4\}. \tag{2.140}$$

So, the closed-loop dynamics under (2.32) can be written as:

$$\begin{pmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{pmatrix} = -(L \otimes I_m) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad x_i \in \mathbb{R}^m. \tag{2.141}$$

This is a convenient way of representing consensus on vectors because x_i is a vector and not a scalar, otherwise $L_{n \times n}$ would not be multipliable by each

component of $\mathbf{x}$ which is an $m \times 1$ vector. In particular, the eigenvalues of $L \otimes I_m$ are the eigenvalues of L, each repeated m times.

Let us now, instead, discuss the closed-loop dynamics under (2.133), i.e.,

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{pmatrix} = -(L \otimes R(\theta)) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad x_i \in \mathbb{R}^m. \tag{2.142}$$

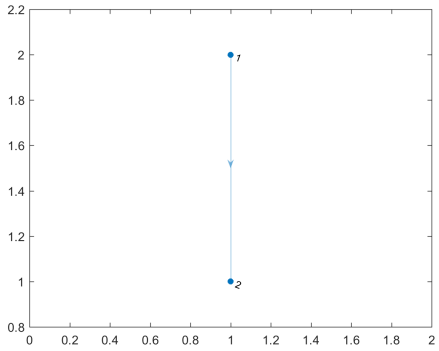


Figure 2.16: A simple example of directed graph.

For instance, if we consider the very simple directed graph shown in Fig. 2.16, for agent 2 we have

$$u_2 = -R(\theta)(x_2 - x_1). \tag{2.143}$$

Note that

$$L \otimes R(\theta) = \begin{bmatrix} 0_{(2\times 2)} & 0_{(2\times 2)} \\ -R(\theta) & R(\theta) \end{bmatrix}. \tag{2.144}$$

By computing this Kronecker product, we multiply each entry of L by $R(\theta)$, thus making all the eigenvalues rotate. So, the eigenvalues of the Kronecker product will be exactly the eigenvalues of L times the eigenvalues of $R(\theta)$. Depending on θ, we have different behaviours (e.g., as periodic orbits) if after the multiplication the eigenvalues fall on the imaginary axis.

Note also that

$$-(L \otimes R(\theta)) = (-L) \otimes R(\theta) \tag{2.145}$$

The eigenvalues of $-L$ are all with negative real parts except for one eigenvalue located in the origin if the graph contains a spanning tree. Instead, $R(\theta)$

has the three above-mentioned eigenvalues. Hence, $(-L) \otimes R(\theta)$ will have twelve eigenvalues, each eigenvalue of L multiplied by each eigenvalue of $R(\theta)$. Multiplying by $e^{-j\theta}$ means to rotate by θ counter-clockwise. Multiplying by $e^{j\theta}$, instead, means to rotate by θ clockwise.

Instead, what about the eigenvectors? Even if we repeat the eigenvalues, we also multiply the eigenvectors. The eigenvectors of $(-L) \otimes R(\theta)$ are the eigenvectors of $(-L)$, each, Kronecker multiplied by each eigenvector of $R(\theta)$. So, the new eigenvectors are all independent. So, is the new system stable? If θ is small, then the new system keeps marginally stable.

What happens if we rotate by a large amount of degrees? The angle may be so big as to induce new eigenvalues on the imaginary axis, thus generating oscillations (and consequently periodic orbits). If θ is extremely big, it may occur that the new eigenvalues fall in the right hand-side of the complex plane, thus causing the system to go unstable (i.e., the agents will spiral out). This is what happens for first-order systems.

Let us see what happens for second-order systems.

$$\ddot{x}_i = u_i \quad \text{or, equivalently,} \quad \begin{cases} \dot{x}_i = v_i, \\ \dot{v}_i = u_i. \end{cases} \tag{2.146}$$

Let us adopt the control law acting on relative position plus absolute damping, i.e.,

$$u_i = -R(\theta) \sum_{j \in \mathcal{N}_i} a_{ij}(x_i - x_j) - K v_i. \tag{2.147}$$

We introduce, in a similar way as before, the rotation matrix, thus yielding:

$$u_i = -R(\theta) \sum_{j \in \mathcal{N}_i} a_{ij}(x_i - x_j) - K R(\theta) \sum_{j \in \mathcal{N}_i} a_{ij}(v_i - v_j). \tag{2.148}$$

Previously, we said that, if θ is small, then the agents spiral in. If, instead, $\theta = \theta^*$ (where θ^* identifies a critical angle accounting for marginal stability), they describe periodic orbits; if θ is big, they spiral out.

Moreover, by relying on $R(a, \theta)$, given the axis a of rotation, the agents can move on a plane perpendicular to this axis. The center of the orbits is the weighted average of the initial conditions. If we want to monitor something on the ground, we need to move on a plane perpendicular to the z-axis. What if you want to explicitly specify the target location to monitor in order to circle around that target? We fix the initial condition of one agent as the point you want to monitor, then you make sure that there is only one spanning tree rooted in that agent. Under (2.148), without the rotation, the agents converged and then kept going due to relative damping. With rotation, the

agents keep tracking a circular orbit but the center moves. The Laplacian still plays an important role here. The center of the orbits is determined by the consensus algorithm. In the second-order algorithm, without rotation, previously we had that:

$$A = \begin{bmatrix} 0 & I \\ -L & KI \end{bmatrix} \tag{2.149}$$

Now, with the rotation, we have

$$A' = \begin{bmatrix} 0 & I \\ -L \otimes R(\theta) & KI \end{bmatrix} \tag{2.150}$$

We can also introduce a leader in any of these cases: we just have to make it the root of the only directed spanning tree in the graph.

2.4 Lyapunov-based Analysis and Control of Nonlinear Models

Nonlinear multi-agent systems will be now investigated by means of Lyapunov stability theory.

Given a single-integrator

$$\dot{x} = u, \tag{2.151}$$

let us assume that we want to drive x to the origin. We can do that by relying on the control law

$$u = -Kx. \tag{2.152}$$

The closed-loop system is therefore

$$\dot{x} = -Kx. \tag{2.153}$$

Hence,

$$x(t) = e^{-Kt}x(0) \tag{2.154}$$

which goes to zero exponentially. So, this controller works. However, if the initial error is big, our control error can be huge!

Instead, we could use the following controller:

$$u = -Ksat(x), \tag{2.155}$$

with

$$sat(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0, \\ -1, & x < 0. \end{cases} \tag{2.156}$$

The closed-loop system dynamics becomes

$$\dot{x} = -Ksat(x). \tag{2.157}$$

What is the solution to this differential equation? We can use a Lyapunov candidate function instead.

$$V = \frac{1}{2}x^2. \tag{2.158}$$

The adoption of such a Lyapunov function yields

$$\dot{V} = \frac{\partial V}{\partial x}\dot{x} = x\dot{x} = x(-Ksat(x)) = -Kxsat(x) < 0 \quad x \neq 0. \tag{2.159}$$

When x is positive, this $\dot{V}$ returns a negative value. When x is negative, this $\dot{V}$ returns a negative value too!

So $\dot{V} < 0, \forall x \neq 0$.

Let us now review some aspects of nonlinear systems and control which will turn useful for investigating multi-agent system stability in the nonlinear domain.

Given an LTI system $\dot{x} = Ax$, in general stability is determined by analyzing the eigenvalues of A. Instead, if we have a nonlinear system, say $\dot{x} = f(x)$, there is no matrix, there are no eigenvalues. It is also hard to figure out the solution. This $\dot{x} = f(x)$ is a closed-loop system, so the control action is already embedded. Let us compare an LTI system with a nonlinear one. The former can be represented as

$$\dot{x} = A_1 x + Bu \tag{2.160}$$

with control law

$$u = -Kx, \tag{2.161}$$

yielding

$$\dot{x} = (A_1 - BK)x. \tag{2.162}$$

This is the closed-loop LTI system. Instead,

$$\dot{x} = f_1(x, u) \tag{2.163}$$

with control law

$$u = \phi(x) \tag{2.164}$$

yields

$$\dot{x} = f(x) \tag{2.165}$$

This is the closed-loop nonlinear system.

Let us analyze the stability of the system. In classical control, stability implies that for any bounded input signal, the output will be bounded too. But in nonlinear systems and Lyapunov control, stability is intepreted in terms of the system behaviour with respect to the equilibrium conditions. In general, an equilibrium condition is identified by a point in the state space such that, if the state starts at that point, the state will remain there. Hence, when we investigate stability in nonlinear systems, we need to aim at an equilibrium position.

For instance, let us assume that we want to keep a pendulum at 45°. An angle of 45° is not an equilibrium point for the pendulum. So, we must use feedback in order to produce a force such that the pendulum is kept at 45°.

How to find equilibrium points? We just have to solve $f(x) = 0$ and, therefore, look for the set $\{x : f(x) = 0\}$. In general, the equilibrium point might not be 0, but it can be made such by just a simple variable change.

So, we can state that the equilibrium point $x = 0$ of $\dot{x} = f(x)$ is

- stable if $\forall \varepsilon > 0, \exists \delta = \delta(\varepsilon) > 0$ such that $\|x(0)\| < \delta \implies \|x(t)\| < \varepsilon, \quad \forall t \geq 0$;
- unstable if it is not stable;
- asymptotically stable if it is stable and if $\exists \delta > 0$ such that, if $\|x(0)\| < \delta \implies \lim_{t\to\infty} \|x(t)\| = 0$.

Asymptotic stability means that the trajectory is not just stable but will eventually approach the equilibrium point. The Lyapunov approach allows to test this definition.

Let us now recall the concept of Lyapunov stability. Let $x = 0$ be an equilibrium point of $\dot{x} = f(x)$. Let $D \subset \mathbb{R}^n$ be a domain containing $x = 0$. Let $V(x)$ be a continuously differentiable function such that $V(0) = 0$ and $V(x) > 0$ in $D \setminus \{0\}$, and $\dot{V}(x) \leq 0$ in D. This implies that $x = 0$ is stable. Instead, $\dot{V}(x) < 0$ in $D \setminus \{0\}$ implies that $x = 0$ is asymptotically stable.

Let us say for some reason that we are able to generate a plot and show all the possible ways the trajectory will move from any initial condition. Then, by means of such simulations we can certainly determine if the system is stable or not! But this is feasible just in 2D. In 3D, this is not feasible any more. In this respect, Lyapunov helps us. Namely, $\dot{V}(x) \leq 0$ in D means that V can never grow above its initial value.

This is just a sufficient condition which, by the way, is very useful for studying a lot of physical systems, easily offering insight about the energy-like function represented by the Lyapunov candidate V.

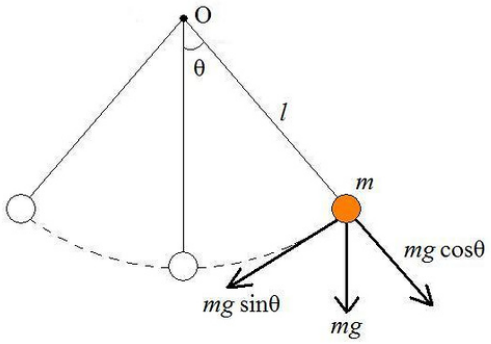


Figure 2.17: A pendulum.

The dynamics of the pendulum shown in (2.17) is given by:

$$ml\ddot{\theta} = -mg\sin\theta - kl\dot{\theta} \tag{2.166}$$

This is a nonlinear system because of the $\sin\theta$ term.

Let us choose the state as

$$\begin{cases} x_1 = \theta, \\ x_2 = \dot{\theta}, \end{cases} \tag{2.167}$$

then the state space representation of such a system becomes:

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = \frac{g}{l}\sin x_1 - \frac{k}{m}x_2, \end{cases} \tag{2.168}$$

with $x_2 \in (-\pi, \pi)$.

This is the classical mathematical model for the pendulum adopted in systems theory.

The equilibrium points are $(0,0)$ and $(\pi,0)$. For the equilibrium point $(0,0)$, let us carry out some analysis by means of the Lyapunov approach.

For simplicity, let us consider the following model

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -a\sin x_1 - bx_2, \end{cases} \tag{2.169}$$

with $a > 0$, $b > 0$. By experience, in general $V(x) = \frac{1}{2}\sum_{i=1}^{n} x_i^2$ with $x \in \mathbb{R}^n$. In this case, we choose $V = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2$. This is such that $V(0) = 0$. Therefore,

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial x}\dot{x} = \frac{\partial V}{\partial x_1}\dot{x}_1 + \frac{\partial V}{\partial x_2}\dot{x}_2 = x_1\dot{x}_1 + x_2\dot{x}_2 = \\ &= x_1x_2 + x_2(-a\sin x_1 - bx_2) = x_1x_2 - ax_2\sin x_1 - bx_2^2. \end{aligned} \tag{2.170}$$

We do not like the first two terms. Instead, the third term is always less than 0. So the function V we suggested does not work because it cannot tell us what is going on in the considered system.

What if we change the Lyapunov function V? For instance, we can choose:

$$V = a(1 - \cos x_1) + \frac{1}{2}x_2^2, \tag{2.171}$$

where the first term is positive because $\cos x_1 \in [-1, 1]$ and the second term is positive too.

By deriving we obtain

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial x_1}\dot{x_1} + \frac{\partial V}{\partial x_2}\dot{x_2} = a \sin x_1 \dot{x_1} + \frac{\partial V}{\partial x_2}\dot{x_2} = \\ &= a \sin x_1 x_2 + x_2(-a \sin x_1 - b x_2) = \\ &= a \sin x_1 \cdot \dot{x_2} - a x_2 \cdot \sin x_1 - b x_2^2 = -b x_2^2 \leq 0. \end{aligned} \tag{2.172}$$

Now the first two terms cancel out. The third term is always negative.

This is only negative semi-definite because x_1 is not involved in the above inequality. The variable x_1 does not show here, so we need to adjust the V if we want to show negative definiteness. We can therefore choose

$$V = \frac{1}{2}\begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{2.173}$$

Our approach is to select p_{11}, p_{12}, p_{21} and p_{22} in order to cancel the indefinite terms in $\dot{V}$ until we end up in a situation where $\dot{V} < 0$.

Another useful approach is, instead, La Salle's invariance principle. In (2.172) we had

$$\dot{V} = -b x_2. \tag{2.174}$$

So, we have that

$$\dot{V} = 0 \implies x_2 = 0. \tag{2.175}$$

Let us say that at some point $x_2 = 0 \implies \dot{V} = 0$. Recall system (2.169). If $x_2 = 0$, $\dot{x_2}$ is not necessarily equal to 0, so $\dot{x_2}$ might be changing. Since $\dot{V} < 0$, over time, if $x_2 = 0$, $x_1 \neq 0$, then V decreases.

In particular, the only way we can make V stop decreasing is by setting

$$\begin{cases} x_1 = 0 \\ x_2 = 0. \end{cases} \tag{2.176}$$

So, the system must be asymptotically stable. This is the rationale behind La Salle's invariance principle:

$$\dot{V} = 0 \implies x_2 = 0. \tag{2.177}$$

If$x_2 = 0$, then

$$\text{setting } \dot{x}_2 = 0 \implies x_1 = 0. \tag{2.178}$$

The only way to make $\dot{x}_2 = 0$ is to zero x_1.

In general, La Salle's invariance principle can be stated as follows.

Theorem 2.5. *[96] Let $\Omega \subset D$ be a compact set that is positively invariant with respect to $\dot{x} = f(x)$. Let $V : D \to \mathbb{R}$ be a continuously differentiable Lyapunov candidate function such that $\dot{V} \leq 0$ in Ω. Let E be the set of all points in Ω where $\dot{V} = 0$. Let M be the largest invariant set in E. Then every solution starting in Ω approaches M as $t \to \infty$.*

Let us now extend this machinery to multi-agent systems:

$$\begin{cases} \dot{x}_i = u_i \\ u_i = -\sum_{j \in \mathcal{N}_i} a_{ij}(x_i - x_j), \end{cases} \tag{2.179}$$

i.e., in vector form,

$$\dot{\mathbf{x}} = -L\mathbf{x}. \tag{2.180}$$

For this linear system, we can use a nonlinear algorithm to show convergence.

We propose a quadratic Lyapunov candidate function:

$$V(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{x} = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 + \ldots + \frac{1}{2}x_n^2. \tag{2.181}$$

Hence, its derivative along the system trajectories is

$$\dot{V} = \left(\frac{\partial V}{\partial \mathbf{x}}\right)^T \dot{\mathbf{x}} = \mathbf{x}^T\dot{\mathbf{x}} = -\mathbf{x}^T L\mathbf{x}. \tag{2.182}$$

Now, if the graph is undirected, L is symmetric. The eigenvalues of L are on the right hand-side of the complex plane plus at least one eigenvalue in the origin. This implies that L is symmetric and positive semi-definite. So $\dot{V}$ is negative semidefinite. If the graph is undirected and connected, then L has a simple zero eigenvalue, yielding

$$\dot{V} = -\mathbf{x}^T L\mathbf{x} \leq 0. \tag{2.183}$$

This way, we show stability, but not asymptotic stability and hence no consensus is proved yet.

We therefore resort to La Salle's invariance principle. The expression $\mathbf{x}^T L\mathbf{x}$ be equal to 0 only when

$$\mathbf{x}^T L\mathbf{x} = 0 \implies \mathbf{x} = c \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{2.184}$$

The vector of all ones is the eigenvector associated to the zero eigenvalue. If the graph is not connected, the first equality is still true, but the second equality is not.

Yet, so far, we have only assumed the graph as un undirected graph. For undirected graphs, this is how we show asymptotic stability by Lyapunov thanks to the fact that L is symmetric. For directed graphs, using the Lyapunov algorithm is more complicated because L is not symmetric.

Indeed, in the case of directed graphs, L is not symmetric. The quantity

$$x^T Lx = x^T \frac{L + L^T}{2} x \tag{2.185}$$

is not non-negative anymore, because L is not symmetric.

Given a symmetric matrix A, it is said to be Positive Definite (PD) if and only if the eigevalues of $(A + A^T) > 0$. Instead, it is Positive SemiDefinite (PSD) if and only if the eigenvalues of $(A + A^T) \geq 0$.

$$\begin{aligned} &x^T Ax > 0 \ \forall x \neq 0 \iff PD \iff eigs(A + A^T) > 0 \\ &x^T Ax \geq 0 \ \forall x \iff PSD \iff eigs(A + A^T) \geq 0 \end{aligned} \tag{2.186}$$

If A is not symmetric, instead we exploit the following relation:

$$x^T Ax = x^T \frac{A + A^T}{2} x, \tag{2.187}$$

and work on $\frac{A+A^T}{2}$.

The problem here is that $\frac{A+A^T}{2}$ is not a Laplacian matrix, which makes the directed case more complicated to deal with.

The Lyapunov approach usually works well in the undirected case. If the graph is directed, we can recover a similar proof only if the graph is strongly connected and balanced, resorting to the following relation:

$$x^T Ax = x^T \frac{A + A^T}{2} x = \frac{x^T Ax}{2} + \frac{x^T A^T x}{2}. \tag{2.188}$$

The first term is a scalar and is equal to the second.

Let us now use a nonlinear control algorithm

$$u_i = -\sum_{j\in\mathcal{N}_i} a_{ij}\phi(x_i - x_j), \quad \forall v_i \in \mathcal{V}. \tag{2.189}$$

Let ϕ be an odd function, i.e.,

$$\begin{cases} \phi(x) & = -\phi(-x) \\ \phi(0) & = 0. \end{cases} \tag{2.190}$$

For instance, we can interpret ϕ as a saturation function in order to avoid large control errors. This could allow to enforce a constraint such as the norm of u_i being upper bounded by a constant number. By introducing a saturation function we are pre-specifying the upper bound for the control effort, and this is independent from the initial conditions of x_i and x_j.

Given $\dot{x}_i = u_i$, under (2.189), the closed-loop system becomes

$$\dot{x}_i = -\sum_{j\in\mathcal{N}_i} a_{ij}\phi(x_i - x_j), \quad \forall v_i \in \mathcal{V}. \tag{2.191}$$

Does this system reach consensus? The Laplacian matrix this time is no longer a suitable choice for showing this. So, let us use a simpler Lyapunov candidate function:

$$V = \frac{1}{2}\mathbf{x}^T\mathbf{x} = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 + \ldots + \frac{1}{2}x_n^2. \tag{2.192}$$

The derivative of this Lyapunov candidate function along the system trajectories is:

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial x_1}\dot{x}_1 + \frac{\partial V}{\partial x_2}\dot{x}_2 + \ldots + \frac{\partial V}{\partial x_n}\dot{x}_n = \\ &= x_1\dot{x}_1 + x_2\dot{x}_2 + \ldots = \\ &= x_1\Big(-\sum_{j\in\mathcal{N}_i} a_{ij}\phi(x_i - x_j)\Big) + \ldots + x_n\Big(-\sum_{j\in\mathcal{N}_n} a_{nj}(x_n - x_j)\Big). \end{aligned} \tag{2.193}$$

Let us assume that the graph is undirected and connected. Let us say that agent 2 is a neighbour of agent 1. Because the graph is undirected, its Laplacian is symmetric. So, (2.193) becomes

$$\dot{V} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i} a_{ij}(x_i - x_j)\phi(x_i - x_j) \leq 0. \tag{2.194}$$

Here, the product $(x_i - x_j)\phi(x_i - x_j)$ is always positive due to the fact that ϕ is an odd function. So, $\dot{V} \leq 0$.

But this goes to 0 only when the arguments of the odd function go to zero.

More precisely, $x_i \to x_j, \forall i, j$, only if the graph is connected. Instead, if the graph is not connected, only the connected parts of the graph are such that $x_i \to x_j$.

For instance, let us consider the very simple example of disconnected graph depicted in Fig. 2.18 below. In this case, equation (2.194) still holds,

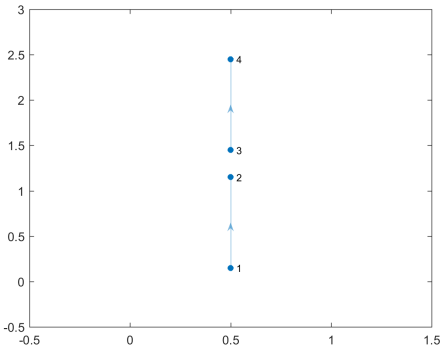


Figure 2.18: An example of disconnected graph.

but, for node 1, we consider only node 2 as its neighbour. Similarly, for node 3, we consider only node 4 as its neighbour:

$$-\frac{1}{2}a_{12}(x_1 - x_2)\phi(x_1 - x_2) - \frac{1}{2}a_{21}(x_2 - x_1)\phi(x_2 - x_1). \tag{2.195}$$

Hence, we will have

$$\begin{aligned} x_1 &\to x_2 \\ x_2 &\to x_1 \\ x_3 &\to x_4 \\ x_4 &\to x_3. \end{aligned} \tag{2.196}$$

Instead, x_2 will not converge to x_3.

Let us now consider the case of interconnected robotic arms. We refer to 2R manipulators, such as the one depicted in Fig. 2.19 below. Such manipulators are typically modelled by resorting to the modelling framework of Euler-Lagrange systems, which is generally given by the following equation:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}) + g(\mathbf{q}) = \mathbf{u}. \tag{2.197}$$

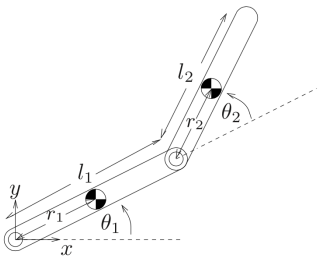


Figure 2.19: Representation of the 2R planar robot manipulator.

This is a nonlinear system, where $M(\mathbf{q})$ represents the inertia matrix and is both symmetric and positive definite, $\mathbf{q}$ represents the generalized coordinates, $C(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}})$ represents the vector of centrifugal torques, $\mathbf{u}$ is the control torque, and $\mathbf{q} = \begin{pmatrix} \theta_1 & \theta_2 \end{pmatrix}^T$ accounts for the joint angles. In particular, $\dot{M}(\mathbf{q}) - 2C(\mathbf{q}, \dot{\mathbf{q}})$ is skew-symmetric.

Recall that if a matrix A is skew-symmetric, then $A^T = -A \implies x^T A x = 0$. This implies that a quadratic form of any x will give us 0. Indeed, $x^T A x = x^T (\frac{A + A^T}{2}) x = 0$ because A is skew-symmetric (i.e., $A + A^T = 0$).

Moreover, $\mathbf{q}^d$ represents the vector of the constant desired generalized coordinates.

The objective is actually to design $\mathbf{u}$ such that $\mathbf{q} \to \mathbf{q}^d$.

We choose to design the following consensus algorithm for multi-robotic arm coordination:

$$\mathbf{u} = g(\mathbf{q}) - K_P(\mathbf{q} - \mathbf{q}^d) - K_D \dot{\mathbf{q}}. \tag{2.198}$$

We are considering a PD controller, where the first term accounts for gravity compensation, the second term is the proportional term and the third term is the derivative term. In particular, we need to know $g(\mathbf{q})$ exactly: this is a drawback.

We are considering a nonlinear system plus a nonlinear controller: so, it is overall nonlinear. Hence, we conduct Lyapunov analysis. We are looking for an equilibrium point in $(\mathbf{q} = \mathbf{q}^d, \dot{\mathbf{q}} = \mathbf{0}, \ddot{\mathbf{q}} = \mathbf{0})$. We want this to be the equilibrium point.

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = -K_P(\mathbf{q} - \mathbf{q}^d) - K_D \dot{\mathbf{q}}. \tag{2.199}$$

Let us now compute the error dynamics. Let the error signal be

$$\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}^d. \tag{2.200}$$

The new system state is therefore $(\tilde{\mathbf{q}}, \dot{\mathbf{q}})$. In this respect, we carry out Lyapunov analysis by means of the following Lyapunov candidate:

$$V = \frac{1}{2}\tilde{\mathbf{q}}^T\tilde{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T\dot{\mathbf{q}}. \tag{2.201}$$

Its derivative along the system trajectories is

$$\dot{V} = \tilde{\mathbf{q}}^T\dot{\tilde{\mathbf{q}}} + \dot{\mathbf{q}}^T\ddot{\mathbf{q}} = \dots \tag{2.202}$$

This Lyapunov candidate does not work if we replace the expression of $\ddot{\mathbf{q}}$ obtained from (2.199) into (2.202).

So, we choose the following Lyapunov candidate, instead,

$$V = \frac{1}{2}\tilde{\mathbf{q}}^T K_P\tilde{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T M(q)\dot{\mathbf{q}}. \tag{2.203}$$

With respect to this Lyapunov candidate, the derivative along the system trajectories yields

$$\dot{V} = \frac{1}{2}\dot{\tilde{\mathbf{q}}}^T K_P\tilde{\mathbf{q}} + \frac{1}{2}\tilde{\mathbf{q}}^T K_P\dot{\tilde{\mathbf{q}}} + \frac{1}{2}\ddot{\mathbf{q}}^T M(\mathbf{q})\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T\dot{M}(\mathbf{q})\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T M(\mathbf{q})\ddot{\mathbf{q}}. \tag{2.204}$$

In particular,

$$\frac{1}{2}\dot{\tilde{\mathbf{q}}}^T K_P\tilde{\mathbf{q}} = \frac{1}{2}\tilde{\mathbf{q}}^T K_P\dot{\tilde{\mathbf{q}}} \tag{2.205}$$

and

$$\frac{1}{2}\ddot{\mathbf{q}}^T M(\mathbf{q})\dot{\mathbf{q}} = \frac{1}{2}\dot{\mathbf{q}}M(\mathbf{q})\ddot{\mathbf{q}}. \tag{2.206}$$

Hence, (2.204) becomes

$$\dot{V} = \tilde{\mathbf{q}}^T K_P\tilde{\mathbf{q}} + \dot{\mathbf{q}}^T M(\mathbf{q})\ddot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T\dot{M}(\mathbf{q})\dot{\mathbf{q}}. \tag{2.207}$$

From (2.199), we know that

$$M(\mathbf{q})\ddot{\mathbf{q}} = -C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - K_P\tilde{\mathbf{q}} - K_D\dot{\mathbf{q}}. \tag{2.208}$$

Hence, thanks to (2.208), we get

$$\begin{aligned}\dot{V} &= \dot{\mathbf{q}}^T K_P\tilde{\mathbf{q}} + \dot{\mathbf{q}}^T(-C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - K_P\tilde{\mathbf{q}} - K_D\dot{\mathbf{q}}) + \frac{1}{2}\dot{\mathbf{q}}^T\dot{M}(\mathbf{q})\dot{\mathbf{q}} = \\ &= \dot{\mathbf{q}}^T K_P\tilde{\mathbf{q}} - \dot{\mathbf{q}}^T C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \dot{\mathbf{q}}^T K_P\tilde{\mathbf{q}} - \dot{\mathbf{q}}^T K_D\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T\dot{M}(\mathbf{q})\dot{\mathbf{q}} = \\ &= -\dot{\mathbf{q}}^T K_D\dot{\mathbf{q}} \leq 0.\end{aligned} \tag{2.209}$$

This results yields that

$$\dot{V} \equiv 0 \implies \dot{\mathbf{q}} = 0 \tag{2.210}$$

where the notation $\equiv 0$ means that $\dot{V}$ equals 0 forever.

From (2.208), imposing $\dot{\mathbf{q}} = 0$ implies that $\ddot{\mathbf{q}} = 0$. So what remains is $\tilde{\mathbf{q}} = 0$, that is, $\mathbf{q} = \mathbf{q}^d$. So we have shown that $(\mathbf{q}_d, 0, 0)$ is a GAS equilibrium point for the considered multi-agent system.

Therefore, with reference to the mathematical model

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) = \mathbf{u}, \tag{2.211}$$

assuming that $\mathbf{q}^d$ represents the constant desired generalized coordinates, under the control law

$$\mathbf{u} = g(\mathbf{q}) - K_P(\mathbf{q} - \mathbf{q}^d) - K_D(\dot{\mathbf{q}}), \tag{2.212}$$

the Lyapunov function (2.201) works. It is a positive definite candidate function because M is positive definite and K_P is positive definite too.

Defining the error signal as

$$\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}^d, \tag{2.213}$$

the derivative of the Lyapunov candidate function along the system trajectories yields

$$\dot{V} = \ldots = -\dot{\mathbf{q}}^T K_D \dot{\mathbf{q}} \leq 0. \tag{2.214}$$

Thus, we achieve tracking because, no matter where we start from, $\mathbf{q} \to \mathbf{q}^d$.

For instance, such a model can represent multiple robotic arms (see Fig. 2.20), i.e.,

$$M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + g_i(q_i) = u_i, \quad i = 1, \ldots, n. \tag{2.215}$$

Let us assume there are no constraints on the $\mathbf{q}$'s. These robots are networked according to the following control law:

$$u_i = g_i(q_i) - \sum_{j \in \mathcal{N}_i} (q_i - q_j) - K_i \dot{q}_i. \tag{2.216}$$

In particular, the aim of such controller design is that we want to synchronize the multiple robot arms.

With respect to such a controller, we can use the Lyapunov function approach. The Lyapunov function is the sum of a bunch of components. In this case, we have a group Lyapunov function, i.e., the sum of everything:

$$V = \frac{1}{4} \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \|q_i - q_j\|^2 + \frac{1}{2} \sum_{i=1}^{n} \dot{q}_i^T M_i(q_i) \dot{q}_i. \tag{2.217}$$

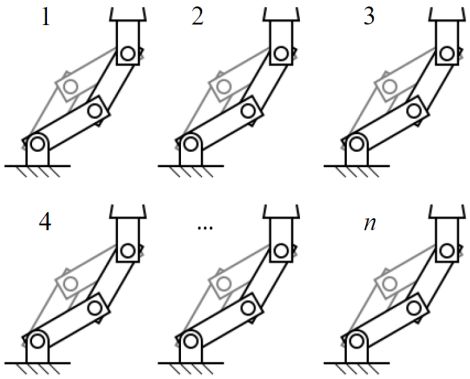


Figure 2.20: Multiple robotic arms.

The norm term in the equation above does not represent the tracking error, but the relative error. This is because

$$\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}\|q_i - q_j\|^2 = 2\mathbf{q}^T(L\otimes I_m)\mathbf{q}. \tag{2.218}$$

We use this form because that is good for showing that such a term is positive definite. Moreover, the reason for that Kronecker product is that $\mathbf{q}_i \in \mathbb{R}^m$.

Previously, we used

$$\mathbf{x}^T L\mathbf{x} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i} a_{ij}(x_i - x_j)^2, \quad x_i \in \mathbb{R}, \mathbf{x}\in\mathbb{R}^n. \tag{2.219}$$

Instead, now we are using

$$\mathbf{x}^T(L\otimes I_m)\mathbf{x} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i} a_{ij}\|x_i - x_j\|^2, \quad x_i \in \mathbb{R}^m, \mathbf{x}\in\mathbb{R}^{m\times n}. \tag{2.220}$$

In vector form, we now have that (2.217) can be rewritten as

$$V = \frac{1}{2}\mathbf{q}^T(L\otimes I_m)\mathbf{q} + \frac{1}{2}\dot{\mathbf{q}}^T M(\mathbf{q})\dot{\mathbf{q}}, \tag{2.221}$$

and this is a more concise version of V. The first term represents some sort of potential energy, whereas the second term represents some sort of kinetic energy. In particular,

$$\dot{\mathbf{q}} = \begin{pmatrix} \dot{q_1} \\ \vdots \\ \dot{q_n} \end{pmatrix}, \tag{2.222}$$

and

$$M(\mathbf{q}) = \begin{pmatrix} M_1(q_1) & \dots & \dots \\ \dots & \ddots & \dots \\ \dots & \dots & M_n(q_n) \end{pmatrix}. \tag{2.223}$$

Let us now compute $\dot{V}$.

$$\dot{V} = \dot{\mathbf{q}}^T(L \otimes I_m)\mathbf{q} + \frac{1}{2}\ddot{\mathbf{q}}M(\mathbf{q})\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T\dot{M}(\mathbf{q})\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T M(\mathbf{q})\ddot{\mathbf{q}}. \tag{2.224}$$

The second term in the equation above is a scalar. The last term is the scalar transpose of the second term, so they must be the same, i.e., (2.224) becomes

$$\dot{V} = \dot{\mathbf{q}}^T(L \otimes I_m)\mathbf{q} + \dot{\mathbf{q}}^T M(\mathbf{q})\ddot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T\dot{M}(\mathbf{q})\dot{\mathbf{q}}. \tag{2.225}$$

From the closed-loop system, we have

$$M(\mathbf{q})\ddot{\mathbf{q}} = -C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - (L \otimes I_m)\mathbf{q} - K\dot{\mathbf{q}}, \tag{2.226}$$

with

$$C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{pmatrix} C_1(q_1, \dot{q}_1) & \dots & \dots \\ \dots & \ddots & \dots \\ \dots & \dots & C_n(q_n, \dot{q}_n) \end{pmatrix}, \tag{2.227}$$

$$K = \begin{pmatrix} K_1 & \dots & \dots \\ \dots & \ddots & \dots \\ \dots & \dots & K_n \end{pmatrix}, \tag{2.228}$$

namely, for each i,

$$M_i(q_i)\ddot{q}_i = -C_i(q_i, \dot{q}_i)\dot{q}_i - \sum_{j \in \mathcal{N}_i}(q_i - q_j) - K_i\dot{q}_i. \tag{2.229}$$

If we put (2.226) into (2.225), we get:

$$\begin{aligned} \dot{V} &= \dot{\mathbf{q}}^T(L \otimes I_m)\mathbf{q} + \dot{\mathbf{q}}^T(-C(\mathbf{q}, \dot{\mathbf{q}}))\dot{\mathbf{q}} - \dot{\mathbf{q}}^T(L \otimes I_m)\mathbf{q} + \\ &+ \frac{1}{2}\dot{\mathbf{q}}^T\dot{M}(\mathbf{q})\dot{\mathbf{q}} - \dot{\mathbf{q}}^T K\dot{\mathbf{q}} = \\ &= -\dot{\mathbf{q}}^T K\dot{\mathbf{q}} \leq 0. \end{aligned} \tag{2.230}$$

The second and fourth term cancel out due to the skew simmetry of $\dot{M} - 2C$. Hence, we have that $\dot{V}$ is negative semi-definite. So, we resort to La Salle's invariance principle.

$$\{\dot{V} = 0\} = \{(\mathbf{q}, \dot{\mathbf{q}}) : \dot{\mathbf{q}} = 0\} \tag{2.231}$$

Therefore, we have that $\dot{q}_i = 0$ in (2.229) implies $\ddot{q}_i = 0$. So, for (2.229) to hold, it is necessary that

$$\sum_{j \in \mathcal{N}_i} (q_i - q_j) = 0 \tag{2.232}$$

If the graph is connected, then we have

$$L\mathbf{x} = 0 \implies \mathbf{x} = c \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \tag{2.233}$$

thus implying synchronization.

If the graph is disconnected, the agents will not synchronize, but just the connected subcomponents of the graph will be synchronized.

In all these cases (nonlinear), the graph is undirected because otherwise the cancellations in the Lyapunov arguments cannot be made anymore.

Instead, let us now use:

$$u_i = g_i(q_i) - \sum_{j \in \mathcal{N}_i} (q_i - q_j) - K \sum_{j \in \mathcal{N}_i} (\dot{q}_i - \dot{q}_i). \tag{2.234}$$

In this case, there is also relative damping: under (2.234), will the robot synchronize and then continue to move on? Unfortunately, (2.234) does not work because for the overall model ($q_i = q_j, \dot{q}_i = \dot{q}_j$) may not be an equilibrium point. But this result can be achieved by following adaptive control ideas.

Let us now consider a second-order model, i.e., the double integrator case,

$$\begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = u_i. \end{cases} \tag{2.235}$$

Let us resort to the control law

$$u_i = -\sum_{j \in \mathcal{N}_i} (x_i - x_j) - K \sum_{j \in \mathcal{N}_i} (v_i - v_j). \tag{2.236}$$

For the single-integrator case, we used Lyapunov analysis to show that

$$\dot{x}_i = u_i \tag{2.237}$$

works with

$$u_i = -\sum_{j \in \mathcal{N}_i} a_{ij} \phi(x_i - x_j), \tag{2.238}$$

where $\phi(\cdot)$ is an odd kind of saturation function.

We now try to copy the previously presented approach and change the linear function with a nonlinear function in the second-order model, for instance a saturation function in order to ensure that the input command is properly bounded.

$$u_i = -\sum_{j\in\mathcal{N}_i} a_{ij}\phi(x_i - x_j) - \sum_{j\in\mathcal{N}_i} a_{ij}\phi(v_i - v_j) \tag{2.239}$$

For single-integrators it is easy. For double integrators, it is more difficult because we are controlling an intermediate state, the acceleration, but we need to get to the velocity. Let us assume again that we have an undirected graph.

Let us choose the following Lyapunov function

$$V = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i} a_{ij}\int_0^{x_i-x_j}\phi(\tau)d\tau + \frac{1}{2}\sum_{i=1}^{n} v_i^2. \tag{2.240}$$

We integrate the function ϕ from 0 to the difference $(x_i - x_j)$. We do not use ϕ^2 because it will not work. Let us see what $\dot{V}$ is:

$$\dot{V} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i} a_{ij}\phi(x_i - x_j)(\dot{x}_i - \dot{x}_j) + \sum_{i=1}^{n} v_i\dot{v}_i, \tag{2.241}$$

where $\dot{x}_i = v_i$ and $\dot{x}_j = v_j$.

$\dot{V}$ is very similar to the controller. For instance, we can select the function ϕ as tanh, i.e.,

$$\int tanh(\cdot) = \log(cosh(\cdot)). \tag{2.242}$$

If we put (2.239) into (2.241), these substitutions cause some cancellations, thus leading to

$$\begin{aligned}\dot{V} &= -\sum_{i=1}^{n} v_i \sum_{j\in\mathcal{N}_i} a_{ij}\phi(v_i - v_j) = \\ &= -\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i} a_{ij}(v_i - v_j)\phi(v_i - v_j) \leq 0\end{aligned} \tag{2.243}$$

In the last equality a nonnegative quantity appears, due to the property that ϕ that is an odd function.

$\dot{V}$ is negative semidefinite. This says that the velocities are to be the same $(v_i = v_j)$. But for this to be true, due to La Salle's invariance principle, the positions have to be the same, too $(x_i = x_j, \forall i, j)$.

2.4.1 Leader Following

In the consensus problem, if we have a leader, we can introduce the leader to specify who is guiding. We can assume to have

- Leader 0 ($\dot{x}_0 = u_0, u_0 = 0$);
- Followers from 1 to n, $\left(\dot{x}_i = u_i, u_i = -\sum_{j\in\mathcal{N}_i} a_{ij}(x_i - x_j)\right)$.

What is the control input for agent 0? Agent 0 has no incoming neighbours, so its input is 0. The leader is indeed a root node in the graph. So, the leader will not move, but the other agents will move towards him.

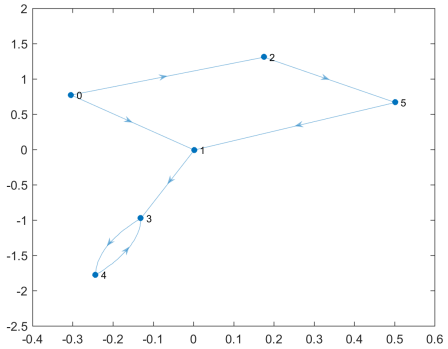


Figure 2.21: An example of leader-following topology.

We are going to use a consensus algorithm except for the fact that there is one node with no neighbours.

$$\begin{aligned} u_1 &= -a_{10}(x_1 - x_0), \\ u_5 &= -a_{52}(x_5 - x_2), \\ &\vdots \end{aligned} \tag{2.244}$$

Whenever there is a leader who receives no information and passes his information to all the other agents, as long as there is a tree, all the other agents will converge to the position of the leader himself, i.e.,

$$x_i \to x_0 \ (constant). \tag{2.245}$$

This is a consensus algorithm and therefore requires x_0 not to move. So, the leader is not moving and the other agents will converge to him.

Yet, there is still a limitation to consider: what if the leader wants to follow a specified trajectory, that is, arbitrary trajectory $x_0(t)$, and we want the others to track it?

$$\begin{cases} \dot{x}_0(t) & = u_0(t) \quad \textit{with} \quad x_0(t) \ \textit{arbitrary}, \\ \dot{x}_i & = u_i. \end{cases} \tag{2.246}$$

How to design u_i in order to track x_0? We could choose:

$$u_i = -K(x_i - x_0). \tag{2.247}$$

Does this controller work for tracking purposes? The closed-loop system will be

$$\dot{x}_i = -K(x_i - x_0), \quad \forall i \in \{1, \ldots, n\}. \tag{2.248}$$

Let the error signal be defined as

$$e = x_i - x_0, \tag{2.249}$$

then the error dynamics is given by

$$\dot{e} = \dot{x}_i - \dot{x}_0 = -Ke - \dot{x}_0. \tag{2.250}$$

Unfortunately, such an approach does not work. So, to solve this, we introduce feedforward

$$u_i = \dot{x}_0 - K(x_i - x_0). \tag{2.251}$$

Hence, the closed-loop system becomes

$$\dot{x}_i = \dot{x}_0 - K(x_i - x_0). \tag{2.252}$$

Given the error signal

$$e = x_i - x_0, \tag{2.253}$$

the error dynamics is

$$\dot{e} = -Ke. \tag{2.254}$$

If we have one single leader followed by one agent, we can use a proportional controller plus a feedforward term. But proportional plus feedforward does not work if we have a bunch of followers (more than one) who want to follow the leader.

In this case, the tracking algorithm for the followers is

$$u_i = -\beta sgn\left(\sum_{j \in \overline{\mathcal{N}}_i} a_{ij}(x_i - x_j)\right), \tag{2.255}$$

where $\overline{\mathcal{N}_i}$ is the neighbour set of node i, including the leader itself. This is a sliding mode controller with

$$sgn(x) = \begin{cases} 1 & x > 0, \\ 0 & x < 0, \\ -1 & x < 0. \end{cases} \tag{2.256}$$

The controller switches back and forth depending on the sign of the error. An important condition for this to work is that the leader passes his information to all the other followers plus the graph for the followers has to be undirected.

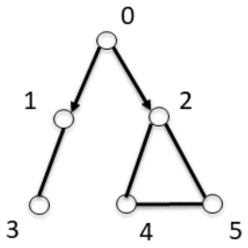


Figure 2.22: A simple example of graph for showing leader following under the sliding mode controller (2.255)

Hence, the algorithm (2.255) is guaranteed to track. Let us set $\tilde{x}_i = x_i - x_0$, accounting for the tracking error for each agent.

$$\dot{\tilde{x}}_i = \dot{x}_i - \dot{x}_0 = -\beta sgn\left(\sum_{j\in\overline{\mathcal{N}}_i}(\tilde{x}_i - \tilde{x}_j) - \dot{x}_0\right). \tag{2.257}$$

Now, let us resort to Lyapunov stability analysis. If we put this in vector for, we are going to have:

$$\dot{\tilde{\mathbf{x}}} = -\beta sgn(M\tilde{\mathbf{x}}) - \mathbf{1}\dot{x}_0, \tag{2.258}$$

with

$$M = L + \begin{pmatrix} a_{10} & \dots & \dots \\ \dots & \ddots & \dots \\ \dots & \dots & a_{n0} \end{pmatrix}, \tag{2.259}$$

This is the tracking error dynamics in vector form.

Namely, with respect to the interconnection of agents shown in Fig. 2.22, we have, for instance,

$$u_2 = -\beta sgn((x_2 - x_0) + (x_2 - x_4) + (x_2 - x_5)), \tag{2.260}$$

where the whole expression between parentheses represents $\overline{\mathcal{N}}_2$. The $(x_2 - x_0)$ term comes from $\begin{bmatrix} a_{10} & \dots & \dots \\ \dots & \ddots & \dots \\ \dots & \dots & a_{n0} \end{bmatrix} \tilde{x}$. The $(x_2 - x_4) + (x_2 - x_5)$ term, instead, comes from the $M\tilde{\mathbf{x}}$ part.

Let us now use a positive definite V. M is a positive definite matrix if and only if the leader has a path to all the other agents. We choose:

$$V = \frac{1}{2}\tilde{\mathbf{x}}^T M \tilde{\mathbf{x}}. \tag{2.261}$$

These are sums of pairwise differences. Its derivative along the system trajectories is

$$\begin{aligned} \dot{V} &= \tilde{\mathbf{x}}^T M \dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{x}} M(-\beta sgn(M\tilde{\mathbf{x}}) - \mathbf{1}\dot{x}) = \\ &= -\beta (M\tilde{\mathbf{x}})^T sgn(M\tilde{\mathbf{x}}) - (M\tilde{\mathbf{x}})^T \mathbf{1}\dot{x}_0 \leq \\ &\leq -\beta \|M\tilde{\mathbf{x}}\|_1 - \|M\tilde{\mathbf{x}}\|_1 \cdot |\dot{x}_0| < 0 \end{aligned} \tag{2.262}$$

We assume that $\beta > |\dot{x}_0|$, so that $|\dot{x}_0|$ is upper bounded. By this assumption, the algorithm works and $\dot{V} < 0$, thus yielding the desired tracking result.

2.5 Networked Lagrangian Systems

Following the observations made in Section 2.3, this Section briefly moves from point models in distributed multi-agent coordination to more realistic Lagrangian models. A class of mechanical systems including autonomous vehicles, robotic manipulators, walking robots, etc. are Lagrangian systems. In particular, we focus on fully-actuated Lagrangian systems. Our objective is to drive a team of agents modeled by Euler-Lagrange equations to achieve the desired relative deviations on thier vectors of generalized coordinates through local interaction.

Let us consider a team of n agents with Euler-Lagrange equations given by

$$M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + g_i(q_i) = \tau_i, \quad i = 1, \dots, n, \tag{2.263}$$

where $q_i \in \mathbb{R}^p$ is the vector of generalized coordinates, $M_i(q_i) \in \mathbb{R}^{p \times p}$ is the symmetric positive-definite inertia matrix, $C_i(q_i, \dot{q}_i)\dot{q}_i \in \mathbb{R}^p$ is the vector accounting for the Coriolis and centrifugal torques, $g_i(q_i)$ is the vector of gravitational torques, and $\tau_i \in \mathbb{R}^p$ represents the vector of torques enforced by the actuators associated with the i-th agent. Throughout the subsequent analysis, we assume that the following assumptions hold.

Assumption 2.1. (*Boundedness*). For any i, there exist positive constants $k_{\underline{m}}, k_{\overline{m}}$,
k_C, k_{C_1}, k_{C_2}, and k_g^* such that $M_i(q_i) - k_{\underline{m}} I_p$ is positive semidefinite, $M_i(q_i) - k_{\overline{m}} I_p$ is negative semidefinite, $\|g_i(q_i)\| \leq k_g$, $\|C_i(x,y)\| \leq k_C\|y\|$, and $\|C_i(x,z)\cdot$

$w - C_i(y,v)w\| \leq k_{C_1}\|z-v\|\|w\| + k_{C_2}\|x-y\| \cdot \|w\| \cdot \|z\|$ for all vectors $x, y, v, w \in \mathbb{R}^p$. □

Assumption 2.2. (*Skew-symmetry property*). $\dot{M}_i(q_i) - 2C_i(q_i, \dot{q}_i)$ is skew-symmetric (i.e., $y^T[\dot{M}_i(q_i) - 2C_i(q_i,\dot{q}_i)]y = 0$ for all $y \in \mathbb{R}^p$). □

Assumption 2.3. (*Linearity in the parameters*). $M_i(q_i)\ddot{q}_i + C_i(q_i,\dot{q}_i)\dot{q}_i + g_i(q_i) = Y_i(q_i,\dot{q}_i,\ddot{q}_i)\Theta_i$, where $Y_i(q_i,\dot{q}_i,\ddot{q}_i)$ is the regressor and Θ_i is the constant parameter vector for the i-th agent. □

Define $q := [q_1^T, \ldots, q_n^T]^T$, and $\dot{q} := [\dot{q}_1^T, \ldots, \dot{q}_n^T]^T$. Also, define $M(q) := diag\ [M_1(q_1), \ldots, M_n(q_n)]$, $C(q,\dot{q}) := diag\ [C_1(q_1,\dot{q}_1), \ldots, C_n(q_n,\dot{q}_n)]$, and $g(q) := [g_1^T(q_1), \ldots, g_n^T(q_n)]^T$.

2.5.1 Distributed Leaderless Coordination

Let us define $\breve{q}_{ij} := \delta_i - \delta_j$, where $\delta_i \in \mathbb{R}^p$ is constant. Here $\breve{q}_{ij}$ denotes the constant desired relative deviation on vectors of generalized coordinates between agents i and j. Our objective here is to design a distributed leaderless coordination algorithm for the networked Lagrangian system (2.263) such that $q_i(t) - q_j(t) \to \breve{q}_{ij}$ and $\dot{q}_i(t) \to \mathbf{0}_p$ as $t \to \infty$. Before presenting such an algorithm, we need the following lemma.

Lemma 2.6. *[154] Let $\psi : \mathbb{R} \to \mathbb{R}$ be a continuous odd function with $\psi(x) > 0$ for $x > 0$. Suppose that $\zeta_i \in \mathbb{R}^p, \phi_i \in \mathbb{R}^p, K \in \mathbb{R}^{p\times p}$, and $D := [d_{ij}] \in \mathbb{R}^{n\times n}$. If D is symmetric, then*

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}(\zeta_i - \zeta_j)^T\psi[K(\phi_i - \phi_j)] = \sum_{i=1}^{n}\zeta_i^T\left\{\sum_{j=1}^{n} d_{ij}\psi[K(\phi_i - \phi_j)]\right\}. \quad (2.264)$$

Let us now consider a coordination algorithm in the form

$$\tau_i = g_i(q_i) - \sum_{j=1}^{n} a_{ij}(q_i - q_j - \breve{q}_{ij}) - \sum_{j=1}^{n} b_{ij}(\dot{q}_i - \dot{q}_j) - K_i\dot{q}_i, \quad (2.265)$$

where $i = 1, \ldots, n$, a_{ij} is the (i,j)-th entry of the adjacency matrix $\mathcal{A} \in \mathbb{R}^{n\times n}$ associated with the undirected graph $\mathcal{G}_A := (\mathcal{V}, \mathcal{E}_A)$ characterizing

the interaction among the n agents for q_i, b_{ij} is the (i,j)-th entry of the adjacency matrix $\mathcal{B} \in \mathbb{R}^{n\times n}$ associated with the undirected graph $\mathcal{G}_B := (\mathcal{V}, \mathcal{E}_B)$ characterizing the interaction among the n agents for $\dot{q}_i$, and $K_i \in \mathbb{R}^{p\times p}$ is a symmetric positive definite matrix. Moreover, $\mathcal{G}_A$ and $\mathcal{G}_B$ are allowed to be different.

Theorem 2.7. *[154] Using control law (2.265) for the networked Lagrangian system (2.263), $q_i(t) - q_j(t) \to \breve{q}_{ij}$ and $\dot{q}_i(t) \to \mathbf{0}_p, i,j = 1,\dots,n$ as $t \to \infty$ if the graph $\mathcal{G}_A$ is undirected connected and the graph $\mathcal{G}_B$ is undirected.*

Proof. Using control law (2.265), the networked Lagrangian system (2.263) can be written as

$$\begin{aligned}
&\frac{d}{dt}(q_i - q_j - \breve{q}_{ij}) = \dot{q}_i - \dot{q}_j,\\
&\frac{d}{dt}\dot{q}_i = -M_i^{-1}(q_i)\Big[C_i(q_i,\dot{q}_i),\dot{q}_i + \sum_{j=1}^{n} a_{ij}(q_i - q_j - \breve{q}_{ij}) + \\
&+ \sum_{j=1}^{n} b_{ij}(\dot{q}_i - \dot{q}_j)K_i\dot{q}_i\Big].
\end{aligned} \tag{2.266}$$

Let $\breve{\mathbf{q}} := [\breve{q}_1^T,\dots,\breve{q}_n^T]^T$ with $\breve{q}_i := q_i - \delta_i$, and $K := diag(K_1,\dots,K_n)$. Let L_A and L_B be, respectively, the Laplacian matrix associated with $\mathcal{G}_A$ and with $\mathcal{G}_B$. Note that both L_A and L_B are symmetric positive semidefinite because both $\mathcal{G}_A$ and $\mathcal{G}_B$ are undirected. Let $\breve{\mathbf{q}}$ be a column stack vector of all $q_i - q_j - \breve{q}_{ij}$, where $i < j$ and $a_{ij} > 0$ (i.e., agents i and j are neighbours). Define $\mathbf{x} := [\breve{\mathbf{q}}^T, \dot{\mathbf{q}}^T]^T$. Let us now consider the Lyapunov function candidate for system (2.266) as

$$V(t,\mathbf{x}) = \frac{1}{2}\breve{\mathbf{q}}^T(L_A \otimes I_p)\breve{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T M(\mathbf{q})\dot{\mathbf{q}}. \tag{2.267}$$

Because the graph $\mathcal{G}_A$ is undirected, then we can write

$$\breve{\mathbf{q}}^T(L_A \otimes I_p)\breve{\mathbf{q}} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\|q_i - q_j - \breve{q}_{ij}\|^2. \tag{2.268}$$

It then follows that V is positive definite and decrescent with respect to $\mathbf{x}$.

The derivative of the Lyapunov candidate function V is given by:

$$\begin{aligned}
\dot{V}(t,\mathbf{x}) &= \dot{\mathbf{q}}^T(L_A \otimes I_p)\breve{\mathbf{q}} + \frac{1}{2}\ddot{\mathbf{q}}^T M(\mathbf{q})\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T\dot{M}(\mathbf{q})\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T M(\mathbf{q})\ddot{\mathbf{q}} =\\
&= \dot{\mathbf{q}}^T(L_A \otimes I_p)\breve{\mathbf{q}} + \dot{\mathbf{q}}^T M(\mathbf{q})\ddot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T\dot{M}(\mathbf{q})\dot{\mathbf{q}},
\end{aligned} \tag{2.269}$$

by using the fact that $M(\mathbf{q})$ is symmetric. Using control law (2.265), system (2.266) can be written in a vector form as

$$M(\mathbf{q})\ddot{\mathbf{q}} = -C(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - (L_A \otimes I_p)\breve{\mathbf{q}} - (L_B \otimes I_p)\dot{\mathbf{q}} - K\dot{\mathbf{q}}. \tag{2.270}$$

Recall that $\dot{M}(\mathbf{q}) - 2C(\mathbf{q},\dot{\mathbf{q}})$ is skew-symmetric. By applying (2.270), the derivative of V can be rewritten as

$$\dot{V}(t,\mathbf{x}) = -\dot{\mathbf{q}}^T(L_B \otimes I_p)\dot{\mathbf{q}} - \dot{\mathbf{q}}^T K\dot{\mathbf{q}} \leq 0. \tag{2.271}$$

Let now $W(t,\mathbf{x}) := \dot{\mathbf{q}}^T(L_A \otimes I_p)\breve{\mathbf{q}}$. Then, $|W(t,x)| \leq \|\dot{\mathbf{q}}\|\|(L_A \otimes I_p)\breve{\mathbf{q}}\|$. Inequality (2.271) implies that $V[t,\mathbf{x}(t)] \leq V[0,\mathbf{x}(0)], \forall t \geq 0$, which in turn implies that $\|\breve{\mathbf{q}}\|$ and $\|\dot{\mathbf{q}}\|$ are bounded. Since $(L_A \otimes I_p)\breve{\mathbf{q}}$ is a column stack vector of all $\sum_{j=1}^{n} a_{ij}(q_i - q_j - \breve{q}_{ij}), i = 1,\ldots,n$, it follows that $\|(L_A \otimes I_p)\breve{\mathbf{q}}\|$ is bounded. It follows that $|W(t,\mathbf{x})|$ is bounded along the solution trajectory.

At this point, the derivative of W along the solution trajectories of (2.270) is

$$\begin{aligned}
\dot{W}(t,\mathbf{x}) &= \ddot{\mathbf{q}}^T(L_A \otimes I_p)\breve{\mathbf{q}} + \dot{\mathbf{q}}^T(L_A \otimes I_p)\dot{\mathbf{q}} = \\
&= -\dot{\mathbf{q}}^T C^T(\mathbf{q},\dot{\mathbf{q}})M^{-1}(\mathbf{q})(L_A \otimes I_p)\breve{\mathbf{q}} - \\
&- \breve{\mathbf{q}}^T(L_A \otimes I_p)M^{-1}(\mathbf{q})(L_A \otimes I_p)\breve{\mathbf{q}} - \\
&- \dot{\mathbf{q}}^T(L_B \otimes I_p)M^{-1}(\mathbf{q})(L_A \otimes I_p)\breve{\mathbf{q}} - \dot{\mathbf{q}}^T K M^{-1}(\mathbf{q})(L_A \otimes I_p)\breve{\mathbf{q}} + \\
&+ \dot{\mathbf{q}}^T(L_A \otimes I_p)\dot{\mathbf{q}}.
\end{aligned} \tag{2.272}$$

Note that $\|\dot{\mathbf{q}}\|$ is bounded. It follows from Assumpt. 2.1 that $\|M^{-1}(\mathbf{q})\|$ and $C(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ are bounded. Therefore, $\dot{W}(t,\mathbf{x})$ can be written as $\dot{W}(t,\mathbf{x}) = g[\beta(t),\mathbf{x}]$, where g is continuous in both arguments and $\beta(t)$ is continuous and bounded. On the set $\Omega := \{(\breve{\mathbf{q}},\dot{\mathbf{q}})|\dot{V} = 0\}, \dot{\mathbf{q}} = \mathbf{0}_{np}$ and $\dot{W}(t,\mathbf{x})$ becomes

$$\dot{W}(t,\mathbf{x}) = -\breve{\mathbf{q}}^T(L_A \otimes I_p)M^{-1}(\mathbf{q})(L_A \otimes I_p)\breve{\mathbf{q}}. \tag{2.273}$$

Note that $M^{-1}(\mathbf{q})$ is symmetric positive definite. It follows from Assumpt. 2.1 that

$$\breve{\mathbf{q}}^T(L_A \otimes I_p)M^{-1}(\mathbf{q})(L_A \otimes I_p) \geq \frac{1}{k_{\overline{m}}}\|(L_A \otimes I_p)\breve{\mathbf{q}}\|^2. \tag{2.274}$$

In particular, note that $\|(L_A \otimes I_p)\breve{\mathbf{q}}\|^2$ is positive definite with respect to $\breve{\mathbf{q}}$. Moreover, on the set Ω, there exists a class $\mathcal{K}$ function, $\alpha(\cdot)$, such that $\|(L_A \otimes I_p)\breve{\mathbf{q}}\|^2 \geq \alpha(\|\breve{\mathbf{q}}\|)$. Therefore, for all $\mathbf{x} \in \Omega, |\dot{W}(t,\mathbf{x})| \geq 1/k_{\overline{m}}\alpha(\|\breve{\mathbf{q}}\|)$. We conclude from this that the equilibrium of (2.266) (i.e., $\|\breve{\mathbf{q}}\| = 0$ and $\|\dot{\mathbf{q}}\| = 0$) is uniformly asymptotically stable, thus implying that $q_i(t) - q_j(t) \to \breve{q}_{ij}$ and $\dot{q}_i(t) \to \mathbf{0}_p$ as $t \to \infty$ because $\mathcal{G}_A$ is undirected connected.

□

2.5.2 Simulation

We now simulate a scenario where six two-link revolute joint arms are coordinated through local interaction according to the algorithm (2.265). For simplicity, we assume each arm to be identical. The Euler-Lagrange equation of each two-link revolute joint arm is given above. In particular, the inertia matrix, the vector of Coriolis and centrifugal torques, and the vector of gravitational torques are given by

$$\begin{aligned} M_i(q_i) &= \begin{bmatrix} \Theta_{i(1)} + 2\Theta_{i(2)}\cos[q_{i(2)}] & \Theta_{i(3)} + \Theta_{i(2)}\cos[q_{i(2)}] \\ \Theta_{i(3)} + \Theta_{i(2)}\cos[q_{i(2)}] & \Theta_{i(3)} \end{bmatrix}, \\ C_i(q_i, \dot{q}_i) &= \begin{bmatrix} -\Theta_{i(2)}\sin[q_{i(2)}]\dot{q}_{i(2)} & -\Theta_{i(2)}\sin[q_{i(2)}][\dot{q}_{i(1)} + \dot{q}_{i(2)}] \\ \Theta_{i(2)}\sin[q_{i(2)}]\dot{q}_{i(1)} & 0 \end{bmatrix}, \\ g_i(q_i) &= \begin{bmatrix} \Theta_{i(4)} g\cos[q_{i(1)}] + \Theta_{i(5)} g\cos[q_{i(1)} + q_{i(2)}] \\ \Theta_{i(5)} g\cos[q_{i(1)} - q_{i(2)}] \end{bmatrix}, \end{aligned} \tag{2.275}$$

where $q_i := [q_{i(1)}, q_{i(2)}]^T$, $g = 9.8\ m/s^2$ is the acceleration due to gravity, $\Theta := [\Theta_{i(1)}, \Theta_{i(2)}, \Theta_{i(3)}, \Theta_{i(4)}, \Theta_{i(5)}] = [m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2) + J_1 + J_2, m_2 l_1 l_{c2}, m_2 l_{c2}^2 + J_2, m_1 l_{c1} + m_2 l_1, m_2 l_{c2}]$.

Here the masses of links 1 and 2 are, respectively, $m_1 = 1\ kg$ and $m_2 = 0.8\ kg$, the lengths of links 1 and 2 are, respectively, $l_1 = 0.8\ m$ and $l_2 = 0.6\ m$, the distances from the previous joint to the center of mass of links 1 and 2 are, respectively, $l_{c1} = 0.4\ m$ and $l_{c2} = 0.3\ m$, and the moments of inertia of links 1 and 2 are, respectively, $J_1 = 0.0533\ kg/m^2$ and $J_2 = 0.024\ kg/m^2$.

For simplicity, we assume that the graphs $\mathcal{G}_A$ and $\mathcal{G}_B$ are identical. Fig. 2.23 shows $\mathcal{G}_A$ (equivalently, $\mathcal{G}_B$) for the six two-link revolute joint arms. The control parameters for algorithm (2.265) are $K_i = I_2,\ i = 1, \ldots, 6, a_{ij} = b_{ij} = 1$ if $(i, j) \in \mathcal{E}_A$ (or $\mathcal{E}_B$), $\breve{q}_{ij} = \mathbf{0}_2$. We let $q_i(0) = [\frac{\pi}{7}i, \frac{\pi}{8}i]^T\ rad$ and $\dot{\mathbf{q}}(0) = [0.1i - 0.4, -0.1i + 0.5]^T rad/s$, where $i = 1, \ldots, 6$.

Figs. 2.25-2.27 show, respectively, the joint angles, their derivatives, and the control torques of the six robot arms using control law (2.265). The joint angles of all arms achieve coordination while their derivatives converge to zero.

2.6 Sampled-Data Control

We will now briefly investigate distributed multi-agent coordination in a sampled-data setting. In particular, we will examine a distributed sampled-data algorithm for coordinated tracking where a group of followers with single-integrator dynamics, by interaction with their neighbours at discrete-time instants, intercepts a dynamic leader who is a neighbour of only a

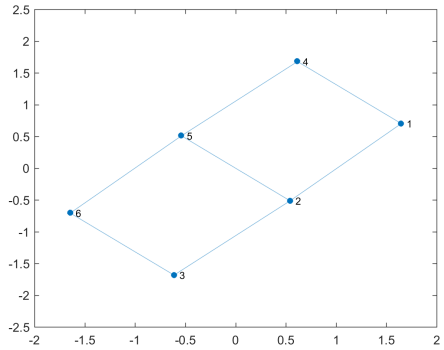


Figure 2.23: Graph $\mathcal{G}_A$ (equivalently, $\mathcal{G}_B$) for six two-link revolute joint arms. An edge between i and j denotes that agents i and j are neighbours. The graph is undirected connected.

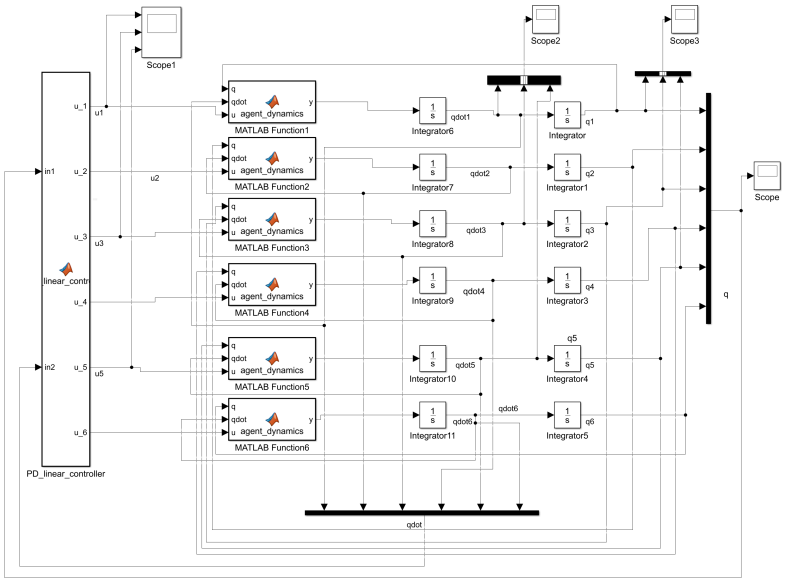


Figure 2.24: Simulink block scheme of control law (2.265) plugged into the networked Lagrangian system described in the current Subsection.

subset of the followers. In this respect, a PD-like discrete-time algorithm is proposed and we consequently study the conditions that must hold on the interaction graph, on the sampling period, and on the control gain in order to guarantee stability under directed fixed interaction, remaining within quantitative bounds for the tracking errors.

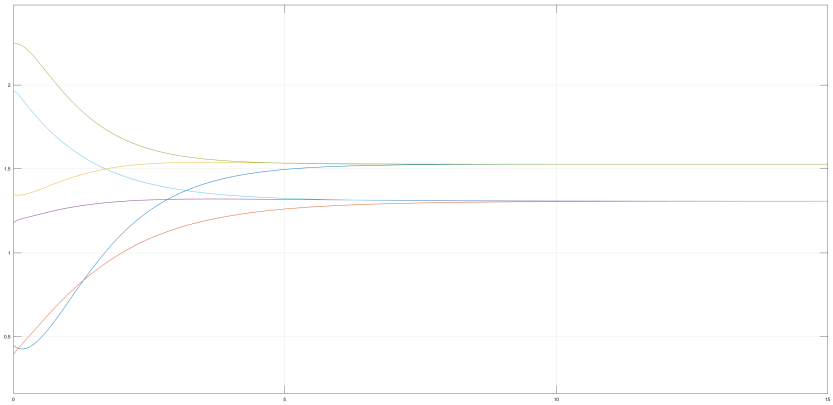


Figure 2.25: Joint angles of the six robot arms using control law (2.265).

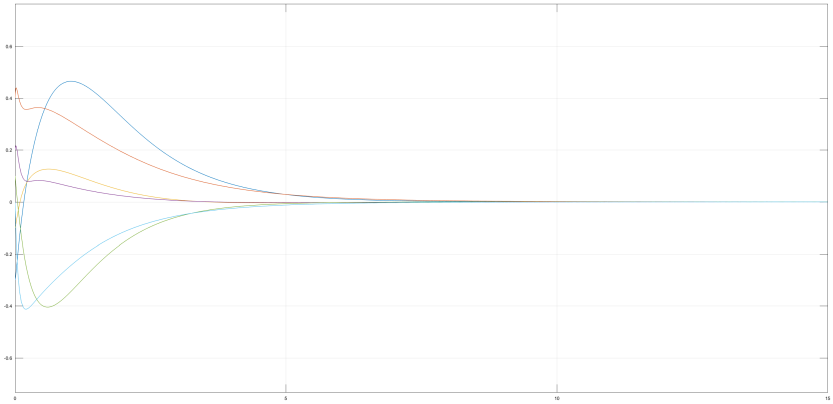


Figure 2.26: Joint angle derivatives of the six robot arms using control law (2.265).

2.6.1 Sampled-Data Coordinated Tracking for Single-Integrator Dynamics

In multi-agent coordination, it often happens that the agents are able to interact with the neighbouring agents only *intermittently* rather than *continuously*, e.g., due to low bandwidth, unreliable communication channels, limited sensing capabilities, or power and cost constraints.

In particular, a multi-agent system with intermittent interaction, where agents with continuous-time dynamics are controlled on the basis of information coming from their neighbours and updated at discrete-time instants, can be treated as a sampled-data system consisting of multiple networked sub-

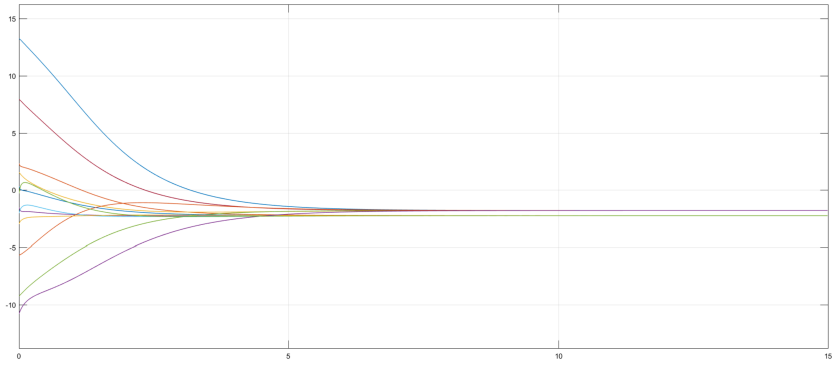


Figure 2.27: Control torques of the six robot arms using control law (2.265).

systems. This is why we are motivated to study also distributed multi-agent coordination in a sampled-data setting, and we explicitly consider the effect of sampled-data control on the stability of the agents. Hence, we will now focus on sampled-data coordinated tracking for single-integrator dynamics.

Let us assume that, in addition to n followers, labeled as agents or followers from 1 to n, with single-integrator dynamics, there exists a dynamic leader, defined as agent 0, whose position is denoted with $x_0(t) \in \mathbb{R}^m$. The leader is assumed to be either physical or virtual. Let $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ be the directed graph characterizing the interaction among the n followers. Let $\overline{\mathcal{G}} := (\overline{\mathcal{G}}, \overline{\mathcal{E}})$ be the directed graph accounting for the interaction among the leader and the followers corresponding to $\mathcal{G}$.

Let us briefly recall a Proportional-Derivative-like (PD-like) continuous-time coordinated tracking algorithm proposed in [151] as

$$\begin{aligned} u_i(t) = & \frac{1}{\sum_{j=0}^{n} a_{ij}} \sum_{j=1}^{n} a_{ij}\{\dot{x}_j(t) - \gamma[x_i(t) - x_j(t)]\} + \\ & + \frac{a_{i0}}{\sum_{j=0}^{n} a_{ij}}\{\dot{x}_0(t) - \gamma[x_i(t) - x_0(t)]\}, \quad \forall v_i \in \mathcal{V}, \end{aligned} \tag{2.276}$$

where $a_{ij},\ i, j = 1, \ldots, n$ is the (i, j)-th entry of the adjacency matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$ associated with the directed graph $\mathcal{G}$, $a_{i0} > 0, i = 1, \ldots, n$, if the leader is a neighbour of follower i and $a_{i0} = 0$ otherwise, and γ is a positive gain. The objective of control law (2.276) is to guarantee that $x_i(t) - x_0(t) \to \mathbf{0}_m,\ i = 1, \ldots, n$, as $t \to \infty$. Note that (2.276) requires that each follower obtains instantaneous measurements of the neighbours' velocities and also the leader's velocity if the leader is a neighbour of the follower. This might not be a realistic requirement in practical applications. So, we propose a discrete-time coordinated tracking algorithm.

Hence, we consider a sampled-data setting where the agents have continuous-time dynamics while the measurements are made at discrete sampling times and the control inputs are enforced via zero-order hold as

$$u_i(t) = u_i[k], \quad kT \leq t < (k+1)T, \quad i = 1, \ldots, n, \tag{2.277}$$

where k denotes the discrete-time index, T denotes the sampling period, and $u_i[k]$ is the control input at $t = kT$. By using direct discretization, the continuous-time system can be discretized as

$$x_i[k+1] = x_i[k] + Tu_i[k], \quad i = 1, \ldots, n, \tag{2.278}$$

where $x_i[k]$ is the position of follower i at $t = kT$. We propose a PD-like discrete-time coordinated tracking algorithm as

$$\begin{aligned} u_i[k] = &\frac{1}{\sum_{j=0}^{n} a_{ij}} \left[\frac{x_j[k] - x_j[k-1]}{T} - \gamma(x_i[k] - x_j[k]) \right] + \\ &+ \frac{a_{i0}}{\sum_{j=0}^{n} a_{ij}} \left[\frac{x_0[k] - x_0[k-1]}{T} - \gamma(x_i[k] - x_0[k]) \right], \end{aligned} \tag{2.279}$$

where $x_0[k]$ denotes the leader's position at time $t = kT$, and $\frac{x_j[k]-x_j[k-1]}{T}$ and $\frac{x_0[k]-x_0[k-1]}{T}$ are used to approximate (namely, via the backward Euler method), respectively, $\dot{x}_j(t)$ and $\dot{x}_0(t)$ by noting that $x_j[k+1]$ and $x_0[k+1]$ cannot be accessed at $t = kT$. Note that using (2.279), each follower's position is updated based on its current position and its neighbours' current and previous positions as well as the leader's current and previous positions if the leader is a neighbour of the considered follower. The PD-like discrete-time control law (2.279) can easily be implemented in practice. In the following, we assume that all agents are in a one-dimensional space (i.e., $m = 1$) for the simplicity of presentation. However, all results are still valid for any high-dimensional space if we introduce the Kronecker product.

Let us analyze the tracking error dynamics as a result of the enforcement of the coordinated tracking algorithm (2.279). Let the tracking error for follower i be denoted with $\varepsilon_i[k] := x_i[k] - x_0[k]$. It follows that the error dynamics for the closed-loop system using (2.279) can be written as

$$\begin{aligned} \varepsilon_i[k+1] = \varepsilon_i[k] &+ \frac{T}{\sum_{j=0}^{n} a_{ij}} \sum_{j=1}^{n} a_{ij} \left[\frac{\varepsilon_j[k] - \varepsilon_j[k-1]}{T} - \gamma(\varepsilon_i[k] - \varepsilon_j[k]) \right] + \\ &+ \frac{Ta_{i0}}{\sum_{j=0}^{n} a_{ij}} \left(\frac{x_0[k] - x_0[k-1]}{T} - \gamma\varepsilon_i[k] \right) - \\ &- (x_0[k+1] - x_0[k]) \frac{\sum_{j=1}^{n} a_{ij}}{\sum_{j=0}^{n} a_{ij}} (x_0[k] - x_0[k-1]), \end{aligned} \tag{2.280}$$

which can then be written in vector form as

$$\varepsilon[k+1] = [(1-T\gamma)I_n + (1+T\gamma)D^{-1}\mathcal{A}]\varepsilon[k] - D^{-1}\mathcal{A}\varepsilon[k-1] + X^r[k], \quad (2.281)$$

where $D := diag\{\sum_{j=0}^{n} a_{1j}, \ldots, \sum_{j=0}^{n} a_{nj}\}, \varepsilon[k] := [\varepsilon_1[k], \ldots, \varepsilon_n[k]]^T, \mathcal{A}$ is the adjacency matrix associated with $\mathcal{G}$, and $X^r[k] := (2x_0[k]x_0[k-1] - x_0[k+1])\mathbf{1}_n$. Setting $Y[k+1] := \begin{bmatrix} \varepsilon[k+1] \\ \varepsilon[k] \end{bmatrix}$, it follows from (2.281) that

$$Y[k+1] = \tilde{A}Y[k] + \tilde{B}X^r[k], \quad (2.282)$$

with

$$\tilde{A} := \begin{bmatrix} (1-T\gamma)I_n + (1+T\gamma)D^{-1}\mathcal{A} & -D^{-1}\mathcal{A} \\ I_n & 0_{n\times n} \end{bmatrix}, \quad (2.283)$$

and $\tilde{B} := \begin{bmatrix} I_n \\ 0_{n\times n} \end{bmatrix}$.The solution of (2.282) is then

$$Y[k] = \tilde{A}^k Y[0] + \sum_{i=1}^{k} \tilde{A}^{k-i}\tilde{B}X^r[i-1]. \quad (2.284)$$

In particular, note that the eigenvalues of $\tilde{A}$ play an important role in determining the value of $Y[k]$ as $k \to \infty$. We therefore study the eigenvalues of $\tilde{A}$. Before moving on, first we study the eigenvalues of $D^{-1}\mathcal{A}$.

Lemma 2.8. *[155] Suppose that in $\overline{\mathcal{G}}$ the leader has directed paths to all followers from 1 to n. Then, $D^{-1}\mathcal{A}$ satisfies $\|(D^{-1}\mathcal{A})^n\|_\infty < 1$ and $D^{-1}\mathcal{A}$ has all eigenvalues within the unit circle.*

Proof. Note that in $\overline{\mathcal{G}}$, if the leader has directed paths to all followers, then each follower has at least one neighbour, that is, $\sum_{j=0}^{n} a_{ij} > 0, i = 1, \ldots, n$. Hence, D^{-1} exists and (2.279) is well defined. Note also that $D^{-1}\, mathcal A$ is nonnegative and each row sum of $D^{-1}\mathcal{A}$ is less than or equal to one. Therefore, it follows that $\|D^{-1}\mathcal{A}\|_\infty \leq 1$. Denote $\bar{i}_1$ as the set of followers that are the children of the leader, and $\bar{i}_j,\ j = 2, \ldots, \kappa$, as the set of followers that are the children of followers in $\bar{i}_{j-1}$ but are not in $\bar{i}_r,\ r = 1, \ldots, j-2$. Because the leader has directed paths to all followers from 1 to n, there are at most n edges from the leader to all followers from 1 to n, which implies that $\kappa \leq n$. Let w_i and y_i^T denote, respectively, the i-th column and row of $D^{-1}\mathcal{A}$. When the leader has directed paths to all followers from 1 to n, assume that the k-th follower is a child of the leader, i.e., $a_{k0} > 0$. It follows that $y_k^T\mathbf{1}_n = 1 - \frac{a_{k0}}{\sum_{j=0}^{n} a_{kj}} < 1$. The same property also applies to the other

elements in the set $\bar{i}_1$. Similarly, assume that the l-th follower (one follower in the set $\bar{i}_2$) is a child of the k-th follower (one follower in the set $\bar{i}_1$), which implies that $a_{lk} > 0$. It follows that the sum of the l-th row of $(D^{-1}\mathcal{A})^2$ can be written as $y_l^T \sum_{i=1}^n w_i \leq q_l^T \mathbf{1}_n = 1 - \frac{a_{lk}}{\sum_{j=0}^n a_{kj}} < 1$. In addition, the sum of the k-th row of $(D^{-1}\mathcal{A})^2$ is less than one. A similar analysis shows that each row sum of $(D^{-1}\mathcal{A})^\kappa)$ is less than one when the leader has directed paths to all followers from 1 to n. That is, $\|(D^{-1}A)^\kappa\|_\infty < 1$. Because $\kappa \leq n$ and $\|D^{-1}\mathcal{A}\|_\infty \leq 1$, the condition that $\|(D^{-1}\mathcal{A})^n\|_\infty < 1$ holds. For the second statement, note that $\rho[(D^{-1}\mathcal{A})^n] \leq \|(D^{-1}\mathcal{A})^n\|_\infty$. Because $\|(D^{-1}\mathcal{A})^n\|_\infty < 1$, then $\rho[(D^{-1}\mathcal{A})^n] < 1$, which implies that $\rho(D^{-1}\mathcal{A}) < 1$. □

We now study the conditions under which all the eigenvalues of $\tilde{A}$ stay inside the unit circle.

Lemma 2.9. *[155] Suppose that in $\overline{\mathcal{G}}$ the leader has directed paths to all followers from 1 to n. Let λ_i be the i-th eigenvalue of $D^{-1}\mathcal{A}$. Then, $\tau_i > 0$ holds, where $\tau_i = \frac{2|1-\lambda_i|^2(2[1-Re(\lambda_i)]-|1-\lambda_i|^2)}{|1-\lambda_i|^4+4[Im(\lambda_i)]^2}$. If the positive scalars T and γ satisfy*

$$T\gamma < \min\{1, \min_{i=1,\ldots,n} \tau_i\}, \tag{2.285}$$

then $\tilde{A}$ is such that all its eigenvalues stay within the unit circle.

Proof. When the leader has directed paths to all followers from 1 to n, the previous lemma implies that $|\lambda_i| < 1$. It follows that $|1 - \lambda_i|^2 > 0$ and $|1 - \lambda_1|^2 = 1 - 2Re(\lambda_i) + [Re(\lambda_i)]^2 + [Im(\lambda_i)]^2 < 2[1 - Re(\lambda_i)]$, thus implying that $\tau_i > 0$. Note that the characteristic polynomial of $\tilde{A}$ is given by

$$\begin{aligned}
&det(zI_{2n} - \tilde{A}) = \\
&= det\left(\begin{bmatrix} zI_n - [(1-T\gamma)I_n + (1+T\gamma)D^{-1}\mathcal{A}] & D^{-1}\mathcal{A} \\ -I_n & zI_n \end{bmatrix}\right) = \\
&= det([zI_n - (1-T\gamma)I_n - (1+T\gamma)D^{-1}\mathcal{A}]zI_n + D^{-1}\mathcal{A}) = \\
&= det([z^2 + (T\gamma - 1)z]I_n + [1 - (1+T\gamma)z]D^{-1}\mathcal{A}).
\end{aligned} \tag{2.286}$$

Since λ_i is the i-th eigenvalue of $D^{-1}\mathcal{A}$, we can get that $det(zI_n + D^{-1}\mathcal{A}) = \prod_{i=1}^n (z + \lambda_i)$. It follows that $det(zI_{2n} - \tilde{A}) = \prod_{i=1}^n \{z^2 + (T\gamma - 1)z + [1 - (1 + T\gamma)z]\lambda_i\}$. Hence, the roots of $det(zI_{2n} - \tilde{A}) = 0$ satisfy the condition

$$z^2 + [T\gamma - 1 - (1 + T\gamma)\lambda_i]z + \lambda_i = 0. \tag{2.287}$$

At this point, we recall that each eigenvalue of $D^{-1}\mathcal{A}$, λ_i, corresponds to two eigenvalues of $\tilde{A}$. Instead of computing the roots of (2.287) directly, we apply the bilinear transformation $z = \frac{s+1}{s-1}$ to (2.287) to get

$$T\gamma(1 - \lambda_i)s^2 + 2(1 - \lambda_i)s + (2 + T\gamma)\lambda_i + 2 - T\gamma = 0. \tag{2.288}$$

Since the bilinear transformation is an exact point-to-point mapping from the interior of the unit circle in the complex z-plane to the open left hand-side of the complex s-plane, it follows that (2.287) has all roots within the unit circle if and only if (2.288) has all roots in the open left hand-side of the complex plane.

In the following, we determine the conditions on T and γ under which (2.288) has all roots in the open left hand-side of the complex plane. Letting s_1 and s_2 denote the roots of (2.288), it follows from (2.288) that

$$s_1 + s_2 = -\frac{2}{T\gamma}, \tag{2.289}$$

and

$$s_1 s_2 = \frac{(2+T\gamma)\lambda_i + 2 - T\gamma}{T\gamma(1-\lambda_i)}. \tag{2.290}$$

Noting that (2.289) implies that $Im(s_1)+Im(s_2) = 0$, we define $s_1 = a_1+\mathbf{j}b$ and $s_2 = a_2-\mathbf{j}b$, with j accounting for the imaginary unit. In particular, s_1 and s_2 have negative real parts if and only if $a_1 a_2 > 0$ and $a_1 + a_2 < 0$. Moreover, (2.289) implies $a_1 + a_2 = -\frac{2}{T\gamma} < 0$ because $T\gamma > 0$. We now show a sufficient condition on T and γ such that $a_1 a_2 > 0$ holds. By substituting the definitions of s_1 and s_2 into (2.290), we have $a_1 a_2 + b^2 + \mathbf{j}(a_2 - a_1)b = \frac{(2+T\gamma)\lambda_i+2-T\gamma}{T\gamma(1-\lambda_i)}$, implying that

$$a_1 a_2 + b^2 = -\frac{2+T\gamma}{T\gamma} + \frac{4[1-Re(\lambda_i)]}{T\gamma|1-\lambda_i|^2}, \tag{2.291}$$

$$(a_2 - a_1)b = \frac{4Im(\lambda_i)}{T\gamma|1-\lambda_i|^2}. \tag{2.292}$$

It follows from (2.292) that $b = \frac{4Im(\lambda_i)}{T\gamma(a_2-a_1)|1-\lambda_i|^2}$. Taking into account that $(a_2 - a_1)^2 = (a_1 + a_2)^2 - 4a_1 a_2 = \frac{4}{T^2\gamma^2} - 4a_1 a_2$, After some manipulation, (2.291) can be written as

$$K_1(a_1 a_2)^2 + K_2 a_1 a_2 + K_3 = 0, \tag{2.293}$$

with $K_1 := T^2\gamma^2|1-\lambda_i|^4$, $K_2 := -|1-\lambda_i|^4 + (2+T\gamma)T\gamma|1-\lambda_i|^4 - 4[1 - Re(\lambda_i)]T\gamma|1-\lambda_i|^2$, and $K_3 := \frac{1}{T\gamma}\{4[1-Re(\lambda_i)]|1-\lambda_i|^2 - (2+T\gamma)|1-\lambda_i|^4\} - 4[IM(\lambda_i)]^2$.

It can be computed that $K_2^2 - 4K_1K_3 = \{|1-\lambda_i|^4 + (2+T\gamma)T\gamma|1-\lambda_i|^4 - 4[1 - Re(\lambda_i)T\gamma|1-\lambda_i|^2\}^2 + 16T^2\gamma^2|1-\lambda_i|^4[Im(\lambda_i)]^2 \geq 0$, which implies that (2.293) has two real roots. Because $|\lambda_i| < 1$, it is straightforward to show that $K_1 > 0$. Therefore, a sufficient condition for $a_1 a_2 > 0$ is that $K_2 < 0$ and $K_3 > 0$. When $0 < T\gamma < 1$, since $|1-\lambda_i|^2 < 2[1 - Re(\lambda_i)]$, it follows that $K_2 < -|1-\lambda_i|^4+(2+T\gamma)T\gamma|1-\lambda_i|^4-2T\gamma|1-\lambda_i|^4 = |1-\lambda_i|^4[-1+(T\gamma)^2] \leq 0$.

Similarly, when $0 < T\gamma < \tau_i$, it follows that $K_3 > 0$. Hence, if the positive scalars γ and T satisfy (2.285), all the eigenvalues of $\tilde{A}$ stay inside the unit circle. □

We now apply Lem. 2.9 in order to derive the main result of this subsection.

Theorem 2.10. *[155] Assuming that the leader's position $x_0[k]$ satisfies $\left|\frac{x_0[k]-x_0[k-1]}{T}\right| \leq \overline{x}$ (i.e., the changing rate of $x_0[k]$ is bounded), and in $\overline{G}$ the leader has directed paths to all followers from 1 to n, when the positive scalars γ and T satisfy (2.285), using control law (2.279) for system (2.278), the maximum tracking error of the n followers is ultimately bounded by $2T\overline{x}\|(I_{2n} - \tilde{A})^{-1}\|_\infty$.*

Proof. It follows from (2.279) that

$$\begin{aligned} \|Y[k]\|_\infty &\leq \|\tilde{A}^k Y[0]\|_\infty + \left\|\sum_{i=1}^{k} \tilde{A}^{k-i}\tilde{B}X^r[i-1]\right\|_\infty \leq \\ &\leq \|\tilde{A}^k\|_\infty \|Y[0]\|_\infty + 2T\overline{x}\left\|\sum_{i=0}^{k-1}\tilde{A}^i\right\|_\infty \|\tilde{B}\|_\infty, \end{aligned} \tag{2.294}$$

where we have used the fact that

$$\|X^r[i]\|_\infty = \|(2x_0[i] - x_0[i-1] - x_0[i+1])\mathbf{1}_n\|_\infty \leq 2T\overline{x}, \tag{2.295}$$

for all i because $\left|\frac{x_0[k]-x_0[k-1]}{T}\right| \leq \overline{x}$. When the leader has directed paths to all followers from 1 to n, it follows from Lemma 2.9 that $\tilde{A}$ has all eigenvalues within the unit circle if the positive scalars T and γ satisfy condition (2.285). Hence, $\lim_{k\to\infty}\|\sum_{i=0}^{k-1}\tilde{A}^i\|_\infty = \|(I_{2n} - \tilde{A})^{-1}\|_\infty$. Also note that $\|\tilde{B}\|_\infty = 1$. Therefore, we have that $\|Y[k]\|_\infty$ is ultimately bounded by $2T\overline{x}\|(I_{2n} - \tilde{A})^{-1}\|_\infty$. At this point, the theorem follows by noting that $\|Y[k]\|_\infty$ denotes the maximum tracking error of the n followers. □

Remark 2.2. The ultimate bound of the tracking errors using the PD-like discrete-time coordinated tracking algorithm (2.279) is proportional to the sampling period T. As T approaches zero, the tracking errors will go to zero ultimately when the changing rate of the leader's position is bounded and the leader has directed paths to all followers from 1 to n.

2.6.2 Comparison Between the Proportional-Like and Proportional-Derivative-Like Discrete-Time Coordinated Tracking Algorithms

A proportional-like (P-like) continuous-time coordinated tracking algorithm is given as represented by

$$u_i(t) = -\sum_{j=1}^{n} a_{ij}[x_i(t) - x_j(t)] - a_{i0}[x_i(t) - x_0(t)], \tag{2.296}$$

where $a_{ij}, i = 1, \ldots, n,\ j = 0, \ldots, n$, are defined as in (2.276). Instead, a P-like discrete-time coordinated tracking algorithm for (2.278) is represented by

$$u_i[k] = -\sum_{j=1}^{n} a_{ij}(x_i[k] - x_j[k]) - a_{i0}(x_i[k] - x_0[k]). \tag{2.297}$$

Letting ε_i and ε be defined as above, we write the error dynamics associated with the closed-loop system (2.278) under control law (2.297) as

$$\varepsilon_i[k+1] = \varepsilon_i[k] - T\sum_{j=1}^{n} a_{ij}(\varepsilon_i[k] - \varepsilon_j[k]) - Ta_{i0}\varepsilon_i[k] - (x_0[k+1] - x_0[k]), \tag{2.298}$$

with $i = 1, \ldots, n$. This can be written in vector form as

$$\varepsilon[k+1] = Q\varepsilon[k] - (x_0[k+1] - x_0[k])\mathbf{1}_n, \tag{2.299}$$

where $Q := I_n - TL - Tdiag(a_{10}, \ldots, a_{n0})$ with L being the nonsymmetric Laplacian matrix associated with $\mathcal{G}$. Note that Q is nonnegative when $0 < T < \min_{i=1,\ldots,n} \frac{1}{\sum_{j=0}^{n} a_{ij}}$.

Lemma 2.11. *[155] Suppose that in $\overline{\mathcal{G}}$ the leader has directed paths to all followers from 1 to n. When $0 < T < \min_{i=1,\ldots,n} \frac{1}{\sum_{j=0}^{n} a_{ij}}$, all eigenvalues of Q stay inside the unit circle.*

Theorem 2.12. *[155] Suppose that the leader's position $x_0[k]$ satisfies $\left|\frac{x_0[k]-x_0[k-1]}{T}\right| \leq \overline{x}$, and in $\overline{\mathcal{G}}$ the leader has directed paths to all followers from 1 to n. When $T < \min_{i=1,\ldots,n} \frac{1}{\sum_{j=0}^{n} a_{ij}}$, using control law (2.297) for (2.278), the maximum tracking error of the n followers is ultimately bounded by $\overline{x}\|[L + diag\{a_{10}, \ldots, a_{n0}\}]^{-1}\|_\infty$.*

Proof. The solution of equation (2.299) is

$$\varepsilon[k] = Q^k \varepsilon[0] - \sum_{i=1}^{k} Q^{k-i}(x_0[k] - x_0[k-1])\mathbf{1}_n. \tag{2.300}$$

The proof of Lemma 2.11 and Theorem 2.12 follows from the proof of Theorem 2.10 by taking into account that $\|\varepsilon[k]\|_\infty$ denotes the maximum tracking error of the n followers. □

Remark 2.3. In contrast with the results in Theorem 2.10, the ultimate bound of the tracking errors using the P-like discrete-time coordinated tracking algorithm (2.297) with a dynamic leader is not proportional to the sampling period T. In fact, even when T approaches zero, the tracking errors using (2.297) are not guaranteed to go to zero ultimately. The comparison between Theorems 2.10 and 2.12 shows the benefit of the PD-like discrete-time coordinated tracking algorithm over the P-like discrete-time consensus algorithm when there exists a dynamic leader who is a neighbour of only a subset of the followers. As a special case, when the leader's position is constant (i.e., $\overline{x} = 0$), it follows from Theorems 2.10 and 2.12 that the tracking errors will go to zero ultimately using both the P-like and PD-like discrete-time coordinated tracking algorithms. Yet, in this case, the coordinated tracking problem reduces to a coordinated regulation problem because the leader's position is constant.

2.6.3 Simulation Example

This subsection shows a simulation example of the PD-like discrete-time coordinated tracking algorithm (2.279). To show the benefit of the PD-like discrete-time coordinated tracking algorithm, the related simulation result obtained by applying the P-like discrete-time coordinated tracking algorithm (2.297) is also presented.

We consider a team consisting of four followers and a leader represented by the directed graph $\overline{\mathcal{G}}$ given by Fig. 2.28. The adjacency matrix of such a graph is

$$\mathcal{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}. \tag{2.301}$$

In particular, the graph topology is such that the leader has directed paths to all four followers. We let $a_{ij} = 1$ if agent j is a neighbour of agent i

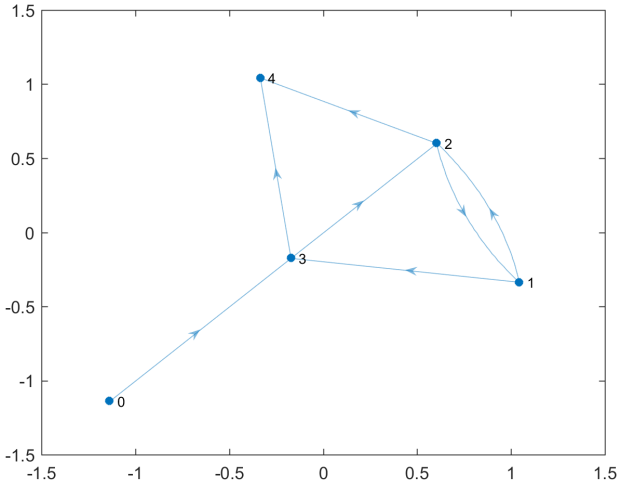


Figure 2.28: Directed graph $\overline{\mathcal{G}}$ associated with four followers and one leader. An arrow from j to i denotes that agent j is a neighbour of agent i.

and $a_{ij} = 0$ otherwise. For both control laws (2.279) and (2.297), we let $x_1[0] = 3, x_2[0] = 1, x_3[0] = -1$, and $x_4[0] = -2$. For control law (2.279), we also let $x_i[-1] = 0, i = 1, \ldots, 4$. The dynamic leader's position is chosen as $x_0[k] = \sin(kT) + kT$.

Figs. 2.29 and 2.30 show the positions x_i and the tracking errors $x_i - x_0$ by using control law (2.279) when $T = 0.3$ s and $\gamma = 1$. In particular, Fig. 2.30 shows that the tracking errors are relatively large. Figs. 2.31 and 2.32 show, instead, x_i and $x_i - x_0$, respectively, by using control law (2.279) when $T = 0.1$ s and $\gamma = 3$. From Fig. 2.32, it can be seen that the tracking errors are very small ultimately. Indeed, the tracking errors become smaller if the amplitude of the sampling period is reduced. Figs. 2.33 and 2.34 show, respectively, x_i and $x_i - x_0$ by using control law (2.279) when $T = 0.25$ s and $\gamma = 3$. Note that, in this case, the product $T\gamma$ is larger than the positive upper bound derived in Theorem 2.10. Therefore, the tracking errors become unbounded. Instead, Figs. 2.35 and 2.36 show, respectively, x_i and $x_i - x_0$ by using (2.297) when $T = 0.1$ s and $\gamma = 3$. By comparing Figs. 2.36 and 2.32, it can be seen that the tracking errors obtained using control law (2.297) are much larger than those achieved using control law (2.297) under the same conditions.

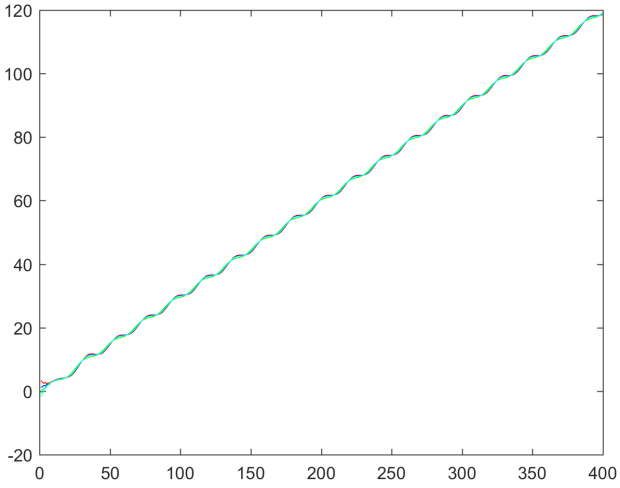


Figure 2.29: Agent positions as a result of distributed discrete-time coordinated tracking using the PD-like discrete-time algorithm (2.279) with $T = 0.3\ s$ and $\gamma = 1$.

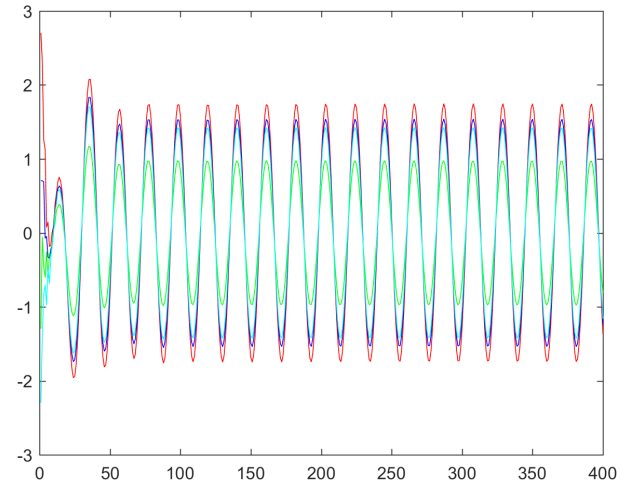


Figure 2.30: Agent tracking errors as a result of distributed discrete-time coordinated tracking using the PD-like discrete-time algorithm (2.279) with $T = 0.3\ s$ and $\gamma = 1$.

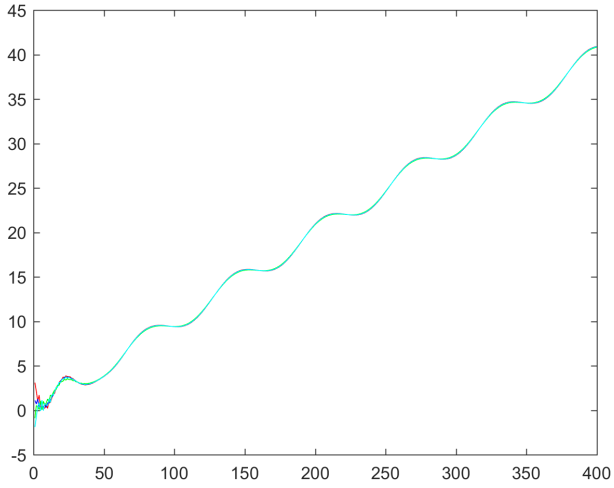


Figure 2.31: Agent positions as a result of distributed discrete-time coordinated tracking using the PD-like discrete-time algorithm (2.279) with $T = 0.1\ s$ and $\gamma = 3$.

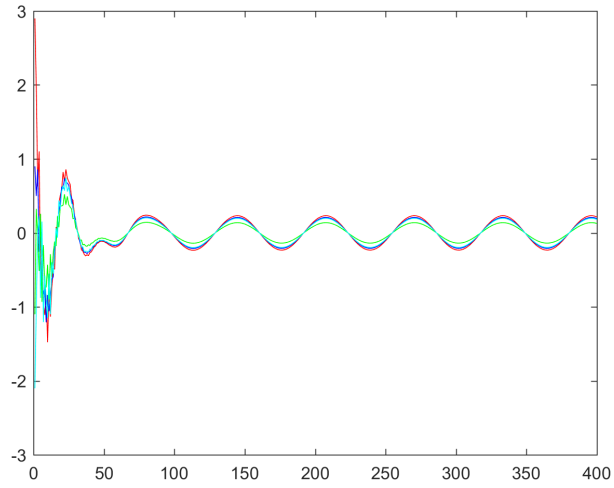


Figure 2.32: Agent tracking errors as a result of distributed discrete-time coordinated tracking using the PD-like discrete-time algorithm (2.279) with $T = 0.1\ s$ and $\gamma = 3$.

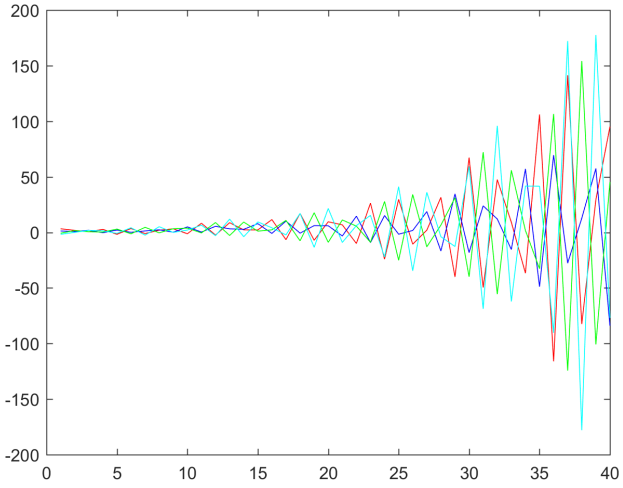


Figure 2.33: Agent positions as a result of distributed discrete-time coordinated tracking using the PD-like discrete-time algorithm (2.279) with $T = 0.25\ s$ and $\gamma = 3$.

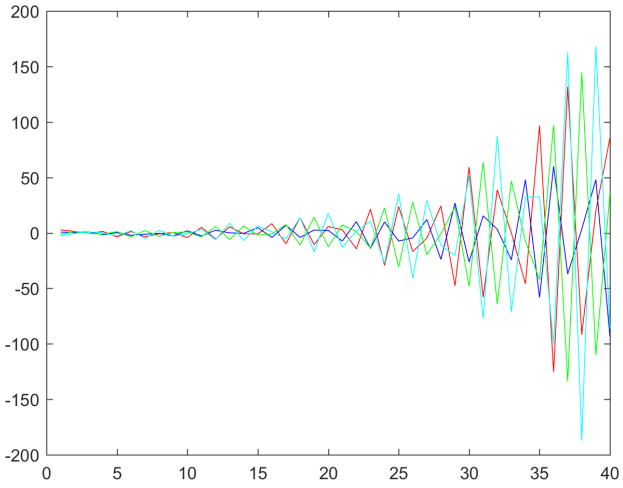


Figure 2.34: Agent tracking errors as a result of distributed discrete-time coordinated tracking using the PD-like discrete-time algorithm (2.279) with $T = 0.25\ s$ and $\gamma = 3$.

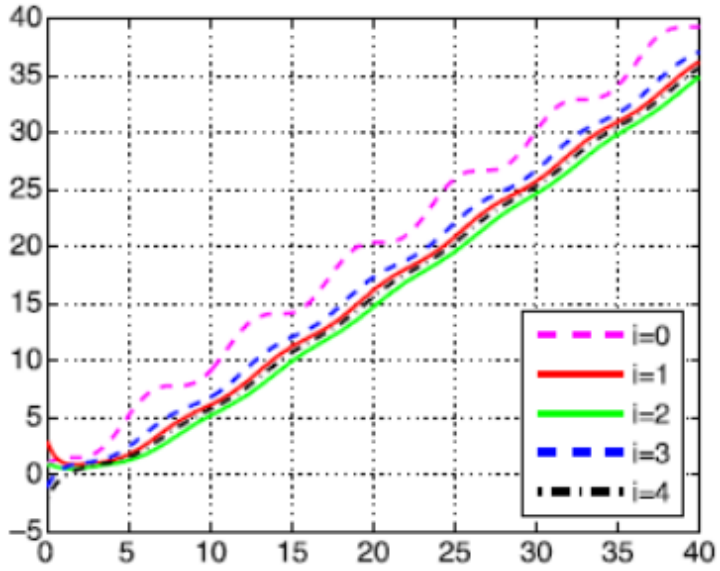


Figure 2.35: Agent positions as a result of distributed discrete-time coordinated tracking using the P-like discrete-time algorithm (2.297 with $T = 0.1\ s$ and $\gamma = 3$ [155].

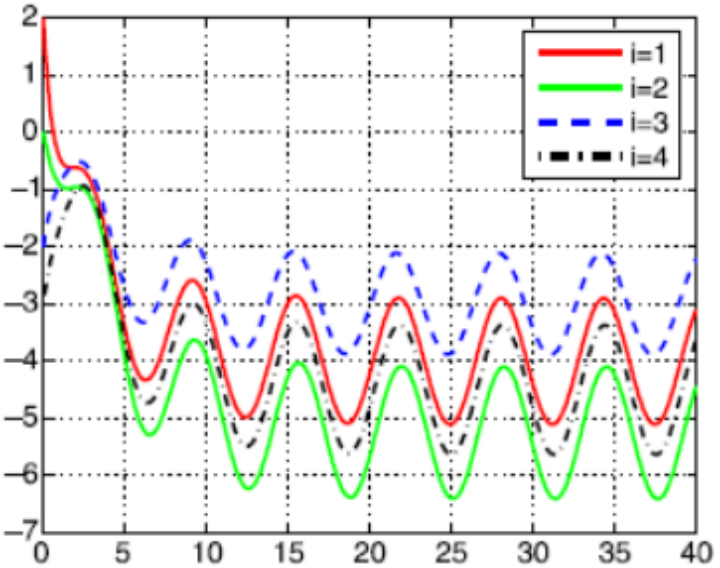


Figure 2.36: Agent tracking errors as a result of distributed discrete-time coordinated tracking using the P-like discrete-time algorithm (2.297 with $T = 0.1\ s$ and $\gamma = 3$ [155].

Chapter 3

Multi-Consensus and Almost Equitable Graph Partitions

3.1 Introduction

In this Chapter, devoted to multi-consensus over static networks, we explore the interdependency between the convergence properties of the considered multi-agent system – defined so that the rate of change of each agent's state is governed by the sum of its relative states with respect to the other neighbouring units – and the structural attributes of the underlying network topology.

In general, consensus in multi-agent systems has attracted a lot of attention due to a wide variety of applications, such as vehicle or robot formation control, swarming, attitude alignment, flocking, rendez-vous problems, and coordinated decision making. Reaching a consensus in a multi-agent system means that the agents' states converge to a common value [150]. After the pioneering works by Vicsek et al. [176] and Jadbabaie et al. [88], a thorough theoretical framework about consensus has been given in [138], [139] and [119]. The consensus problem has also been widely investigated in more complex systems, such as nonlinear [108, 129, 141] or high-order systems [99, 151, 155]. Also, over the last years, there has been an increasing interest in the redesign of the network topology under feedback in order to achieve desired consensus or synchronization objectives [9, 24, 25, 46, 109].

Yet, multi-agent systems may sometimes be required to achieve not just a single global consensus value, but different consensuses in different groups, that is, different parts of a multi-agent system are expected to eventually achieve different consensus states simultaneously. Such a situation is defined as multi-consensus (or cluster/group consensus). In general, multi-consensus

is currently proving to be more momentous than single consensus in control science, physics, brain science, computer science and economics, with a relevant number of applications, such as space-based interferometers, the design of surveillance and reconnaissance systems, and distributed wireless sensor networks, where sensors have to be arranged into different groups, each with a different searching purpose in a different searching area. Hence, it is of great significance to study the behaviour of multiple agents achieving several types of consensus simultaneously. Recently, there have been some interesting investigations in this respect, such as [17], where, in the context of opinion dynamics, the agents are proven to converge to different clusters so that the agents belonging to each cluster share a common opinion. In [92], within each group, the agents cooperate and reach a consensus state. However, in that setting, agents in non-independent subsystems actually do not reach agreement in the common sense. Yu and Wang in [183] further studied the cluster consensus of multi-agent systems in directed networks with time-varying topologies and delays by dividing the network into several corresponding subnetworks. In [110], the cluster consensus problem is investigated in directed networks of nonlinearly coupled multi-agent systems by using pinning control. Other interesting works investigating multi-consensus of continuous-time nonlinearly coupled multi-agent systems are [2], [111] and [105]. Moreover, in [34] discrete-time multi-agent systems with a linear protocol and a time-varying topology are discussed. Yet, in the existing literature about the cluster consensus of multi-agent systems, the clusters of agents are formed artificially and the number of clusters is not determined on the basis of the digraph Laplacian matrix.

In particular, this Chapter is aimed at shedding light on the relationship between the number of formed clusters and the structural properties of the digraph Laplacian matrix. Hence, we tackle the problem of multi-consensus, being inspired by the most recent results on the relationship between a specific kind of graph partitions (namely, the so-called almost equitable ones) and geometric control theory, especially by the invariance properties associated with such graph partitions [29, 133, 164]. Indeed, the almost equitability of a graph partition is an important graph-theoretical property which admits an interesting geometrical interpretation [4, 113, 128, 185] and thus can be profitably used to set and solve networked analysis and control problems relying on a geometric approach. In particular, the mathematical foundation for the results presented in this paper has been laid by the seminal work [32] by Caughman and Veerman, where for the first time a block lower triangular structure is given for the Laplacian matrix of digraphs such that algebraic multiplicity of the zero eigenvalue is larger than one.

The purpose of this Chapter is therefore to analyze in detail the multi-

consensus problem and relate it to the notion of almost equitable graph partition, by providing a clear formalism for examining how the network topology determines some properties of the dynamic evolution and of the steady-state behaviour of a multi-agent system. In this Chapter, we focus on the analysis of multi-consensus behaviours of linearly coupled multi-agent systems over static network topologies only. The relationship between the digraph Laplacian and the number of clusters is highlighted. A necessary and sufficient condition for the global asymptotic stability of multi-consensus in the considered multi-agent system is eventually provided. The topological arrangement proposed highlights that, as in the single-consensus case, also in the multi-consensus one the steady-state values are influenced only by nodes playing the role of digraph roots.

All the contents of this Chapter belong to the paper [127], which is currently under review. The Chapter is organized as follows. Section 3.2 provides background information and notation. Sections 3.3 and 3.4 recall the notions of single consensus, multi-consensus and almost equitable graph partition, thus setting the stage for the subsequent Section 3.5, which shows the main result relating multi-consensus and almost equitable graph partitions, by (i) providing a necessary and sufficient condition for the global asymptotic stability of multi-consensus and (ii) identifying how many different consensuses can be achieved based on the properties of the digraph Laplacian matrix. Concluding remarks in Section 3.6 end the Chapter.

3.2 Notation

Let us consider an *unweighted directed graph* (or digraph) of order n represented by $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{w_1, \ldots, w_n\}$ is a finite nonempty node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is an edge set of ordered pairs of nodes, called edges. For two distinct nodes $w_i, w_j \in \mathcal{V}$, we have $(w_i, w_j) \in \mathcal{E}$ if there is an edge from w_i to w_j with w_i being the *tail* and w_j being the *head* of the edge: hence, w_i is said to be a *neighbour* of w_j. We call any subset ρ of $\mathcal{V}$ a *cell* of $\mathcal{V}$.

A digraph $\mathcal{G}$ contains a *rooted out-branching* as a subgraph if it does not contain a directed cycle and if it has a node w_{root} (i.e., the *root node*) such that for every other node $w \in \mathcal{V}$ there exists a directed path from w_{root} to w.

We call a collection of cells, given by $\pi = \{\rho_1, \rho_2, \ldots, \rho_k\}$, a *partition* of $\mathcal{V}$ if $\rho_i \cap \rho_j = \emptyset$, whenever $i \neq j$, and $\cup_{i=1}^{k} \rho_i = \mathcal{V}$. For a cell $\rho \subseteq \mathcal{V}$, we define the *characteristic vector* of ρ as $p(\rho) \in \mathbb{R}^n$ such that:

$$p_i(\rho) = \begin{cases} 1 & \text{if } \ v_i \in \rho, \\ 0 & \text{otherwise.} \end{cases}$$

For a partition $\pi = \{\rho_1, \rho_2, \ldots, \rho_k\}$, we define the *characteristic matrix* of π as $P(\pi) = \begin{pmatrix} p(\rho_1) & p(\rho_2) & \ldots & p(\rho_k) \end{pmatrix}$. With $Im\ P(\pi)$, we denote the *range space* of $P(\pi)$, that is, the span of the column vectors of $P(\pi)$. A partition π_1 is said to be *finer* than another partition π_2, or alternatively π_2 is *coarser* than π_1, if each cell of π_1 is a subset of some cell of π_2. With the symbol ρ^{tr}, we denote a generic *trivial* cell, i.e., a cell containing one node only.

Moreover, let $\mathbf{1}_n$ represent the n-dimensional vector of all ones and let the vectors $\mathbf{e}_1 = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \end{pmatrix}^T$, $\mathbf{e}_2 = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \end{pmatrix}^T, \ldots,$ $\mathbf{e}_n = \begin{pmatrix} 0 & 0 & \ldots & 0 & 1 \end{pmatrix}^T$ denote the standard basis of $\mathbb{R}^n$. Let $GL_n(\mathbb{R})$ be the general linear group of all $n \times n$ invertible matrices and let $\mathcal{T}_n(\mathbb{R}) \subseteq GL_n(\mathbb{R})$ be the set of the $n!$ *permutation* matrices T_P. Recall that a permutation matrix is a square matrix obtained from the identity matrix of the same size by a permutation of rows and it is always row-equivalent to the identity matrix. Furthermore, every permutation matrix T_P is orthogonal, i.e., $T_P^T = T_P^{-1}$.

Before presenting the main contribution of this paper, some well-known facts are reported for the sake of completeness.

3.3 Recalls on Single Consensus and on Multi-Consensus

For the purposes of multi-agent system analysis and in order to investigate the related consensus properties, the literature typically refers to the *multi-agent system* model

$$\dot{\mathbf{x}} = -L\mathbf{x}, \tag{3.1}$$

consisting of $n > 1$ agents labeled by the node set $\mathcal{V}$, that is, each agent corresponds to a specific node of the underlying digraph $\mathcal{G}$. The state vector is therefore defined as $\mathbf{x}(t) := col(x_1(t), \ldots, x_n(t))$, with $x_i(t)$ denoting the state of node (or agent) w_i at time t. The matrix $L = (l_{ij}) \in \mathbb{R}^{n \times n}$, with $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, n\}$, denotes the *in-degree Laplacian matrix* of the unweighted digraph $\mathcal{G}$.

In particular, we are interested in examining the steady-state behaviour of system (3.1) in the case when the underlying digraph admits a single consensus value or multiple consensuses. A preliminary topological characterization in terms of connectedness is needed in order to specify how different consensus situations are reached.

Definition 3.1. [119] We distinguish among four different topologies in terms of connectedness.

- A digraph is said to be weakly connected if its disoriented version is connected, that is, if its disoriented version is such that there always exists a path between every pair of nodes and there are no unreachable nodes.

- A digraph is said to be a rooted digraph if it is weakly connected and it contains at least one rooted out-branching.

- A digraph is said to be strongly connected if there always exists a directed path between every pair of nodes and there are no unreachable nodes.

- A digraph is said to be balanced if, for every node, the *in-degree* (i.e., the number of head ends adjacent to the considered node) and the *out-degree* (i.e., the number of tail ends adjacent to the considered node) are equal.

Remark 3.1. A strongly connected digraph is both weakly connected and rooted, but the converse is not necessarily true. Moreover, a balanced digraph is strongly connected, but the converse is not necessarily true.

Definition 3.1 will be used to specify how the degree of connectedness of the network topology underlying the dynamics (3.1) impacts on the consensus value that is eventually reached. In this respect, we now give the definition of single consensus and multi-consensus, respectively.

Definition 3.2. Given a digraph $\mathcal{G}$, a single consensus, or simply consensus, is achieved when the differences between the state trajectories of all nodes in $\mathcal{V}$ converge to zero as time goes to infinity, i.e.,

$$\begin{aligned} &\lim_{t\to\infty}[x_i(t) - x_j(t)] = 0, \\ &\forall i, j \mid w_i, w_j \in \mathcal{V}. \end{aligned} \tag{3.2}$$

□

Definition 3.3. Given a digraph partition $\pi = \{\rho_1, \rho_2, \ldots, \rho_k\}$, multi-consensus is defined as the condition when:

- the differences between the state trajectories of nodes belonging to the same cell of π converge to zero as time goes to infinity, i.e.,

$$\begin{aligned} &\lim_{t\to\infty}[x_i(t) - x_j(t)] = 0, \\ &\forall i, j \mid w_i \in \rho_\beta,\ w_j \in \rho_\beta,\ i \neq j,\ \beta = 1, \ldots, k; \end{aligned} \tag{3.3}$$

- the differences between the state trajectories of nodes belonging to different cells of π do not converge to zero, i.e.,

$$\lim_{t\to\infty} |x_{i'}(t) - x_{j'}(t)| > 0 \tag{3.4}$$

holds for each $w_{i'} \in \rho_\beta$ and $w_{j'} \in \rho_{\beta'}$ with $\beta \neq \beta'$.

Multi-consensus is equivalent to the asymptotic stability of the following multi-consensus manifold with respect to partition π:

$$\begin{aligned} \mathcal{M}^\pi = \{x \in \mathbb{R}^n |\ & x_i = x_j,\ \forall i,j\ | \\ & w_i \in \rho_\beta, w_j \in \rho_\beta,\ i \neq j,\ \beta = 1,\dots,k\}. \end{aligned} \tag{3.5}$$

□

As known, consensus or multi-consensus is achieved depending on the algebraic multiplicity, μ, of the zero eigenvalue (e.g., see [182]). First of all, we make a few interesting statements, borrowed from the literature – namely, from [151] and [119] – and reformulated so as to recover the classical consensus and multi-consensus definitions.

Proposition 3.1. *[119] The presence of at least one rooted out-branching in a weakly connected digraph $\mathcal{G}$ is equivalent to the condition that (i) the algebraic multiplicity of the zero eigenvalue of the Laplacian matrix L be unitary, i.e., $\mu(\lambda_1 = 0) = 1$, and (ii) all the other eigenvalues of L have positive real parts.*

Proof. Refer to Proposition 3.8, page 51 in [119]. □

With this in mind, we recall the following result.

Proposition 3.2. *[119] Any rooted digraph underlying the dynamics (3.1) is such that the states of its nodes converge to a unique Globally Asymptotically Stable (GAS) consensus c, i.e.,*

$$\mathbf{x}_\infty = \mathbf{1}_n \cdot c, \quad \textit{with} \quad c = \mathbf{v}_1^T \mathbf{x}_0. \tag{3.6}$$

Proof. The proof, as is well known, immediately follows from the spectral factorization of the exponential of the square matrix $(-L)$, i.e.,

$$\mathbf{x}(t) = e^{-Lt}\mathbf{x}_0 = (\mathbf{u}_1\mathbf{v}_1^T)\mathbf{x}_0 + T\sum_{i=2}^{n}\left(e^{-J(\lambda_i)t}\right)T^{-1}\mathbf{x}_0, \tag{3.7}$$

where the second term comes from the Jordan decomposition of L, i.e., from $L = T^{-1}J(\Lambda)T$ with

$$J(\Lambda) = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & J(\lambda_2) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & J(\lambda_n) \end{pmatrix}. \tag{3.8}$$

Due to Proposition 3.1, the eigenvalues $\lambda_1, \dots, \lambda_n$ of L must be such that

$$\lambda_1 = 0 \text{ and } 0 < Re[\lambda_2] \leq \dots \leq Re[\lambda_n]. \tag{3.9}$$

Hence, considering that $\mathbf{u}_1 = \mathbf{1}_n$ and that the second term in (3.7) vanishes at steady state, we easily recover (3.6). □

Remark 3.2. If the digraph is balanced, L is symmetric and $\mathbf{v} = \mathbf{1}_n$: thus, classical average consensus is recovered, i.e., $c = \sum_{i=1}^{n} \frac{x_{0,i}}{n}$, where $x_{0,i}$ is the initial condition of the state of each digraph node, for $i = 1, \dots, n$.

A deeper understanding of such results can be gained by making use of the properties of the Laplacian matrix L when the digraph is rooted. For that, the following definition is instrumental.

Definition 3.4. Let $\mathcal{I}^*$ denote the maximal invariant subset of $\mathcal{V}$ under the action of L. Formally,

$$\mathcal{I}^* : \{w_i \in \mathcal{V} \mid Lp(\rho^{tr}_{w_i}) = \sum_{j=1}^{l} \alpha_j p(\rho^{tr}_{w_j})\}, \tag{3.10}$$

where $l := |\mathcal{I}^*|$, $\alpha_j \in \mathbb{Z}$, and $p(\rho^{tr}_{w_i})$ denotes the characteristic vector of a trivial cell containing node w_i only.

Remark 3.3. Given $\mathcal{I}^*$ defined as in Definition 3.4, then:

$$\mathcal{I}^* \equiv \mathcal{V} \setminus \mathcal{Q}, \tag{3.11}$$

where $\mathcal{Q}$ denotes the subset of the root nodes of the digraph.

Remark 3.4. The set $\mathcal{I}^*$ is empty if and only if the digraph is strongly connected.

Proposition 3.3. *If $\mathcal{I}^* \equiv \emptyset$, then the permutation matrix*

$$T_P \ : \ T_P^{-1} = \left(\forall \ \middle| \ p(\rho^{tr}_{w_1}) \ \dots \ p(\rho^{tr}_{w_l})\right), \tag{3.12}$$

where the $\forall$ symbol denotes any suitable basis completion consisting of characteristic vectors of trivial cells, yields the following block lower triangular structure:

$$\overline{L} = T_P \; L \; T_P^{-1} = \begin{pmatrix} R & 0 \\ K_1 & K \end{pmatrix}, \tag{3.13}$$

where R and K are square matrices of order $(n-l)$ and l, respectively.

Proof. The proof immediately follows from (3.10) by noting that the characteristic vectors $p(\rho_{w_1}^{tr}), \ldots, p(\rho_{w_l}^{tr})$ are linearly independent. Note that, instead, the first $(n-l)$ columns of T_P^{-1} can be accounted for as the characteristic vectors of as many trivial cells, each containing one of the digraph root nodes. □

Indeed, from the point of view of dynamics, $\mathcal{Q}$ acts as a dominating/independent subset of network nodes, while $\mathcal{I}^*$ acts as the dominated/dependent one.

With this in mind, we can further detail Proposition 3.2 as follows.

Theorem 3.4. *A rooted digraph underlying the dynamics (3.1) is such that the states of its nodes converge to a unique GAS consensus defined as:*

$$\mathbf{x}_\infty = \mathbf{1}_n \cdot c, \quad \textit{with} \quad c = \mathbf{v}_1^T \mathbf{x}_0 = \sum_{i=1}^{r} v_i^r \cdot \overline{x}_{0,i}, \tag{3.14}$$

where $r := |\mathcal{Q}|$, $\mathbf{v}_1$ yields $\mathbf{v}_1^T \mathbf{u}_1 = 1$ (with $\mathbf{u}_1$ and $\mathbf{v}_1$ being the right and left eigenvectors associated with the zero eigenvalue of L, respectively), the v_i^r's are the components of $\mathbf{v}^r$, defined as the vector such that the left eigenvector $\mathbf{v}_1$, in the $\overline{\mathbf{x}} = T_P \; \mathbf{x}$ coordinates, takes the form $\mathbf{v}_1 = \begin{pmatrix} \mathbf{v}^r & | & 0 \end{pmatrix}^T$, and the $\overline{x}_{0,i}$'s are the components of $\overline{\mathbf{x}}_0$, denoting the initial conditions of the system expressed in the $\overline{\mathbf{x}}$-coordinates. Moreover, each v_i^r, $i = 1, \ldots, r$, is such that $v_i^r \neq 0$ and $\sum_{i=1}^r v_i^r = 1$.

Proof. In the $\overline{\mathbf{x}} = T_P \cdot \mathbf{x}$ coordinates, the left eigenvector $\mathbf{v}_1^T$ takes the form $\big((\mathbf{v}^r)^T \; | \; 0\big)$, i.e., it has the last components equal to 0, since R in (3.13) is a Laplacian matrix. This proves that only the initial conditions of the root nodes influence the consensus value that is eventually achieved. Moreover, since $(\mathbf{v}^r)^T$ is a left eigenvector, $\sum_{i=1}^r v_i^r = 1$. It remains to prove that all the v_i^r's are nonzero. This follows from the observation that if one or more elements of $\mathbf{v}^r$ were zero, then the corresponding network nodes would belong to $\mathcal{I}^*$. □

From Theorem 3.4, as the intuition suggests, we infer that only the digraph root nodes influence the achieved consensus value. Equation (3.14) specifies such an influence. As a particular case, note that, if the digraph is strongly connected, all the network nodes are root nodes and must therefore influence the achieved consensus value (i.e., all the coefficients of $\mathbf{v}^T$ must be nonzero).

Instead, if $\mu(\lambda_1) > 1$, then the multi-agent system (3.1) converges to a different GAS equilibrium state than (3.14) or the average consensus. In the following, whenever $\mu(\lambda_1) > 1$, we will refer to such a quantity simply as μ.

Proposition 3.5. *Weak connectedness of the digraph $\mathcal{G}$ and the absence of a rooted out-branching in $\mathcal{G}$ are sufficient conditions for the algebraic multiplicity of the zero eigenvalue to grow above 1, i.e., $\mu > 1$, yielding $rank(L) = n - \mu$. In such a case, the GAS equilibrium state the multi-agent system (3.1) converges to is given by*

$$\begin{aligned} \mathbf{x'}_\infty = \lim_{t\to\infty} \mathbf{x}(t) = (\mathbf{u}_1\mathbf{v}_1)^T\mathbf{x}_0 + (\mathbf{u}_2\mathbf{v}_2)^T\mathbf{x}_0 + \\ + \ldots + (\mathbf{u}_\mu\mathbf{v}_\mu)^T\mathbf{x}_0, \end{aligned} \tag{3.15}$$

where $\mathbf{u}_1, \ldots, \mathbf{u}_\mu$ are μ distinct and linearly independent eigenvectors associated with the zero eigenvalue of L (i.e., $L\mathbf{u}_i = 0,\ i = 1, \ldots, k$) such that $\{\mathbf{u}_1, \ldots, \mathbf{u}_\mu\}$ is a basis of ker $L := \mathcal{U}$.

Proof. It easily follows from the proof of Proposition 3.2, with the only difference that in this case we recover (3.15) since, by assumption, $\lambda_1 = \lambda_2 = \ldots = \lambda_\mu = 0$ and $\lambda_{\mu+1}, \ldots, \lambda_\mu$ are such that their real parts are strictly positive. □

Note that the $\mathbf{u}_i$ eigenvectors, for $i = 1, \ldots, \mu$, are always such that $\sum_{i=1}^{\mu} \mathbf{u}_i = \mathbf{1}_n$. Note also that the GAS equilibrium condition (3.15) does not imply that the same value is reached for all the components of the state vector $\mathbf{x}$, as it occurs, instead, in (3.6) and (3.14). In the following Section, we will therefore investigate the properties the network topology has to fulfil in order to allow the agents belonging to suitable clusters (more precisely, cells) of nodes to converge each to the same final value, thus yielding multi-consensus according to Definition 3.3.

In particular, according to Proposition 3.2 and Definition 3.3, it is possible to numerically compute the distinct consensus values achieved by the agents and, therefore, to identify the different cells $\rho_i, i = 1, \ldots, \omega$, the agents will form at steady state. Such computations could pave the way for the design of a heuristic procedure that is suitable for calculating the almost equitable partition π^*_{AE}, which will be introduced in Section 3.5.

From now till the end of this Section, we will rely on the results appearing in [32] to recall some important properties and introduce our main contribution. In fact, still taking into account the situation when $\mu > 1$, there exists a redenomination of nodes yielding a suitable lower-triangularization of the Laplacian matrix (distinct from the one given by (3.13)), namely allowing to interpret the digraph topology in terms of reaches and thus giving further insight into the kernels of digraph Laplacians.

More in detail, given a Laplacian matrix L associated with the digraph $\mathcal{G}$, we write $w_j \rightsquigarrow w_i$ if there exists a directed path from node w_j to node w_i. In this respect, for any node w_j, we define the reachable set from w_j, $\mathcal{R}(w_j)$, to be the set containing w_j and all nodes w_i such that $w_j \rightsquigarrow w_i$. The maximal reachable set from w_j, $\mathcal{R}_{max}(w_j)$, is called a *reach* [3, 32]. Given the reach $\mathcal{R}_{max}(w_j)$ from a node $w_j \in \mathcal{V}$, then we define w_j as the *reach root node*, that is, the root node for $\mathcal{R}_{max}(w_j)$, since, by definition of reachable set, $w_j \rightsquigarrow w_i,\ \forall w_i \in \mathcal{R}_{max}(w_j)$.

Let $\mathcal{R}_1, \ldots, \mathcal{R}_\mu$ denote the reaches of $\mathcal{G}$. For each reach $\mathcal{R}_i$, we define the *exclusive* part of $\mathcal{R}_i$ to be the set $\mathcal{H}_i = \mathcal{R}_i \setminus \cup_{j \neq i} \mathcal{R}_j$. Likewise, we define the *common* part of $\mathcal{R}_i$ to be the set $\mathcal{C}_i = \mathcal{R}_i \setminus \mathcal{H}_i$. Let $\mathcal{C} = \cup_{i=1}^{\mu} \mathcal{C}_i$ denote the union of the common parts of all the reaches.

Proposition 3.6. *The algebraic multiplicity of $\lambda_1 = 0$ as an eigenvalue of L equals the number, μ, of reaches of $\mathcal{G}$.*

Proof. Refer to Theorem 3.2 in [32]. □

As a result of this, by means of a coordinate change that suitably reorders the digraph nodes, L takes the following lower-triangular form:

$$\overline{L} = \begin{pmatrix} L_1 & 0_{h_1 \times h_2} & \ldots & 0_{h_1 \times h_\mu} & 0_{h_1 \times \delta} \\ 0_{h_2 \times h_1} & L_2 & \ldots & 0_{h_2 \times h_\mu} & 0_{h_2 \times \delta} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_{h_\mu \times h_1} & 0_{h_\mu \times h_2} & \ldots & L_\mu & 0_{h_\mu \times \delta} \\ M_1 & M_2 & \ldots & M_\mu & M \end{pmatrix}, \tag{3.16}$$

where the L_i's are $h_i \times h_i$ Laplacian matrices associated with the $\mathcal{H}_i$'s, the M_i's are $\delta \times h_i$ matrices, and M is a square matrix of order δ associated with the union of the common parts of all the digraph reaches (i.e., with $\mathcal{C}$), with $h_i := |\mathcal{H}_i|$, and $\delta := |\mathcal{C}|$.

The Laplacian matrix $\overline{L}$, as given in (3.16), is the result of a coordinate transformation that makes use of permutation matrices and yields a mere reordering of the network nodes. This is outlined in Proposition 3.7.

Proposition 3.7. *Let perm* : $GL_n(\mathbb{R}) \to GL_n(\mathbb{R})$ *be a bijective function accounting for the permutation transformation, i.e., the transformation rearranging the* $\mathbf{e}_i$ *vectors, for* $i = 1, \ldots, n$*, by column, in the following way:*

$$\begin{pmatrix} \overline{\mathbf{e}}_1 & \ldots & \overline{\mathbf{e}}_n \end{pmatrix} = perm \begin{pmatrix} \mathbf{e}_1 & \ldots & \mathbf{e}_n \end{pmatrix}. \tag{3.17}$$

Then, there always exists a coordinate change

$$\overline{\mathbf{x}} = T_P \, \mathbf{x} \tag{3.18}$$

with T_P^{-1} *chosen as*

$$T_P^{-1} = \begin{pmatrix} \overline{\mathbf{e}}_1 & \ldots & \overline{\mathbf{e}}_n \end{pmatrix}, \tag{3.19}$$

yielding

$$T_P \; L \; T_P^{-1} = \overline{L} \implies \dot{\overline{\mathbf{x}}} = -\overline{L} \, \overline{\mathbf{x}}, \tag{3.20}$$

with $\overline{L}$ *expressed in the lower-triangular form (3.16).*

Example 3.1. Consider a digraph with Laplacian matrix

$$L = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & -1 & 0 & 2 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{pmatrix}, \tag{3.21}$$

which is not in the lower-triangular form (3.16). This Laplacian matrix is such that $\mu = 2$ and the two right eigenvectors associated with the zero eigenvalue are

$$\mathbf{u}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ \frac{1}{2} \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{u}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \frac{1}{2} \\ 1 \\ 0 \end{pmatrix}. \tag{3.22}$$

Such eigenvectors suggest a coordinate change in the form (3.18)-(3.19), defined as

$$\begin{pmatrix} \overline{\mathbf{x}}_1 \\ \overline{\mathbf{x}}_2 \\ \overline{\mathbf{x}}_3 \\ \overline{\mathbf{x}}_4 \\ \overline{\mathbf{x}}_5 \\ \overline{\mathbf{x}}_6 \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_3 \\ \mathbf{x}_6 \\ \mathbf{x}_2 \\ \mathbf{x}_5 \\ \mathbf{x}_4 \end{pmatrix} \implies \overline{\mathbf{x}} = \begin{pmatrix} \mathbf{e}_1^T \\ \mathbf{e}_3^T \\ \mathbf{e}_6^T \\ \mathbf{e}_2^T \\ \mathbf{e}_5^T \\ \mathbf{e}_4^T \end{pmatrix} \mathbf{x}, \tag{3.23}$$

thus yielding

$$\overline{L} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}, \tag{3.24}$$

which is clearly expressed in the lower-triangular form (3.16), with

$$\begin{aligned} L_1 &= \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix}, \; L_2 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \\ M_1 &= \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \; M_2 = \begin{pmatrix} 1 & 0 \end{pmatrix}, \; M = -2, \end{aligned} \tag{3.25}$$

and $\mathcal{H}_1 = \{w_1, w_3, w_6\}$, $\mathcal{H}_2 = \{w_2, w_5\}$, $\mathcal{C} = \{w_4\}$, as shown in Fig. 3.1.

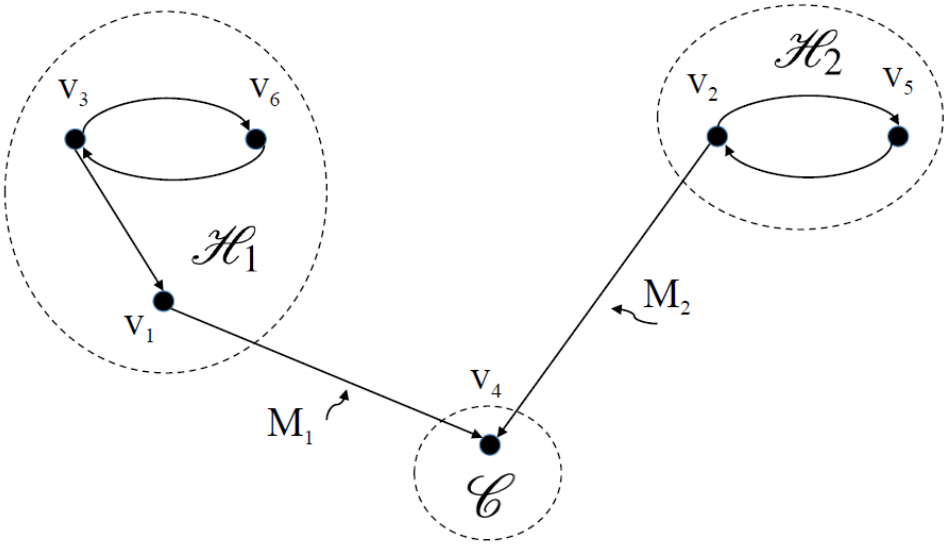


Figure 3.1: Digraph topology proposed in Example 1 and its interpretation in terms of reaches as suggested by (3.16).

From the computational point of view, such a permutation can be obtained from the digraph properties outlined above or, equivalently, by calculating the eigenvectors of the Laplacian matrix and by determining a suitable basis for the eigenspace of dimension μ, as further detailed in Section 3.5. Note that such a basis always exists.

3.4 Almost Equitable Partitions

An important issue in investigating networked dynamical systems is to infer certain network properties from the network topology, which is generally represented by the underlying network digraph. In this respect, some notions from graph theory, in particular the concept of graph partition, have proved rather useful and are reviewed in this Section, since they lay the foundation for the subsequent results reported in Section V.

A well-known case of graph partition is the *equitable partition*, which groups nodes with constant in-degree into cells [77]. The concept of *almost equitable partition* (AEP) – also defined as *external equitable partition* in [133], or *relaxed equitable partition* in [113] – is less restrictive, demanding that the in-degree from nodes in a cell is constant with respect to any other cell but not within each cell.

With respect to the digraph $\mathcal{G}$, for a given cell $\rho \subseteq \mathcal{V}$, we denote the *neighbourhood* of node w_j restricted to cell ρ with

$$\mathcal{N}(w_j, \rho) = \{w_i \in \rho : (w_i, w_j) \in \mathcal{E}\}. \tag{3.26}$$

First, we provide a graph-theoretical definition of an AEP.

Definition 3.5. *(Almost Equitable Partition)* [29, 128] A partition $\pi_{AE} = \{\rho_1, \rho_2, \ldots, \rho_k\}$ is said to be an AEP of $\mathcal{G}$ if, for each $i, j \in \{1, 2, \ldots, k\}$, with $i \neq j$, there exists an integer d_{ij} such that $|\mathcal{N}(w, \rho_j)| = d_{ij}$ for all $w \in \rho_i$, where $|\mathcal{N}|$ denotes the cardinality of the set $\mathcal{N}$. In other words, a partition such that each node in ρ_i has the same number of neighbours in ρ_j, for all i, j with $i \neq j$, is an AEP. □

The property of almost equitability is equivalent to the invariance of the subspaces generated by the characteristic vectors of its cells. So, we can also give an equivalent definition of an AEP, yet this time provided in terms of invariant subspaces.

Definition 3.6. (Alternative to Definition 3.5) A partition π_{AE} is said to be an AEP if and only if $LP(\pi_{AE}) \subset Im\ P(\pi_{AE})$. Hence, the notion of almost equitability of π_{AE} with respect to $\mathcal{G}$ is equivalent to the concept of L-invariance of $Im\ P(\pi_{AE})$. □

Indeed, by making use of geometric tools, one immediately recovers the *quotient graph* representation [133].

Remark 3.5. If π_{AE} is an AEP of $\mathcal{G}$, then, following the proof of Lemma 7 in [128], we have

$$LP(\pi_{AE}) = P(\pi_{AE})L_{\pi_{AE}} \tag{3.27}$$

where $L_{\pi_{AE}}$ is the Laplacian matrix associated with the quotient graph of $\mathcal{G}$ over π_{AE}, i.e.,

$$(L_{\pi_{AE}})_{ij} = \begin{cases} -d_{ij} & \text{if } i \neq j, \\ s_i & \text{otherwise,} \end{cases} \tag{3.28}$$

where $s_i = \sum_{j \neq i} d_{ij}$.

Example 3.2. Consider a digraph $\mathcal{G}$ characterized by the following Laplacian matrix

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 0 & 3 \end{pmatrix}. \tag{3.29}$$

This Laplacian matrix is such that $\mu = 2$ and, by Definition 3.6, the digraph partition $\pi = \{\{w_1\}, \{w_2, w_3\}, \{w_4\},$
$\{w_5\}\}$ is an AEP for $\mathcal{G}$ since $LP(\pi) \subset Im\ P(\pi)$, with

$$P(\pi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.30}$$

The usefulness of such notions is widely documented in the recent literature: e.g., in [128], sufficient conditions in terms of AEPs are derived for diffusively coupled networked systems to be *disturbance decoupled*, and also conditions for guaranteeing the solvability of the disturbance decoupling problem under feedback are provided.

3.5 Main Result Relating Multi-Consensus and Almost Equitable Graph Partitions

We are now in a position to present the main contribution of the paper, thus outlining and clarifying how the network topology determines some of the properties of the dynamic evolution and of the steady-state behaviour of a multi-agent system in the form (3.1). Namely, with respect to Proposition 3.3, when $\mu = 1$, we can make the following statement.

Proposition 3.8. *A partition π_{AE} with characteristic matrix in the $\overline{\mathbf{x}}$-coordinates*

$$P(\pi_{AE}) = \begin{pmatrix} \mathbf{1}_{n-l} & 0_{(n-l)\times l} \\ — & — \\ 0_{l\times(n-l)} & I_l, \end{pmatrix}, \tag{3.31}$$

i.e., such that it divides all the root nodes of $\mathcal{G}$, represented by R, in $(n-l)$ trivial cells and collects the rest of the digraph nodes in another separate cell, with I_l the identity matrix of order l, is an AEP of a digraph $\mathcal{G}$ whose Laplacian is such that $\mu = 1$.

Proof. Since

$$\overline{L}P(\pi_{AE}) = \begin{pmatrix} 0 & 0 \\ M_1 & M \end{pmatrix} \tag{3.32}$$

with $P(\pi_{AE})$ as in (3.31), then

$$\begin{aligned} \overline{L}P(\pi_{AE}) \subset span \begin{pmatrix} 0_{(n-l)\times l} \\ I_l \end{pmatrix} = \\ = span\left\{ \begin{pmatrix} 0 \\ \mathbf{e}_1 \end{pmatrix}, \dots, \begin{pmatrix} 0 \\ \mathbf{e}_l \end{pmatrix} \right\}, \end{aligned} \tag{3.33}$$

with $\mathbf{e}_1, \dots \mathbf{e}_l$ denoting the standard basis of $\mathbb{R}^l$. □

Proposition 3.9. *Given L as the Laplacian of a weakly connected digraph $\mathcal{G}$ with $\mu = 1$, all the nodes in $\mathcal{I}^*$ have the same in-degree evaluated with respect to the $\mathcal{Q}$ set.*

Proof. According to Proposition 3.8,

$$\begin{pmatrix} L & 0 \\ M_1 & M \end{pmatrix} \begin{pmatrix} 0_{(n-l)\times 1} \\ \mathbf{1}_l \end{pmatrix} = \begin{pmatrix} 0_{(n-l)\times 1} \\ M\mathbf{1}_l \end{pmatrix}. \tag{3.34}$$

Moreover, by Definition 3.6, being the partition π_{AE} with characteristic matrix as in (3.31) an AEP of $\mathcal{G}$, then

$$\begin{pmatrix} 0_{(n-l)\times 1} \\ M\mathbf{1}_l \end{pmatrix} \in span \begin{pmatrix} 0_{(n-l)\times 1} \\ \mathbf{1}_l \end{pmatrix}, \tag{3.35}$$

which, by Definition 3.5, implies that all the nodes in $\mathcal{I}^*$ have the same in-degree evaluated with respect to the $\mathcal{Q}$ set. □

Example 3.3. Consider a directed graph $\mathcal{G}$ with Laplacian matrix

$$L = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 2 \end{pmatrix}. \tag{3.36}$$

Let us exchange the agent denoted by w_3 with the agent denoted by w_2, so we choose

$$T_P^{-1} = \begin{pmatrix} \mathbf{e}_1 & \mathbf{e}_3 & \mathbf{e}_2 & \mathbf{e}_4 \end{pmatrix}. \tag{3.37}$$

Then, we obtain

$$\begin{aligned} \overline{L} &= T_P \; L \; T_P^{-1} = \\ &= \begin{pmatrix} \mathbf{e}_1^T \\ \mathbf{e}_3^T \\ \mathbf{e}_2^T \\ \mathbf{e}_4^T \end{pmatrix} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} \mathbf{e}_1 & \mathbf{e}_3 & \mathbf{e}_2 & \mathbf{e}_4 \end{pmatrix} = \\ &= \begin{pmatrix} R & 0 \\ K_1 & K \end{pmatrix}, \end{aligned} \tag{3.38}$$

with $R = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$, $K_1 = \begin{pmatrix} 0 & -1 \\ -1 & -1 \end{pmatrix}$, $K = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$.

Hence, in order to obtain the transformed Laplacian (3.13), we need to have preliminarily identified the $\mathcal{I}^*$ and $\mathcal{Q}$ sets.

More in general, considering the situation when $\mu > 1$, the L_i matrices, for $i = 1, \ldots, \mu$, in (3.16) identify each a balanced subgraph $\mathcal{H}_i$ of $\mathcal{G}$, for $i = 1, \ldots, \mu$, respectively, and such subgraphs $\mathcal{H}_i$ are disconnected from each other. Instead, the union of the common parts of all the reaches, $\mathcal{C}$, certainly contains all the leaves in the digraph. Also, note that the assumption of weak connectedness, according to the dynamics $\dot{\overline{\mathbf{x}}} = -\overline{L}\overline{\mathbf{x}}$ with $\overline{L}$ as in (3.16), implies that each node in $\mathcal{C}$ can be reached starting from any node in each of the $\mathcal{H}_i$'s.

At this point, following Proposition 3.3, we present an interesting result, especially concerning the eigenvectors associated with the zero eigenvalue of L.

Proposition 3.10. *Given L as the Laplacian of a weakly connected digraph $\mathcal{G}$ with $\mu > 1$, after a suitable reordering of the network nodes, such a matrix*

can be rewritten in the form

$$\overline{L}^{'} = T_P^{'}\ L\ (T_P^{'})^{-1}$$
$$\begin{pmatrix} \begin{pmatrix} R_1 & 0 \\ K_{11} & K_1 \end{pmatrix} & 0_{h_1\times h_2} & \dots & 0_{h_1\times h_\mu} & 0_{h_1\times\delta} \\ 0_{h_2\times h_1} & \begin{pmatrix} R_2 & 0 \\ K_{12} & K_2 \end{pmatrix} & \vdots & 0_{h_2\times h_\mu} & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_{h_\mu\times h_1} & 0_{h_\mu\times h_2} & \vdots & \begin{pmatrix} R_\mu & 0 \\ K_{1\mu} & K_\mu \end{pmatrix} & 0_{h_\mu\times\delta} \\ M_1 & M_2 & \dots & M_\mu & M \end{pmatrix}, \tag{3.39}$$

where each $\begin{pmatrix} R_i & 0 \\ K_{1i} & K_i \end{pmatrix}$ *block accounts for the corresponding* L_i *block, for* $i = 1, \dots, \mu$, *in* $\overline{L}$ *(given as in (3.16)), as a result of a coordinate change in the form (3.18)-(3.19) with permutation matrix* $T_P^{'}$, *and also* M *is the same as in (3.16) and accounts for the* $\mathcal{C}$ *set, whose nodes are reachable by any* $\mathcal{H}_i$, $i = 1, \dots, \mu$. *Moreover, a possible basis for the kernel of* L, *i.e., for* $\mathcal{U}$, *generated by the* $\mathbf{u}_i$*'s, takes the form* $\{\mathbf{u}_1, \dots, \mathbf{u}_\mu\}$, *where*

$$\mathbf{u}_1 = \begin{pmatrix} \mathbf{1}_{h_1} \\ 0 \\ \vdots \\ 0 \\ \gamma^1 \end{pmatrix},\ \mathbf{u}_2 = \begin{pmatrix} 0 \\ \mathbf{1}_{h_2} \\ \vdots \\ 0 \\ \gamma^2 \end{pmatrix}, \dots, \mathbf{u}_\mu = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \mathbf{1}_{h_\mu} \\ \gamma^\mu \end{pmatrix}, \tag{3.40}$$

with

$$\gamma^i = \begin{pmatrix} \gamma^i_{w_1} \\ \gamma^i_{w_2} \\ \vdots \\ \gamma^i_{w_\delta} \end{pmatrix}, \tag{3.41}$$

such that $0 < \gamma^i_{w_j} < 1, \forall i \in \{1, \dots, \mu\}, \forall j \in \{1, 2, \dots, \delta\}$, *and* $\sum_{i=1}^{\mu} \gamma^i = \mathbf{1}_\delta$. *Moreover, the corresponding left eigenvectors* $\mathbf{v}_i$, *for* $i = 1, \dots, \mu$, *take the*

form

$$\mathbf{v}_1 = \begin{pmatrix} \begin{pmatrix} \mathbf{v}^{r_1} \\ 0 \end{pmatrix}_{h_1} \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ \mathbf{v}^2 = \begin{pmatrix} 0 \\ \begin{pmatrix} \mathbf{v}_{r_2} \\ 0 \end{pmatrix}_{h_2} \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \quad \ldots, \ \mathbf{v}^\mu = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \begin{pmatrix} \mathbf{v}_{r_m} \\ 0 \end{pmatrix}_{h_\mu} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{3.42}$$

where $r_i := |\mathcal{Q}_i| \leq h_i$, with $\mathcal{Q}_i$ denoting the subset of the roots of the balanced subgraph $\mathcal{H}_i$, for $i = 1, \ldots, \mu$.

Proof. By applying Proposition 3.3 to each L_i block, for $i = 1, \ldots, \mu$, given as in (3.16), we obtain (3.39). For the rest of the proof, refer to Theorem 3.2 in [32].

Note that each $\mathbf{u}_i$ is associated with each $\mathcal{H}_i$, while the vector γ^i in each $\mathbf{u}_i$ is associated with $\mathcal{C}$, i.e., with the union of the common parts of all the reaches, and hence must satisfy the relation $\sum_{i=1}^{\mu} \gamma^i = \mathbf{1}_\delta$.

Moreover, proceeding analogously to the proof of Theorem 3.4, in the $\overline{\mathbf{x}} = T_P' \cdot \mathbf{x}$ coordinates, the left eigenvectors take the form (3.42) and, specifically, the $v_i^{r_z}$'s, for $i = 1, \ldots, r_z$ and $z = 1, \ldots, \mu$, are such that $v_i^{r_z} \neq 0$ and $\sum_{i=1}^{\mu} v_i^{r_z} = 1$. □

Example 3.4. Consider a directed graph $\mathcal{G}$ with Laplacian matrix

$$L = \begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 2 \end{pmatrix}. \tag{3.43}$$

Let us reorder the nodes so that the agents denoted with w_3 and w_4 come first, i.e., let us choose

$$(T_P^{'})^{-1} = \begin{pmatrix} \mathbf{e}_4 & \mathbf{e}_3 & \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_5 \end{pmatrix}, \tag{3.44}$$

then we obtain

$$\begin{aligned} \overline{L} = T_P^{'}\ L\ (T_P^{'})^{-1} = & \begin{pmatrix} \mathbf{e}_4^T \\ \mathbf{e}_3^T \\ \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_5^T \end{pmatrix} \\ \times & \begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} \mathbf{e}_4 & \mathbf{e}_3 & \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_5 \end{pmatrix} = \\ = & \begin{pmatrix} L_1 & 0 & 0 \\ 0 & L_2 & 0 \\ M_1 & M_2 & M \end{pmatrix}, \end{aligned} \tag{3.45}$$

with $L_1 = \begin{pmatrix} 0 \end{pmatrix}$, $L_2 = \begin{pmatrix} 0 \end{pmatrix}$, $M_1 = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$, $M_2 = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$, $M = \begin{pmatrix} 2 & 0 & -1 \\ -1 & 2 & 0 \\ -1 & -1 & 2 \end{pmatrix}$.

Example 3.5. Consider a directed graph $\mathcal{G}$ with Laplacian matrix

$$L = \begin{pmatrix} 6 & 0 & -1 & -1 & 0 & -1 & -1 & -1 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 5 & -1 & -1 & -1 & 0 & 0 & -1 & 0 \\ -1 & 0 & -1 & 5 & 0 & 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & -1 & -1 & 0 & -1 & 0 & 0 & 0 & 5 & -1 \\ -1 & -1 & -1 & -1 & 0 & 0 & 0 & -1 & -1 & 6 \end{pmatrix}. \tag{3.46}$$

Let us exchange the agent denoted by w_1 with the agent denoted by w_6, the agent denoted by w_3 with the agent denoted by w_7, and the agent denoted

by w_4 with the agent denoted by w_8: so we choose a coordinate change with permutation matrix

$$(T_P^{'})^{-1} = \begin{pmatrix} \mathbf{e}_6 & \mathbf{e_2} & \mathbf{e}_7 & \mathbf{e}_8 & \mathbf{e}_5 & \mathbf{e}_1 & \mathbf{e}_3 & \mathbf{e}_4 & \mathbf{e}_9 & \mathbf{e}_{10} \end{pmatrix}. \tag{3.47}$$

Then, we obtain the transformed Laplacian matrix

$$\overline{L} = T_P^{'} L (T_P^{'})^{-1} = \\ = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & 0 & 6 & -1 & -1 & 0 & -1 \\ -1 & -1 & 0 & 0 & -1 & 0 & 5 & -1 & -1 & 0 \\ 0 & 0 & -1 & -1 & 0 & -1 & -1 & 5 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 & -1 & -1 & 0 & 5 & -1 \\ 0 & -1 & 0 & -1 & 0 & -1 & -1 & -1 & -1 & 6 \end{pmatrix}, \tag{3.48}$$

with

$$L_1 = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ -1 & -1 & 2 \end{pmatrix}, \\ L_2 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \tag{3.49}$$

and where, in the original $\mathbf{x}$-coordinates, w_2 and w_6 are the reach root nodes of $\mathcal{H}_1$, and w_5 and w_8 are the reach root nodes of $\mathcal{H}_2$.

Proposition 3.10 can be interpreted as an extension of the related result in [32] concerning the left and right eigenvectors associated with the Laplacian matrix of a weakly connected digraph with $\mu > 1$.

We are now in a position to generalize Proposition 3.8 to the case when $\mu(\lambda_1) = k > 1$, that is, when the Laplacian can be turned into the form (3.39).

Proposition 3.11. *A partition* $\pi'_{AE} = \{\{\mathcal{H}_1\}, \{\mathcal{H}_2\}, \ldots, \{\mathcal{H}_\mu\}, \rho_{\mu+1}, \rho_{\mu+2}, \ldots, \rho_{\mu+\delta}\}$ *with characteristic matrix in the* $\overline{\mathbf{x}}$*-coordinates*

$$P(\pi_{AE}^{'}) = \begin{pmatrix} \mathbf{1}_{h_1} & 0_{h_1 \times 1} & \ldots & 0_{h_1 \times 1} & 0_{h_1 \times \delta} \\ 0_{h_2 \times 1} & \mathbf{1}_{h_2} & \ldots & 0_{h_2 \times 1} & 0_{h_2 \times \delta} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0_{h_\mu \times 1} & 0_{h_\mu \times 1} & \ldots & \mathbf{1}_{h_\mu} & 0_{h_\mu \times \delta} \\ \hline 0_{\delta \times 1} & 0_{\delta \times 1} & \ldots & 0_{\delta \times 1} & I_\delta, \end{pmatrix} \tag{3.50}$$

i.e., such that it collects all the nodes belonging to each $\mathcal{H}_i$ into a separate cell and the rest of the digraph nodes each into a trivial cell $\rho_{\mu+j}$ for $j = 1, \dots, \delta$, with I_δ denoting the identity matrix of order δ, is an AEP of a digraph $\mathcal{G}$ whose Laplacian is such that $\mu > 1$.

Proof. Since

$$\overline{L}P(\pi_{AE}) = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ M_1 & M_2 & \dots & M_\mu & M \end{pmatrix} \tag{3.51}$$

with $P(\pi_{AE})$ as in (3.50), then

$$\begin{aligned} \overline{L}P(\pi_{AE}) &\subset span \begin{pmatrix} 0_{(h_1+\dots+h_\mu)\times\delta)} \\ I_\delta \end{pmatrix} = \\ &= span\left\{ \begin{pmatrix} 0 \\ \mathbf{e}_1 \end{pmatrix}, \dots, \begin{pmatrix} 0 \\ \mathbf{e}_\delta \end{pmatrix} \right\}, \end{aligned} \tag{3.52}$$

with $\mathbf{e}_1, \dots \mathbf{e}_\delta$ denoting the standard basis of $\mathbb{R}^\delta$. □

With this in mind, we can show the interesting result proposed by Theorem 3.12. But, first, we ought to make a few observations.

Assuming that $\mu = 2$, two fundamental relations follow from the properties of the Laplacian matrix and of its eigenvectors as well as from Proposition 3.10:

$$\begin{cases} M_1\mathbf{1}_{h_1} + M_2\mathbf{1}_{h_2} + M\mathbf{1}_\delta = 0 \\ M_1\mathbf{1}_{h_1} + M\gamma^1 = 0. \end{cases} \tag{3.53}$$

These two relations imply that

$$M_2\mathbf{1}_{h_2} + M\gamma^2 = 0, \tag{3.54}$$

with $\gamma^2 = \mathbf{1} - \gamma^1$ according to Proposition 3.10. Since M is nonsingular, it follows from (3.53) that

$$\gamma^1 = -M^{-1}M_1\mathbf{1}_{h_1}. \tag{3.55}$$

Remark 3.6. If $M_1\mathbf{1}_{h_1} = M_2\mathbf{1}_{h_2}$, then $\gamma = \frac{\mathbf{1}_\delta}{2}$, thus implying that the states of all the nodes in the $\mathcal{C}$ set converge to the same value.

Remark 3.7. In general, the components of the γ^i's are such that

$$\gamma^i_{w_j} < 1, \quad j = 1, \dots, \delta. \tag{3.56}$$

Otherwise, they would account for nodes belonging not to the $\mathcal{C}$ set, but to the $\mathcal{H}_i$'s.

Remark 3.8. The value of the components of the γ^i's, for $i = 1, \ldots, k$, do not depend on the location of the links connecting the $\mathcal{H}_i$'s with the $\mathcal{C}$ set (i.e., they do not depend on the location of the nonzero elements appearing in the columns of the M_i matrices).

Example 3.6. Let us consider a digraph characterized by the Laplacian matrix

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix}. \tag{3.57}$$

For such a digraph, the $\mathcal{C}$ set can be shown to contain only node w_3. Let, absurdly, $\mathcal{H}_1 = \{w_1\}, \mathcal{H}_2 = \{w_2\}$, and $\mathcal{C} = \{w_3, w_4\}$. Then, $M\gamma^1 = -M_1\mathbf{1}$, i.e.,

$$\begin{pmatrix} -2 & 1 \\ 0 & -1 \end{pmatrix} \gamma^1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \tag{3.58}$$

which implies that

$$\gamma^1 = \begin{pmatrix} \frac{1}{2} & 0 \end{pmatrix}^T, \quad \gamma^2 = \mathbf{1} - \gamma^1 = \begin{pmatrix} \frac{1}{2} & 1 \end{pmatrix}^T. \tag{3.59}$$

Similarly, since $M\gamma^2 = -M_2\mathbf{1}$, we have

$$\begin{pmatrix} -2 & 1 \\ 0 & -1 \end{pmatrix} (\mathbf{1} - \gamma^1) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \tag{3.60}$$

implying that

$$\mathbf{1} - \gamma^1 = \begin{pmatrix} \frac{1}{2} & 1 \end{pmatrix}^T. \tag{3.61}$$

Then, according to Proposition 3.10, the eigenvectors associated with the zero eigenvalue are

$$\mathbf{u}_1 = \begin{pmatrix} 1 & 0 & \frac{1}{2} & 0 \end{pmatrix}^T, \tag{3.62}$$

$$\mathbf{u}_2 = \begin{pmatrix} 0 & 1 & \frac{1}{2} & 1 \end{pmatrix}^T. \tag{3.63}$$

It is then clear from eigenvector $\mathbf{u}_2$ that node w_4, assumed to belong to $\mathcal{C}$, belongs instead to $\mathcal{H}_2$.

We now outline a particular condition, related to AEPs, that will be instrumental in proving the subsequent Theorem 3.12. This is aimed at determining an AEP, denoted with π^*_{AE}, that is coarser than $\pi^{'}_{AE}$ and such that its $k < \delta$ cells in $\mathcal{C}$ are as many as the groups of identical components in the γ^i vectors, $i = 1, \ldots, \mu$. Indeed, such k cells in $\mathcal{C}$ can be proved to identify the different clusters the considered multi-agent system is divided into at steady state.

Remark 3.9. A necessary external condition for a subset of nodes in $\mathcal{C}$ to compose a cell of an AEP that is coarser than π'_{AE} (only with respect to the trivial cells $\rho_{\mu+j},\ j=1,\dots,\delta$, accounting for the nodes in $\mathcal{C}$) is that they have the same in-degree with respect to each $\mathcal{H}_i,\ i=1,\dots,\mu$. This reflects into a property of the row sums of each M_i matrix in (3.39), for $i=1,\dots,\mu$: more precisely, in the δ-dimensional vector $M_i\mathbf{1}_{h_i}$, the components that are associated with nodes belonging to the same cell $\rho_j,\ j=1,\dots,k$, must be equal.

Theorem 3.12. *A necessary and sufficient condition for two nodes in $\mathcal{C}$ to belong to the same cell ρ of an AEP is that the components of the γ^i vectors, $i=1,\dots,\mu$, associated with the nodes belonging to cell ρ are equal.*

Proof. For the sake of clarity, let the nodes in $\mathcal{C}$ be denoted with $w_1,\dots,w_\delta$ and let $\rho_1,\dots,\rho_k$ denote k cells – each with cardinality $\zeta_i := |\rho_i| \geq 0$ $(i=1,\dots,k)$ so that $\sum_{i=1}^{k}\zeta_i = \delta$ – partitioning $\mathcal{C}$ in the following way:

$$\begin{cases} \rho_1 = \{w_1,\dots,w_{\zeta_1}\}, \\ \rho_2 := \{w_{\zeta_1+1},\dots,w_{\zeta_1+\zeta_2}\}, \\ \vdots \\ \rho_k := \{w_{\zeta_1+\dots+\zeta_{k-1}+1},\dots,w_\delta\}. \end{cases} \tag{3.64}$$

Furthermore, the condition that, for each cell $\rho_j,\ j=1,\dots,k$, the elements of the γ^i vectors, $i=1,\dots,\mu$, associated with the nodes belonging to cell ρ_j be equal can be formalized as follows:

$$\begin{cases} \gamma^i_{w_1} & = \dots = \gamma^i_{w_{\zeta_1}} := \tau^i_1, \\ \gamma^i_{w_{\zeta_1+1}} & = \dots = \gamma^i_{w_{\zeta_1+\zeta_2}} := \tau^i_2, \\ & \vdots \\ \gamma^i_{w_{\zeta_1+\dots\zeta_{k-1}+1}} & = \dots = \gamma^i_{w_\delta} := \tau^i_k, \end{cases} \tag{3.65}$$
$$\forall i \in \{1,2,\dots,\mu\},$$

where $\gamma^i_{w_j}$ denotes the element of the γ^i vector that is associated with node w_j in $\mathcal{C}$ according to (3.64), and $\tau^i_j \in \mathbb{R}$, for $j=1,\dots,k,\ i=1,\dots,\mu$.

Let us first apply the coordinate change

$$\overline{\mathbf{x}} = T'_P \cdot \mathbf{x} = \begin{pmatrix} \overline{\mathbf{x}}_1 \\ \overline{\mathbf{x}}_2 \\ \vdots \\ \overline{\mathbf{x}}_k \\ \overline{\mathbf{x}}_\delta \end{pmatrix}, \tag{3.66}$$

where $T_P^{'}$ is the permutation matrix yielding the transformed Laplacian matrix (3.39). By means of a further coordinate change with respect to the nodes in $\mathcal{C}$ only (identified by $\overline{\mathbf{x}}_\delta$ in the $\overline{\mathbf{x}}$-coordinates), i.e., by relying on

$$\tilde{\mathbf{x}} = T_P^{''} \cdot \overline{\mathbf{x}} = \begin{pmatrix} \overline{\mathbf{x}}_1 \\ \overline{\mathbf{x}}_2 \\ \vdots \\ \overline{\mathbf{x}}_k \\ \tilde{\mathbf{x}}_\delta \end{pmatrix}, \tag{3.67}$$

we reorder the nodes in $\mathcal{C}$, identified in the new $\tilde{\mathbf{x}}$-coordinates by $\tilde{\mathbf{x}}_\delta$, so that

$$\tilde{\mathbf{x}}_\delta = \begin{pmatrix} x_{w_1} \\ \vdots \\ x_{w_{\zeta_1}} \\ x_{w_{\zeta_1+1}} \\ \vdots \\ x_{w_{\zeta_1+\zeta_2}} \\ \vdots \\ x_{w_{\zeta_1+\ldots+\zeta_{k-1}}} \\ \vdots \\ x_{w_\delta} \end{pmatrix}. \tag{3.68}$$

We now have that the system dynamics is expressed by $\dot{\tilde{\mathbf{x}}} = -\tilde{L}\tilde{\mathbf{x}}$, where

$$\tilde{L} = \begin{pmatrix} L_1 & 0 & \cdots & 0 & 0 \\ 0 & L_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & L_\mu & 0 \\ \tilde{M}_1 & \tilde{M}_2 & \cdots & \tilde{M}_\mu & \tilde{M} \end{pmatrix}, \tag{3.69}$$

whose only difference from $\overline{L}$ is due to the reordering of the nodes belonging to $\mathcal{C}$.

(*Necessity*) We need to prove that if the cells $\rho_1, \ldots, \rho_k$ form an AEP among the nodes of the $\mathcal{C}$ set according to (3.64), then, for each cell $\rho_j, j = 1, \ldots, k$, the elements of the γ^i vectors, $i = 1, \ldots, \mu$, associated with the nodes belonging to cell ρ_j are equal, i.e., (3.65) holds.

By assumption, for each cell $\rho_j,\ j = 1, \ldots, k$, the sum of the columns of $\tilde{M}$ associated with the node belonging to cell ρ_j is a linear combination of the

characteristic vectors of the cells $\rho_j,\ j = 1, \ldots, k$, which the $\mathcal{C}$ set is assumed to be partitioned into, i.e.,

$$\tilde{M}p(\rho_j) = \sum_{\eta=1}^{k} \varepsilon_{\eta}^{j} p(\rho_\eta), \quad j = 1, \ldots, k. \tag{3.70}$$

In particular, condition (3.70) is equivalent to saying that each node belonging to any cell $\rho_j,\ j = 1, \ldots, k$, has the same in-degree evaluated with respect to any other cell ρ_θ, for $\theta \in \{1, \ldots, k\} \setminus \{j\}$. Moreover, since $\rho_1, \ldots, \rho_k$ are cells forming an AEP of the $\mathcal{C}$ set by assumption, then the matrix product $\tilde{M}p(\rho_j),\ j = 1, \ldots, k$, is invariant with respect to any further addition of edges inside such cells.

On the other hand, since

$$\tilde{\mathbf{u}}_1 = \begin{pmatrix} \mathbf{1}_{h_1} & 0 \ldots & 0 & \tilde{\gamma}^1 \end{pmatrix}^T, \ldots,\ \tilde{\mathbf{u}}_\mu = \begin{pmatrix} 0 & 0 & \ldots & \mathbf{1}_{h_\mu} & \tilde{\gamma}^\mu \end{pmatrix} \tag{3.71}$$

are right eigenvectors of $\tilde{L}$ in (3.69), it follows that

$$\tilde{M}_i \mathbf{1}_{h_i} + \tilde{M}\tilde{\gamma}^i = 0, \quad i = 1, \ldots, \mu. \tag{3.72}$$

With this in mind, $\tilde{M}$ can be rewritten in the following form:

$$\tilde{M} = \begin{pmatrix} \Gamma_{11} & \Gamma_{12} & \ldots & \Gamma_{1k} \\ \Gamma_{21} & \Gamma_{22} & \ldots & \Gamma_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_{k1} & \Gamma_{k2} & \ldots & \Gamma_{kk} \end{pmatrix}, \tag{3.73}$$

where each Γ_{jz} block is a $\zeta_j \times \zeta_z$ matrix.

In particular, the sum of the columns of the main diagonal blocks Γ_{jj}, each being a nonsingular square matrix of order ζ_j, for $j = 1, \ldots, k$, yields a vector with as many components as the cardinality of cell $\rho_j,\ j = 1, \ldots, k$. Moreover, such a sum is a vector whose components are all equal to a positive integer constant, i.e.,

$$\Gamma_{jj}\mathbf{1}_{\zeta_j} = \sigma \mathbf{1}_{\zeta_j}, \quad \sigma \in \mathbb{Z}^+,\ j = 1, \ldots, k. \tag{3.74}$$

Note also that each off-diagonal block $\Gamma_{jz},\ j = 1, \ldots, k, z = 1, \ldots, k,\ z \neq j$, is such that its row sums are all equal to a nonnegative integer constant, i.e.,

$$\Gamma_{jz}\mathbf{1}_{\zeta_z} = \phi \mathbf{1}_{\zeta_z}, \quad \phi \in \mathbb{N}. \tag{3.75}$$

In plain words, equations (3.70) and (3.74) remain invariant with respect to any change in the position, along the rows, of the ones appearing in the off-diagonal blocks of $\tilde{M}$.

Hence, (3.70), (3.72), (3.74), and (3.75), relying on Remark 3.9, imply that:

$$\begin{aligned}\tilde{M}\tilde{\gamma}^i &= \begin{pmatrix} \Gamma_{11} & \Gamma_{12} & \dots & \Gamma_{1k} \\ \Gamma_{21} & \Gamma_{22} & \dots & \Gamma_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_{k1} & \Gamma_{k2} & \dots & \Gamma_{kk} \end{pmatrix} \begin{pmatrix} \tilde{\gamma}_1^i \\ \vdots \\ \tilde{\gamma}_k^i \end{pmatrix} = \\ &= \sum_{j=1}^{k} \tau_j^i p(\rho_j) = \tau_1^i \begin{pmatrix} \mathbf{1}_{\zeta_1} \\ 0_{\zeta_2} \\ \vdots \\ 0_{\zeta_k} \end{pmatrix} + \tau_2^i \begin{pmatrix} 0_{\zeta_1} \\ \mathbf{1}_{\zeta_2} \\ \vdots \\ 0_{\zeta_k} \end{pmatrix} + \\ &+ \ldots + \tau_k^i \begin{pmatrix} 0_{\zeta_1} \\ 0_{\zeta_2} \\ \vdots \\ \mathbf{1}_{\zeta_k} \end{pmatrix}, \quad i = 1, \dots, \mu, \end{aligned} \tag{3.76}$$

where, for each i, 0_{ζ_i} is a vector whose ζ_i components are all zeros. This allows to recover the thesis, i.e., that the solution $\tilde{\gamma}^i$ to (3.76) is such that $\gamma^i = (T_P'')^{-1}\tilde{\gamma}^i$, with $\tilde{\gamma}_j^i = \tau_j^i \mathbf{1}_{\zeta_j}$, for $i = 1, \dots, \mu$, and $j = 1, \dots, k$, yields (3.65) for a partition of $\mathcal{C}$ defined as in (3.64).

(*Sufficiency*) We now need to prove that, if, for each cell $\rho_j, j = 1, \dots, k$, the elements of the γ^i vectors, $i = 1, \dots, \mu$, associated with the nodes belonging to cell ρ_j are equal (as in (3.65)), then the cells $\rho_1, \dots, \rho_k$ form an AEP among the nodes of the $\mathcal{C}$ set as in (3.64).

First of all, let $\rho_1, \dots, \rho_k$ denote a generic partition of $\mathcal{C}$. The assumption that, for each cell $\rho_j,\ j = 1, \dots, k$, the elements of the γ^i vectors, $i = 1, \dots, \mu$, associated with the nodes belonging to cell ρ_j are equal can be formalized as follows:

$$\gamma^i = \sum_{j=1}^{k} \tau_j^i p(\rho_j), \quad i = 1, \dots, \mu, \tag{3.77}$$

with $\tau_j^i \in \mathbb{R}$, for $i = 1, \dots, \mu,\ j = 1, \dots, k$.

So, we need to prove that (3.72) and (3.77) imply that $\rho_1, \dots, \rho_k$ actually form an AEP of $\mathcal{C}$. Indeed, if, after the coordinate change (3.66), (3.67), and

(3.68), we substitute (3.77) into (3.72), we get:

$$\tilde{M}_i \mathbf{1}_{h_i} + \begin{pmatrix} \Gamma_{11} & \Gamma_{12} & \dots & \Gamma_{1k} \\ \Gamma_{21} & \Gamma_{22} & \dots & \Gamma_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_{k1} & \Gamma_{k2} & \dots & \Gamma_{kk} \end{pmatrix} \cdot T_P'' \cdot \left(\sum_{j=1}^{k} \tau_j^i \; p(\rho_j) \right) = 0, \qquad (3.78)$$

$i = 1, \dots, \mu.$

Such a relation holds only if, in the δ-dimensional vector $\tilde{M}_i \mathbf{1}_{h_i}$, the components that are associated with nodes belonging to the same cell ρ_j are equal, for each cell ρ_j, $j = 1, \dots, k$. Since, by Remark 3.9, this is a necessary condition for $\rho_1, \dots, \rho_k$ to form an AEP of $\mathcal{C}$, then (3.72) and (3.77) imply (3.70), thus recovering the thesis required for the desired sufficiency proof. □

From Theorem 3.12, it follows that the characteristic vectors of the cells of a suitable almost equitable partition denoted with π^*_{AE} (with respect to which ${\pi'}_{AE}$ is finer) identify each a different set of nodes whose states converge to the same steady-state value. More precisely, according to the lower-triangular structure (3.16), the cells of π^*_{AE} are identified by the $\mathcal{H}_i$'s, on the one hand, and by a suitable sub-partition $\rho_1, \dots, \rho_k$ of $\mathcal{C}$ (yielding the relevant property outlined by Theorem 3.12), on the other hand.

Corollary 3.13. *The partition* $\pi^*_{AE} := \{\{\mathcal{H}_1\}, \{\mathcal{H}_2\}, \dots, \{\mathcal{H}_k\}, \; \rho_{\mu+1}, \rho_{\mu+2}, \dots, \rho_{\mu+k}\}$*, with* $k < \delta$*, defined so that each cell* $\rho_{\mu+j}$ *, for* $j = 1, \dots, k$*, groups the nodes in* $\mathcal{C}$ *associated with equal components of the* γ^i *vectors (for* $i = 1, \dots, \mu$*), is the coarsest AEP of* $\mathcal{G}$*, provided that we exclude the trivial partition* $\pi = \rho \equiv \{\mathcal{V}\}$*.*

Proof. The proof follows from Theorem 3.12 and from the fact that a node appearing in any of the $\mathcal{H}_i$'s and a node appearing in $\mathcal{C}$ cannot belong to the same cell of an AEP. □

We can now state the final result linking multi-consensus to the notion of AEP.

Corollary 3.14. *A multi-agent system in the form (3.1), characterized by an underlying weakly connected digraph* $\mathcal{G}$ *with Laplacian matrix such that* $\mu > 1$*, achieves multi-consensus with respect to groups of nodes which coincide with the cells of partition* π^*_{AE}*: namely, for each* $i = 1, \dots, \mu$*, the nodes belonging to* $\mathcal{H}_i$ *converge to a value which depends on the initial conditions of the reach root nodes of* $\mathcal{H}_i$*, whereas, for each* $j = 1, \dots, k$*, the nodes belonging to cell* $\rho_{\mu+j}$ *in* $\mathcal{C}$ *converge to a distinct convex combination of the consensuses achieved within the* $\mathcal{H}_i$*'s (*$i = 1, \dots, \mu$*).*

Proof. The proof follows from Proposition 3.5 and Theorem 3.12. Indeed, let us apply a coordinate change $\overline{\mathbf{x}} = (T_P^{''} \cdot T_P^{'})\ \mathbf{x}$ as in (3.66), (3.67) and (3.68), where each $\overline{\mathbf{x}}_i$ groups the nodes belonging to $\mathcal{H}_i$, for $i = 1, \ldots, \mu$, and $\overline{\mathbf{x}}_\delta$ groups all the nodes belonging to $\mathcal{C}$. Let the vector of the initial conditions of the transformed system be denoted with $\overline{\mathbf{x}}_0$ and composed internally as follows:

$$\overline{\mathbf{x}}_0 = \begin{pmatrix} \begin{pmatrix} \overline{\mathbf{x}}_0^{r_1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{h_1} \\ \begin{pmatrix} \overline{\mathbf{x}}_0^{r_2} \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{h_2} \\ \vdots \\ \begin{pmatrix} \overline{\mathbf{x}}_0^{r_\mu} \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{h_\mu} \end{pmatrix}, \tag{3.79}$$

where, consistently with the coordinate change due to (3.66), (3.67) and (3.68), each $\overline{\mathbf{x}}_0^{r_z}$ vector collects the initial conditions of the reach root nodes of $\mathcal{H}_z$ only (i.e., of the nodes belonging to $\mathcal{Q}_z$), for $z = 1, \ldots, \mu$.

Then, according to Proposition 3.5, the state of the transformed multi-agent system $\dot{\overline{\mathbf{x}}} = -\overline{L}\overline{\mathbf{x}}$, with $\overline{L}$ as in (3.39), converges to a GAS equilibrium state

$$\overline{\mathbf{x}}_\infty^{'} = \begin{pmatrix} \mathbf{1}_{h_1} \\ 0 \\ \vdots \\ 0 \\ \tau_1^1 \mathbf{1}_{\zeta_1} \\ \vdots \\ \tau_k^1 \mathbf{1}_{\zeta_k} \end{pmatrix} \sum_{i=1}^{r_1} v_i^{r_1} \cdot \overline{x}_{0,i}^{r_1} + \begin{pmatrix} 0 \\ \mathbf{1}_{h_2} \\ \vdots \\ 0 \\ \tau_1^2 \mathbf{1}_{\zeta_1} \\ \vdots \\ \tau_k^2 \mathbf{1}_{\zeta_k} \end{pmatrix} \sum_{i=1}^{r_2} v_i^{r_2} \cdot \overline{x}_{0,i}^{r_2} +$$

$$+\ldots+\begin{pmatrix}0\\0\\\vdots\\\mathbf{1}_{h_\mu}\\\tau_1^\mu\mathbf{1}_{\zeta_1}\\\vdots\\\tau_k^\mu\mathbf{1}_{\zeta_k}\end{pmatrix}\sum_{i=1}^{r_\mu}v_i^{r_\mu}\cdot\overline{x}_{0,i}^{r_\mu}=$$

$$=\begin{pmatrix}\mathbf{1}_{h_1}\cdot c^1\\\\\mathbf{1}_{h_2}\cdot c^2\\\vdots\\\mathbf{1}_{h_\mu}\cdot c^\mu\\\mathbf{1}_{\zeta_1}\cdot\sum_{z=1}^{\mu}\tau_1^z c^z\\\vdots\\\mathbf{1}_{\zeta_k}\cdot\sum_{z=1}^{\mu}\tau_k^z c^z\end{pmatrix}, \tag{3.80}$$

where $\overline{x}_{0,i}^{r_z}$ is the i-th component of the $\overline{\mathbf{x}}_0^{r_z}$ vector in (3.79), and the scalars $c^1,\ldots,c^\mu$ are such that

$$c^z=\sum_{\xi=1}^{r_z}v_\xi^{r_z}\cdot\overline{x}_{0,\xi}^{r_z}, \tag{3.81}$$

for $z=1,\ldots,\mu$, with $\tau_i^j\in\mathbb{R}$ for $i=1,\ldots,\zeta_j$ and $j=1,\ldots,k$, and $v_\xi^{r_z}$ denoting the ξ-th component of the $\mathbf{v}^{r_z}$ vector introduced in (3.42), for $\xi=1,\ldots,r_z$. Note that the extrema of the sums are chosen so as to consider the initial conditions of the nodes belonging to $\mathcal{Q}_i$ only ($i=1,\ldots,\mu$). □

Remark 3.10. The γ^i vectors, for $i=1,\ldots,\mu$, appearing in the last equality of (3.80) have δ components which can be divided into groups of $k<\delta$ equal components (i.e., as many as the cells of π_{AE}^* in $\mathcal{C}$) as a consequence of Theorem 3.12. Hence, a system in the form (3.1), with underlying digraph $\mathcal{G}$ such that $\mu>1$, yields $\mu+k$ distinct consensuses, satisfying, according to Definition 3.3,

$$\mathbf{x}\to\mathcal{M}^{\pi_{AE}^*}. \tag{3.82}$$

Such a number of distinct consensuses is equal to the cardinality of π_{AE}^*.

This implies that a group of nodes belonging to $\mathcal{C}$ converge to the same steady-state value if and only if they can be proved to belong to the same cell of partition π_{AE}^*. Such a situation at steady state is clearly a multi-consensus one. In this respect, the examples reported above – namely, those with digraphs such that the Laplacian matrix has $\mu>1$ – yield the following multi-consensus results.

As regards Example 1, the steady state of the related multi-agent system yields 3 distinct consensuses: one for nodes w_1, w_3 and w_6 belonging to $\mathcal{H}_1$, one for nodes w_2 and w_5 belonging to $\mathcal{H}_2$, and the last one for the only node belonging to $\mathcal{C}$.

As regards Example 4, the steady state of the related multi-agent system yields 5 distinct consensuses: one for node w_4 representing $\mathcal{H}_1$, one for node w_3 representing $\mathcal{H}_2$, and one for each of the three trivial cells into which $\mathcal{C}$ can be partitioned.

As regards Example 5, the steady state of the related multi-agent system satisfies Theorem 3.12, Definition 3.13 and Corollary 3.14 with $\mu = 2$ and $k = 2$ – i.e., $\pi^*_{AE} = \{\{\mathcal{H}_1\}, \{\mathcal{H}_2\}, \rho_3, \rho_4\}$. Hence, it yields 4 distinct consensuses: one for nodes w_2, w_6 and w_7 composing $\mathcal{H}_1$, one for nodes w_5 and w_8 composing $\mathcal{H}_2$, one for nodes w_1 and w_3 composing ρ_3, and one for nodes w_4, w_9 and w_{10} composing ρ_4, where ρ_3 and ρ_4 are two cells such that $\rho_3 \cup \rho_4 = \mathcal{C}$, according to the transformed Laplacian matrix (3.48).

Finally, as regards Example 6, the steady state of the related multi-agent system yields 3 distinct consensuses: one for node w_1 representing $\mathcal{H}_1$, one for nodes w_2 and w_4 composing $\mathcal{H}_2$, and one for node w_3 accounting for $\mathcal{C}$.

3.6 Conclusion

In this Chapter, the multi-consensus problem in multi-agent systems has been investigated by putting the Laplacian matrix of the underlying digraph in a form which specializes the one proposed by Caughman and Veerman in [32] and thus gives insight into the topological structure of the networked system itself. On this basis, it is shown that multi-consensus is achieved when the underlying digraph admits a suitable almost equitable partition denoted with π^*_{AE}. In particular, according to the definitions given above, on the one hand, a consensus is achieved for the states of all the nodes belonging to the exclusive part of each digraph reach, while, on the other hand, different consensuses are achieved by the states of the nodes belonging to the union of the common parts of all the reaches, each of these consensuses being a convex combination of the influences of the exclusive reaches. More precisely, the total number of distinct consensuses achieved by the multi-agent system is equal to the number of cells of partition π^*_{AE}. By contrast with the existing literature, in this paper the relationship between the number of cells and the Laplacian is outlined. The results shown in this paper are obtained only for a group of agents whose individual dynamics lies in a one-dimensional space.

We are currently drawing up techniques for efficiently computing the eigenvectors associated with the zero eigenvalue of the digraph Laplacian

according to the structure introduced in Proposition 3.10, especially in the context of complex networks – namely, networks with an extremely large number of agents, with large algebraic multiplicity of the zero eigenvalue of the Laplacian matrix, and yielding a large number of distinct consensuses.

Moreover, further study will be focused on multi-consensusability, on the one hand, when the single agents evolve in n-dimensional spaces ($n > 1$) and, on the other hand, in multi-agent systems with switching topologies and subject to time delays.

Chapter 4

Sampled-Data Design with Aerospace Applications

This Chapter discusses the design and simulation of multirate sampled-data nonlinear feedback control laws in the context of PVTOL maneuvering in the presence of input delays (see Section 4.1) and of spacecraft quasi-halo orbit following about the L_2 translunar libration point (see Section 4.2), as proposed in [157] [158]. In particular, it presents some slight improvements (compared to the existing literature) in terms of feedback finite discretizability, which can be exploited to achieve satisfactory performance results by means of suitable control designs, even in the presence of input delays.

4.1 On the Exact Steering of Finite Sampled Nonlinear Dynamics with Input Delays

Actuator and sensor delays are among the most common phenomena that arise in engineering practice, which is why the compensation of such delays has become an active area of research during the last five decades (e.g., see [170], [8], [114], [43] and [115]). In particular, the control (e.g., stabilization) of dynamical systems with input delays can be performed by means of a predictor-based approach and recent studies have shed light on new opportunities for predictor feedback, such as its extension to nonlinear systems (e.g., see [98] and [75]).

While studying time-delay systems in continuous time typically relies on setting an infinite dimensional problem, the presence of input delays in discrete time can be dealt with by considering appropriate extensions of the dynamics. In this case, the design of a controller is greatly simplified as long as the problem can be re-stated in terms of its sampled-data equivalent

model. For instance, considering continuous-time input-affine dynamics which admit finitely computable discrete-time equivalent models (e.g., see [122], [52], and [123]), a sampled-data predictor-based stabilizing and delay-compensating controller can be designed. Such an approach has been proposed in [125] and then applied to the example of a wheeled mobile robot in [175]. The aim of the present paper consists in showing how this approach – consisting of a piecewise-continuous overall control law obtained by combining preliminary continuous predictor feedback with multirate digital control – can be successfully extended to nonlinear dynamics with a non-zero drift term exhibiting, through preliminary state feedback and a change of coordinates, a finitely computable discrete-time equivalent model. In particular, the delay-free multirate digital control law introduced in [126] is completed with a state predictor for compensating the input delays.

In general, computing the state predictor in a nonlinear context is a very challenging task which almost always compels the designer to resort to approximate models. Nevertheless, should the system turn out to be finitely computable under some suitable state feedback and coordinate transformation, there will be a significant advantage as, in such a case, we can determine the desired control law exactly.

An interesting case study, which is frequently encountered in the literature, is represented by the *Planar Vertical Take-Off and Landing* (PVTOL) aircraft. This dynamical system is based on a simplified aircraft model with a minimal number of states and inputs. It is well known in the control community as it retains the main features that must be considered when designing control laws for a real aircraft. It has attracted and still continues attracting the attention of researchers as it offers the opportunity to pose challenging nonlinear control problems. Over the years, several control designs for stabilization and trajectory tracking have been proposed for the PVTOL aircraft model (e.g., see [81], [165], [112], and [137]).

Although the list of works on PVTOL aircraft control we have given is not even remotely exhaustive, to the best of our knowledge all the results available in the literature, except for [69], assume that there is no delay in the inputs. Nonetheless, such a delay, due to sensors and information processing, is often present in practice. For instance, in [140] the position and roll angle of the aircraft are measured by means of a vision system that induces a delay of approximately 40 ms.

Referring to the dynamics of a PVTOL aircraft as an example of slightly non minimum-phase system or ε non minimum-phase system [81], in [54] the authors pointed out that, under suitable ε-dependent state feedback and change of coordinates, the PVTOL aircraft dynamics admits a finite sampled representation. Therefore, a preliminary feedback and a change of coordinates

render the modified dynamics directly invertible under a digital controller of a suitable multirate order. The introduction of a digital control loop brings to a piecewise-continuous control law which can steer the state from any initial value to any prefixed final one. Our contribution consists in showing that the exact steering control result is preserved even in the presence of input delays as long as we resort to predictor feedback.

4.1.1 Theoretical Framework

General Remarks on Sampled-Data Design

Let us consider the continuous-time input-affine dynamics

$$\begin{aligned} \Sigma: \ \dot{x}(t) &= f(x(t)) + g(x(t))u(t) = \\ &= f(x(t)) + \sum_{i=1}^{m} g_i(x(t))u_i(t) \end{aligned} \tag{4.1}$$

where $x \in M \subset \mathbb{R}^n$ denotes the continuous-time state of the system, $u \in U \subset \mathbb{R}^m$ and $f, g_1 \ldots, g_m$ are real analytic vector fields on the smooth manifold $M \subset \mathbb{R}^n$.

The sampled-data equivalent model of (4.1) is given by

$$\begin{aligned} \Sigma_D := x[k+1] = \\ \exp\left\{ T_s L_{f(x)+\sum_{i=1}^{m} u_i[k] g_i(x)} \right\} (Id)\Big|_{x=x[k]} = \\ = F(T_s; x[k], u[k]), \end{aligned} \tag{4.2}$$

where T_s denotes a sampling period that is small enough to guarantee the convergence of the series expansion manipulated, and the inputs u_i are assumed to be constant over time intervals of length T_s.

Definition 4.1. If there exists a function $F : M \times U_0 \to M, U_0 \subset U$, sum of the series in (4.2), such a function F can be denoted as the sampled closed form of (4.1). Therefore, (4.2) is said to be a Computable Exact Sampled Representation (C-ESR) of (4.1) whenever a closed form for $F(T_s; x[k], u[k])$ exists. Moreover, if the formal exponential series in (4.2) is finite, that is, $\exists k$ such that for any $(x, u) \in M \times U$ the series is zero $\forall k \geq \overline{k}$ and $\forall u[k] \in U$, then (4.2) is said to be the finite discretization of (4.1) or, alternatively, a Finitely Computable Exact Sampled Representation (FC-ESR) of (4.1).

□

Definition 4.2. It is worthy of note that finitely discretizable dynamics are transformed, under coordinate changes, into sampled closed representations, that is, an FC-ESR, under a generic diffeomorphism, is transformed into a C-ESR. Therefore, the notion of finite discretizability is not coordinate free, whereas the existence of closed forms is.

□

In particular, with the aim of further understanding the proposed ideas in the framework of sampled-data representations of nonlinear systems with a drift term as in (4.1), the following interesting commutation relations will now be outlined.

By discretizing the system (4.1) first and applying some coordinate transformation $\xi = \Phi(x)$ afterwards, we get exactly the same system as the one we obtain by applying the same change of coordinates first and then discretizing, that is, $\Sigma_{D\Phi} \equiv \Sigma_{\Phi D}$.

In a similar way, by applying some continuous-time state feedback γ first and then the coordinate transformation Φ, we get the same system as the one we obtain by applying the same coordinate transformation first and then the state feedback γ expressed in the new coordinates ξ, that is, $\Sigma_{\gamma\Phi} \equiv \Sigma_{\Phi\gamma}$.

Furthermore, discretizing, applying some discrete-time feedback γ_D and changing the coordinates is exactly the same as performing the last two modifications the other way around, that is, $\Sigma_{D\gamma_D\Phi} \equiv \Sigma_{D\Phi\gamma_D}$.

By contrast, applying the state feedback γ first and then performing the discretization is not the same as discretizing and later applying the discrete-time feedback γ_D, that is, in general $\Sigma_{\gamma D} \not\equiv \Sigma_{D\gamma_D}$, unless we use a specific $\gamma_D^\star$ which has turned out to be capable of bridging such a gap. This is exactly the multirate digital controller, so long as we assume that the multirate degrees r_i of each input are such that $\sum r_i = n$.

Similarly, $\Sigma_{D\Phi\gamma_D} \not\equiv \Sigma_{\gamma\Phi D}$ in general, but the multirate controller $\gamma_D^\star$ is such that $\Sigma_{D\Phi\gamma_D^\star} \equiv \Sigma_{\gamma\Phi D}$.

The relations above can be summarized in the proposition below.

Proposition 4.1. *In the context of sampled-data representations of nonlinear systems with a drift term, the following relations hold:*

$$\begin{cases} \Sigma_{D\Phi} & \equiv \Sigma_{\Phi D}, \\ \Sigma_{\gamma\Phi} & \equiv \Sigma_{\Phi\gamma}, \\ \Sigma_{D\gamma_D\Phi} & \equiv \Sigma_{D\Phi\gamma_D}, \\ \Sigma_{D\gamma_D^\star} & \equiv \Sigma_{\gamma D}, \\ \Sigma_{D\Phi\gamma_D^\star} & \equiv \Sigma_{\gamma\Phi D}. \end{cases}$$

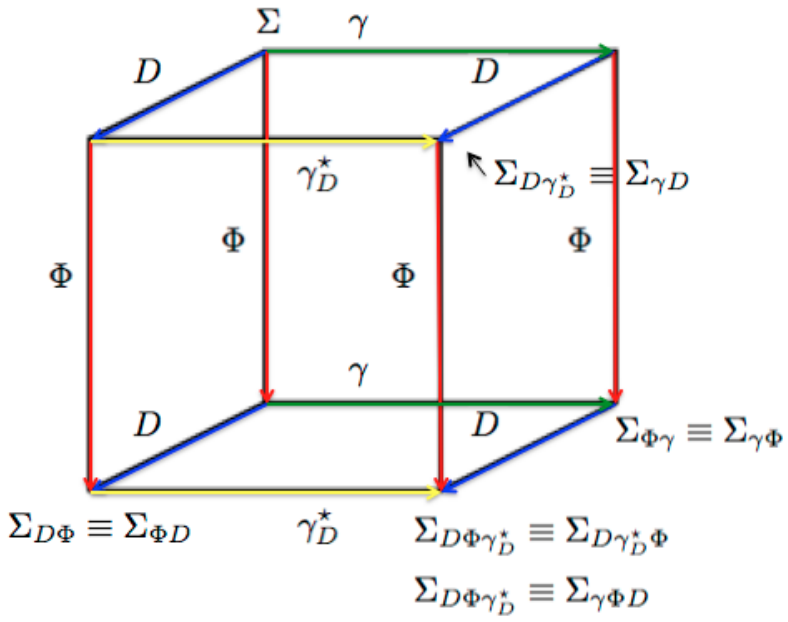


Figure 4.1: Commutation properties within the framework of nonlinear systems with a drift term.

Sampled-data design under input delays

Let us now investigate how the general properties discussed above can be fruitfully employed for designing an exact steering controller for nonlinear dynamics with delays in the inputs.

In particular, studying the sampled-data equivalent model of a time-delay system allows to overcome the problem of infinite dimensionality that arises in continuous time. Moreover, the preservation of feedback finite discretizability under input delays overcomes the difficulty that arises in predictor computability. Indeed, it allows us to compute the state predictor exactly, i.e., without resorting to approximate models.

Following [122] and [52], we will now consider the assumption below.

Assumption 4.1. *There exist a change of coordinates*

$$\xi = \Phi(x) \tag{4.3}$$

and preliminary state feedback

$$u = \gamma(x, v) = \gamma_0(x) + \gamma_1(x)v, \tag{4.4}$$

under which the dynamics (4.1) is transformed into the finitely discretizable system

$$\Sigma_{\gamma\Phi}: \ \dot{\xi} = \tilde{f}(\xi) + \tilde{g}(\xi)v = \tilde{f}(\xi) + \sum_{i=1}^{m} \tilde{g}_i(\xi)v_i. \tag{4.5}$$

Moreover, with reference to the sampled-data equivalent model of (4.5), which is given by

$$\begin{aligned} \Sigma_{\gamma\Phi D} := \xi[k+1] = & \\ & \exp\left\{T_s L_{\tilde{f}(\xi)+\sum_{i=1}^{m}\tilde{g}_i(\xi)v_i[k]}\right\}(Id)\Big|_{\xi=\xi[k]} = \\ & = F(T_s;\xi[k],v[k]), \end{aligned} \tag{4.6}$$

we assume that there exists a multirate sampled-data control law (playing the role of $\gamma_D^\star$ in Section 2.1)

$$v[k] = F^{-1}(T_s;\xi[k],\xi[k+1]) := \gamma_D^\star(\xi[k],\xi[k+1]) \tag{4.7}$$

which satisfies a specific control objective (e.g., steering the evolution of (4.6) to a given target position $\xi(k+1) = \xi^{final}$ in one step of amplitude T_s).

□

Our aim is now to show that finite sampling under coordinate transformation and feedback is preserved even in the presence of input delays, as long as we employ a predictor-based controller. Let us consider, for instance, an input delay that is a multiple of the sampling period T_s, i.e.,

$$\Sigma_\tau : \dot{x} = f(x) + g(x)u(t-\tau) \tag{4.8}$$

where $\tau = NT_s$ with $N \in \mathbb{Z}^+$. By resorting to the same change of coordinates as in (4.3) and to the following preliminary state feedback

$$u(t-\tau) = \gamma(x(t), v(t-\tau)) = \gamma_0(x(t)) + \gamma_1(x(t))v(t-\tau),$$

we get the input-delayed transformed representation of the system dynamics

$$\Sigma_{\tau(\gamma\Phi)} : \dot{\xi} = \tilde{f}(\xi) + \tilde{g}(\xi)v(t-\tau).$$

In the presence of an input delay $\tau = NT_s$, we can use the following solution

$$\begin{aligned} u(t-NT_s) &= \gamma(x(t), v(t-NT_s)) = \\ &= \gamma_0(x(t)) + \gamma_1(x(t))v[k-NT_s], \end{aligned}$$

for $t \in [kT_s, (k+1)T_s[$, where we choose $v[k-NT_s] = \gamma_D^\star(\xi[k], \xi^{final})$.

Such a controller can be rewritten in terms of the predicted state $z(t) = \xi(t + NT_s)$ as

$$v[k] = \gamma_D^\star(z[k], \xi^{final}). \tag{4.9}$$

This way, we recover, with respect to the predicted state z, the delay-free and delay-compensating controller

$$u(t) = \gamma(z(t), v[k]) = \gamma_0(z(t)) + \gamma_1(z(t))v[k], \tag{4.10}$$

for $t \in [kT_s, (k+1)T_s[$. The discrete-time state predictor values $z[k]$ can be computed – with suitable initial conditions $(z[0], v[-1], \ldots, v[-N])$ – directly from the sampled dynamics (4.6) through N compositions as

$$z[k] = F'(T_s; \xi[k], v[k - T_s], v[k - 2T_s], \ldots, v[k - NT_s]), \tag{4.11}$$

and this expression turns out to be finite as (4.5) has been assumed to be finitely discretizable.

Therefore, thanks to a suitable change of coordinates and preliminary state feedback, an exact point-to-point steering controller can be computed in the delay-free case and, in the presence of input delays, a steering controller based on the state predictor can be exactly computed, too. As a result, we can state the following proposition.

Proposition 4.2. *By employing the same change of coordinates (4.3) and preliminary state feedback (4.4) as in the undelayed case, feedback finite discretizability of a nonlinear system with a drift term is maintained in the presence of input delays thanks to the predictor-based control law in (4.9)-(4.10).*

In general, this strategy can be applied to all input-affine nonlinear dynamics admitting, through preliminary transformations (i.e., change of coordinates plus state feedback) exact sampled-data equivalent dynamics.

Hence, feedback finite discretizability ensures that the implementation of the predictor-based controller designed on the basis of the discrete-time equivalent model (which is modified by feedback) can be realized by applying to the system with input delays the same control law as in the undelayed case. This allows us to state another proposition, thus adding a further result to those presented at the end of Section 2.1.

Proposition 4.3. *In a nonlinear system with a drift term described by (4.1), the insertion of a time delay τ and the application of a finitely discretizing feedback do commute with each other, i.e.,*

$$\Sigma_{(\gamma \Phi D)\tau} \equiv \Sigma_{\tau(\gamma \Phi D)}$$

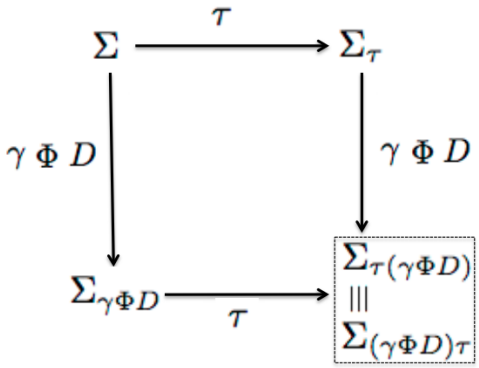


Figure 4.2: The insertion of a time delay τ and the application of a finitely discretizing feedback commute with each other.

4.1.2 The PVTOL Example

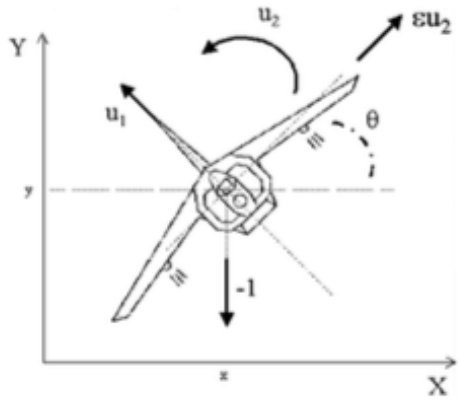


Figure 4.3: The PVTOL aircraft (front view).

The mathematical model of a PVTOL aircraft, under the usual simplifying assumptions, takes the following form

$$\begin{aligned} \ddot{x} &= -\sin(\theta)u_1(t-\tau_1) + \varepsilon\cos(\theta)u_2(t-\tau_2) \\ \ddot{y} &= \cos(\theta)u_1(t-\tau_1) - 1 + \varepsilon\sin(\theta)u_2(t-\tau_2) \\ \ddot{\theta} &= u_2(t-\tau_2) \end{aligned} \tag{4.12}$$

The variables x and y denote the horizontal and vertical position of the aircraft, respectively. θ is the roll angle that the aircraft makes with the

horizon, the control inputs u_1 and u_2 represent, respectively, the total thrust and the angular acceleration (rolling moment) and $\tau_1 > 0$ and $\tau_2 > 0$ are the delays acting on them. The constant term (-1) is the normalized gravitational acceleration. Note that the coefficient $\varepsilon = (J\tan(\alpha))/(mgl) \neq 0$ represents the coupling between the rolling moment and the lateral acceleration of the aircraft [137]. Since ε explicitly depends on the physical parameters of the aircraft that can be measured, the assumption that ε is known is justified. In the expression above, J represents the moment of inertia of the aircraft, α the fixed angle of the wings with respect to the horizontal line, m the mass of the aircraft, l the wing length and g the acceleration due to gravity.

The equations (4.12) are equivalent to a six-dimensional dynamical system in the form of (4.1), i.e., $\dot{\eta} = f + g_1 u_1 + g_2 u_2$, where the state space variables are collected in the vector

$$\eta = \begin{pmatrix} x & \dot{x} & y & \dot{y} & \theta & \dot{\theta} \end{pmatrix},$$

the drift vector field is

$$f = \eta_2 \frac{\partial}{\partial \eta_1} + \eta_4 \frac{\partial}{\partial \eta_3} - \frac{\partial}{\partial \eta_4} + \eta_6 \frac{\partial}{\partial \eta_5}$$

and the two delayed inputs u_1 and u_2 act along the two vector fields

$$\begin{aligned} g_1 &= -\sin(\eta_5)\frac{\partial}{\partial \eta_2} + \cos(\eta_5)\frac{\partial}{\partial \eta_4} \\ g_2 &= \varepsilon\cos(\eta_5)\frac{\partial}{\partial \eta_2} + \varepsilon\sin(\eta_5)\frac{\partial}{\partial \eta_4} + \frac{\partial}{\partial \eta_6}, \end{aligned}$$

respectively.

As stated in [83] and [52], the design procedure sketched in the previous section requires the vector fields f, g_1, g_2 describing the given dynamics to be transformed in order for the resulting vector fields $\tilde{f}, \tilde{g_1}, \tilde{g_2}$ to generate a nilpotent distribution. Therefore, as shown in [54], we consider the feedback

$$u = \begin{pmatrix} \frac{1}{cos(\eta_5)} + \varepsilon\eta_6^2 \\ -2\eta_6^2\tan(\eta_5) \end{pmatrix} + \begin{pmatrix} \frac{1}{cos(\eta_5)} & 0 \\ 0 & \cos^2(\eta_5) \end{pmatrix} v, \tag{4.13}$$

which plays the role of the state feedback (4.4) introduced above. The feedback (4.13) makes it possible to compute a change of coordinates which brings the vector fields to a certain polynomial subtriangular form, thus achieving finite discretizability. According to [52] and [54], we may choose the following state

space diffeomorphism

$$\xi = \Phi(\eta) = \begin{pmatrix} y \\ \dot{y} \\ \tan(\theta) \\ x \\ \frac{\dot{\theta}}{\cos^2(\theta)} \\ \dot{x} \end{pmatrix} + \varepsilon \begin{pmatrix} \cos(\theta) \\ -\dot{\theta}\sin(\theta) \\ 0 \\ -\sin(\theta) \\ 0 \\ -\dot{\theta}\cos(\theta) \end{pmatrix}.$$

In these new coordinates the system dynamics is described by the following vector fields

$$\begin{aligned} \tilde{f} &= \xi_2 \frac{\partial}{\partial \xi_1} + \xi_5 \frac{\partial}{\partial \xi_3} + \xi_6 \frac{\partial}{\partial \xi_4} - \xi_3 \frac{\partial}{\partial \xi_6} \\ \tilde{g}_1 &= \frac{\partial}{\partial \xi_2} - \xi_3 \frac{\partial}{\partial \xi_6} \\ \tilde{g}_2 &= \frac{\partial}{\partial \xi_5}. \end{aligned}$$

Then, assuming for now the absence of time delays on the control inputs, we get the following finite sampled dynamics

$$\begin{aligned} \xi_1[k+1] &= \xi_1[k] + T_s\xi_2[k] + \frac{T_s^2}{2!}v_1[k] \\ \xi_2[k+1] &= \xi_2[k] + T_s v_1[k] \\ \xi_3[k+1] &= \xi_3[k] + T_s\xi_5[k] + \frac{T_s^2}{2!}v_2[k] \\ \xi_4[k+1] &= \xi_4[k] + T_s\xi_6[k] - \frac{T_s^2}{2!}\xi_3[k](1+v_1[k]) + \\ &\quad - \frac{T_s^3}{3!}\xi_5[k](1+v_1[k]) - \frac{T_s^4}{4!}v_2[k](1+v_1[k]) \\ \xi_5[k+1] &= \xi_5[k] + T_s v_2[k] \\ \xi_6[k+1] &= \xi_6[k] - T_s\xi_3[k](1+v_1[k]) + \\ &\quad - \frac{T_s^2}{2!}\xi_5[k](1+v_1[k]) - \frac{T_s^3}{3!}v_2[k](1+v_1[k]) \end{aligned}$$

Now, following the procedure described in [122], i.e., denoting the multirate degree of v_1 and v_2 with r_1 and r_2, respectively, we choose $r_1 = 2$ and $r_2 = 4$, which lead us to a solvable system of equations satisfying the condition $r_1 + r_2 = 6$. Such a choice for the multirate degrees guarantees that, by

setting

$$\begin{aligned} v_1[k] = v_{1,i} \text{ in } \left[\left(k+\frac{i-1}{2}\right)T_s, \left(k+\frac{i}{2}\right)T_s\right[\text{ for } i=1,2, \\ v_2[k] = v_{2,i} \text{ in } \left[\left(k+\frac{i-1}{4}\right)T_s, \left(k+\frac{i}{4}\right)T_s\right[\\ \text{for } i=1,2,3,4, \end{aligned} \tag{4.14}$$

the resulting finite multirate discrete-time dynamics

$$\begin{aligned} \xi[k+1] &= F(T_s;\xi[k],v_{1,1},v_{1,2},v_{2,1},v_{2,2},v_{2,3},v_{2,4}) = \\ &= F(T_s;\xi[k],v[k]) \end{aligned} \tag{4.15}$$

will be fully invertible with respect to the control inputs $v_{i,j}$ for any value of $\xi[k]$ and $\xi[k+1]$, thus giving

$$v[k] = F^{-1}(T_s;\xi[k],\xi[k+1]). \tag{4.16}$$

Hence, having fixed an initial state $\xi^{initial}$ as well as the desired final one ξ^{final}, v_1 and v_2 in (4.14) can be computed from (4.15) by a simple map inversion, due to the polynomial expression of (4.15) with respect to $v_{i,j}$, once $\xi[k] = \xi^{initial} = \Phi(\eta^{initial})$ and $\xi[k+1] = \xi^{final} = \Phi(\eta^{final})$ are posed. The final control scheme is then composed of the continuous feedback (4.13) together with the digital one in the form (4.16).

Introducing an input delay $\tau_1 = \tau_2 = NT_s$ with $N \in \mathbb{Z}^+$ in (4.12), the same approach can be performed in order to control the corresponding input-delayed transformed dynamics

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= v_1(t-NT_s) \\ \dot{\xi}_3 &= \xi_5 \\ \dot{\xi}_4 &= \xi_6 \\ \dot{\xi}_5 &= v_2(t-NT_s) \\ \dot{\xi}_6 &= -\xi_3 - \xi_3 v_1(t-NT_s), \end{aligned} \tag{4.17}$$

by resorting to the following piecewise-continuous overall solution

$$\begin{aligned} u(t-NT_s) = &\begin{pmatrix} \frac{1}{cos(\eta_5(t))} + \varepsilon\eta_6^2(t) \\ -2\eta_6^2(t)\tan(\eta_5(t)) \end{pmatrix} + \\ &+ \begin{pmatrix} \frac{1}{cos(\eta_5(t))} & 0 \\ 0 & \cos^2(\eta_5(t)) \end{pmatrix} v[k-NT_s] \end{aligned}$$

with $v[k - NT_s] = F^{-1}(T_s; \xi[k], \xi^{final})$ and F^{-1} obtained from map inversion of the finite multirate sampled dynamics (4.15) when setting $\xi[k+1] = \xi^{final}$.

Such a controller can be rewritten in terms of the predicted state $z(t) = \xi(t + NT_s)$ as

$$v[k] = F^{-1}(T_s; z[k], \xi^{final}). \tag{4.18}$$

This way, we recover, with respect to the predicted state z, the delay-free and delay-compensating controller (4.10), where the discrete-time predictor dynamics, required to calculate $v[k]$, can be easily computed from the multirate sampled dynamics (4.15) through N compositions as (4.11), with suitable initial conditions $(z[0], v[-1], \ldots, v[-N])$ so that $z[0] = \xi[N]$.

4.1.3 Simulation Results

The above results have been verified on an elementary maneuver for the PVTOL motion, aimed at achieving some lateral displacement, i.e., moving the aircraft from the starting position $(x, y) = (0, 0)$ to a predetermined final one, say, $(x, y) = (1, 0)$. This maneuver is usually addressed in the literature as a case study (e.g., see [81]). A steering maneuver from the initial state $\eta^{initial} = (0, 0, 0, 0, 0, 0)^T$ to the final one $\eta^{final} = (1, 0, 0, 0, 0, 0)^T$ has been simulated, corresponding to a lateral displacement of about 10 meters, with the parameter $\varepsilon = 0.8$.

It is worthy of note that the presented multirate digital approach brings to an overall control law u which turns out to be piecewise-continuous, as it combines the continuous-time state feedback with the piecewise-constant control v.

Should we try to use the multirate controller (4.13)-(4.16) in the presence of input delays without employing any state predictor, it is clear that such an approach would not work: as soon as an input delay τ is introduced in the system dynamics, the controller will not generate the right input signal any more.

However, in the presence of an input delay $\tau_1 = \tau_2 = NT_s$ and assuming $N = 1$ for the sake of simplicity, by means of the predictor feedback (4.10)-(4.18) the convergence of the dynamical system to the desired final position will be achieved in two sampling steps. In other words, the closed-loop system is delay-compensated only after the predictor-based controller "kicks in" at $t = T_s$.

Under the above assumptions on the values of τ_1, τ_2 and N and choosing the sampling time $T_s = 10$ s, the results shown in Figs. 4.4-**??** are obtained. The dashed lines in Figs. 4.4-4.5 depict the behaviour of the state variables under the multirate controller (4.13)-(4.16) in the undelayed case, whereas the

solid lines depict the behaviour of the state variables under the predictor-based control law (4.10)-(4.18) in the delayed case.

In Fig. 3 the dashed lines show the control inputs u_i for $i = 1, 2$ in the undelayed case, whereas the solid lines show $u_i(t - \tau_i)$ for $i = 1, 2$ in the delayed case under the predictor-based controller.

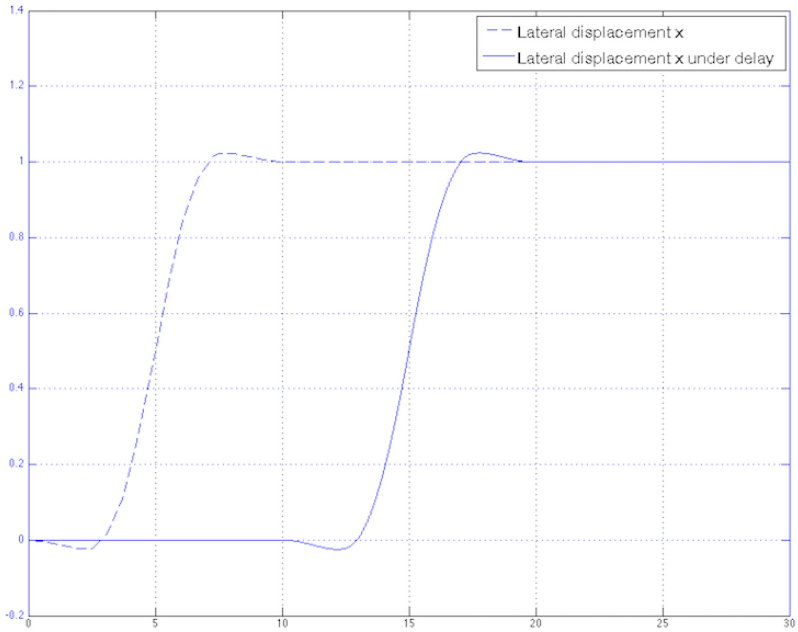


Figure 4.4: $T_s = 10\ s$: the solid line represents x versus t in the delayed case, whereas the dashed line represents it in the undelayed case.

Moreover, Table 1 collects the values of the integral $\int u_2^2(t)dt$ (referring to the delayed case shown in Fig. 3), which can be considered as an "energy-like" function $E(T_s)$. The corresponding contribution due to u_1 is negligible.

Table 4.1: Normalized values of $E(T_s)$

Time	$E(T_s)$
10	0
15	1.63
20	3.26

The illustrated example is intended to show the effectiveness of the proposed control strategy and to check the validity of the formulae presented in

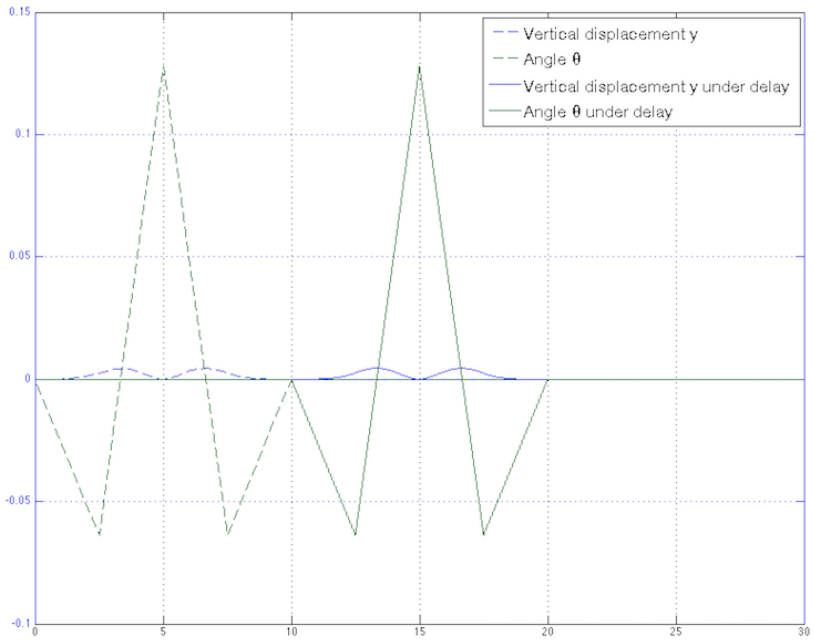


Figure 4.5: $T_s = 10\ s$: the solid lines represent y and θ versus t in the delayed case, whereas the dashed lines represent them in the undelayed case.

the previous sections.

4.1.4 Conclusions

Although in general the computation of the state predictor for a nonlinear system is a very challenging task – sometimes even almost impossible to solve – and requires the designer to rely on approximate models, we have outlined the interesting role played by feedback finite discretizability, which offers us the opportunity to calculate the exact state predictor whenever some specific preliminary transformations are operated on the nonlinear dynamics under scrutiny. Moreover, we have shown that feedback finite discretizability commutes with the insertion of a time delay in the nonlinear dynamics. From a practical point of view, we have also illustrated the possibility of applying the theoretical results for nonlinear systems with delays in the inputs to a case study, namely a physical system, designing an exact delay-compensating piecewise-continuous steering controller. Yet, much remains to be done, e.g., extending the obtained results to the case when the exact values of the delays are unknown and also achieving good performances in terms of the robustness

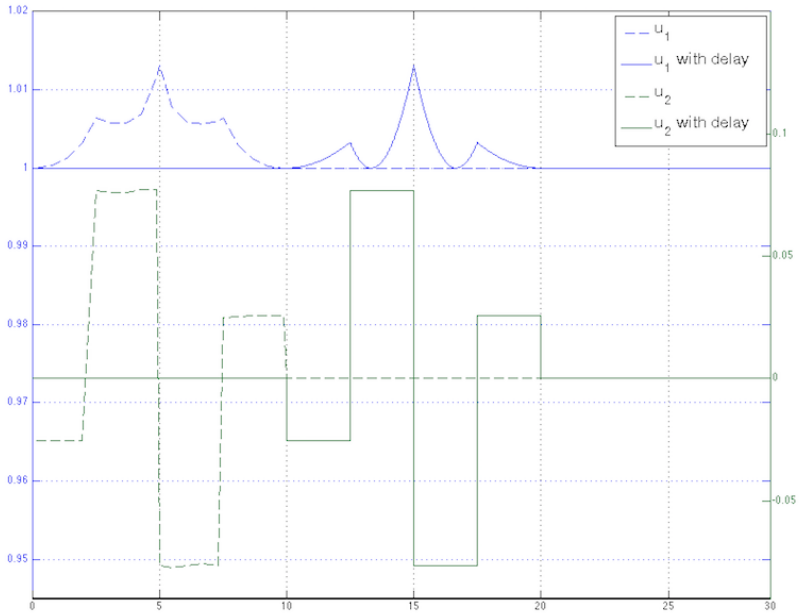


Figure 4.6: $T_s = 10\ s$: the solid lines represent $u_1(t - T_s)$ and $u_2(t - T_s)$ versus t in the delayed case, whereas the dashed lines represent u_1 and u_2 in the undelayed case.

of the presented control scheme with respect to disturbances.

4.2 Sampled-Data Stabilization Around the L_2 Translunar Libration Point

Space missions near the Lagrangian points (also known as libration points) of the Sun-Earth system and of the Earth-Moon system have always attracted significant attention for their strategic benefits in long-term astronomical observation. In particular, ARTEMIS-P1 has been the first spacecraft to navigate to and perform station-keeping operations around the Earth-Moon L_1 and L_2 libration points.

Halo orbits around the L_2 translunar libration point in the Earth-Moon system have the property of ensuring visibility both from the dark side of the Moon and from the Earth at any time. Hence, they offer the opportunity to establish a bridge for radio communications, which is a crucial problem for space missions planning to use the dark side of the Moon either as a powerful observation point or as a launch site for low-frequency radio astronomy, deep

space observation, solar system exploration and other scientific researches in a low-gravity environment. However, it is well known that L_2 is unstable, which implies the instability of any equilibrium trajectory around L_2. Hence, an active control is necessary in order to maintain the spacecraft as close as possible to the nominal orbit. In this respect, several control methods have been proposed [62] [63] [21] [149] [95] [90]. In particular, the authors of [51] and [53] highlight the advantage that nonlinear control turns out to have over linear control in terms of energy consumption as well as in terms of robustness with respect to the approximation that is intrinsic to the reference determination. This paper investigates the design and simulation of sampled-data nonlinear feedback control laws for tracking prescribed quasi-halo orbits around the L_2 translunar libration point.

Even interplanetary CubeSats could be exploited in order to fulfill such a technological task, as they allow less costly missions due to their reduced power, mass and volume as well as due to their less demanding launch requirements, provided that the related issue of fitting big science within a small package is properly dealt with [177].

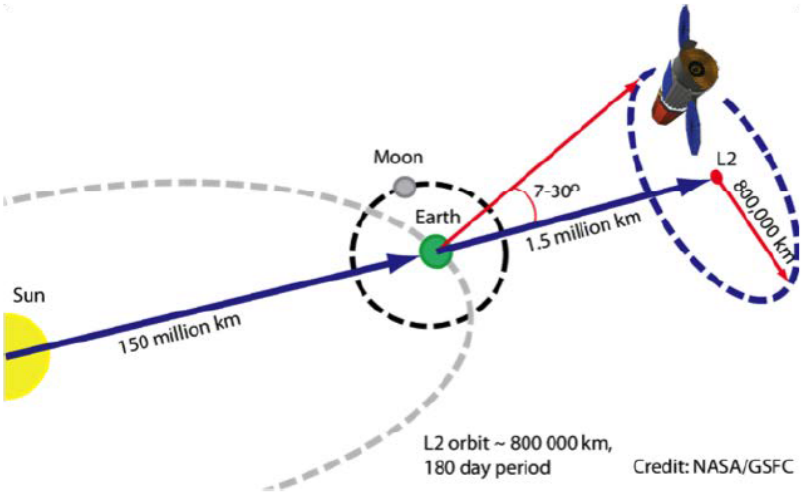


Figure 4.7: Halo orbit around the L_2 translunar libration point.

4.2.1 Quasi-Halo Orbit Following via Nonlinear Regulation

State-Space Representation

Let us first recall the following state-space representation of the dynamics of a satellite of mass $m_3 \ll m_1, m_2$ within the framework of the Restricted Three-Body Problem, as given in [53]:

$$\begin{aligned} \dot{x} &= f(x,z) + Gu = \begin{pmatrix} f_1(x) \\ f_2(x,z) \end{pmatrix} + Gu \\ y &= Cx. \end{aligned} \tag{4.19}$$

In particular, in the mathematical model above, $x(t) \in \mathbb{R}^6$ and $z(t) \in \mathbb{R}^4$ are such that

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} \xi_r(t) \\ \eta_r(t) \\ \zeta_r(t) \\ \dot{\xi}_r(t) \\ \dot{\eta}_r(t) \\ \dot{\zeta}_r(t) \end{pmatrix}, \; z(t) = \begin{pmatrix} \beta(t) \\ \beta^2(t) \\ \dot{\beta}(t) \\ \mu(1-\mu)\alpha(t) \end{pmatrix} = \begin{pmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \\ z_4(t) \end{pmatrix}, \tag{4.20}$$

where $\alpha(t)$ and $\beta(t)$ are periodic functions – introduced in [62] and used in [53] in order to define this model – whose average value is equal to zero, and $\mu = \frac{m_2}{m_1+m_2} \approx 0.012149$, with m_1 and m_2 (such that $m_1 > m_2$) the masses of the Earth and of the Moon, respectively. Moreover,

$$\begin{aligned} f_1(x) &= x_2, \\ f_2(x,z) &= -Mx_1 - 2Nx_2 - (2Mx_1 + 2Nx_2)z_1 - Mx_1 z_2 - \\ &\quad - Nx_1 z_3 - \frac{1-\mu}{|x_1 - d_{1,r}(z_4)|^3}(x_1 - d_{1,r}(z_4)) - \\ &\quad - \frac{\mu}{|x_1 - d_{2,r}(z_4)|^3}(x_1 - d_{2,r}(z_4)), \end{aligned}$$

where

$$M = R^T(\theta)\frac{\partial^2 R(\theta)}{\partial \theta^2} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, N = R^T(\theta)\frac{\partial R(\theta)}{\partial \theta} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$d_{1,r}(z_4(t)) = \begin{pmatrix} -\mu - \frac{z_4(t)}{1-\mu} \\ 0 \\ 0 \end{pmatrix}, \; d_{2,r}(z_4(t)) = \begin{pmatrix} 1 - \mu + \frac{z_4(t)}{\mu} \\ 0 \\ 0 \end{pmatrix},$$

$G = \begin{pmatrix} 0 \\ I \end{pmatrix}$, and $C = \begin{pmatrix} I & 0 \end{pmatrix}$. In particular,

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

is the rotation matrix describing the orientation of the rotating reference frame $O\xi_r\eta_r\zeta_r$ with respect to a given inertial reference frame $O\xi\eta\zeta$, $\theta(t)$ denotes the angle between ξ and ξ_r, and the inputs u are external controls acting on $\dot{x}_2$, i.e., on the accelerations along the axes of the rotating reference frame. For the sake of brevity, the index '(t)' is omitted, where possible, in continuous-time signals. Assuming that perturbations due to solar gravity and solar radiation pressure can be neglected, equation (4.19) describes the evolution of the considered continuous-time input-affine system.

Disturbance Model and Reference Trajectories

Let us now briefly analyze the disturbance term $z(t)$. According to [62], the periodic functions $\alpha(t)$ and $\beta(t)$ can be expressed as

$$\alpha(t) = \sum_{h=1}^{\infty} a_h e^h \cos(ht + \phi), \;\; \beta(t) = \sum_{k=1}^{\infty} b_k e^k \cos(kt + \phi),$$

where e is the eccentricity of the elliptical motion of the two-body (Earth-Moon) rotating system. Since in the present case $e \approx 0.0549$, we can consider either the first term of each power series

$$\alpha(t) \approx \alpha_1(t) = a_1 e \cos(t + \phi), \;\; \beta(t) \approx \beta_1(t) = b_1 e \cos(t + \phi),$$

or the first two terms

$$\begin{aligned} \alpha(t) &\approx \alpha_2(t) = a_1 e \cos(t + \phi) + a_2 e^2 (1 + \cos 2(t + \phi)), \\ \beta(t) &\approx \beta_2(t) = b_1 e \cos(t + \phi) + b_2 e^2 \cos 2(t + \phi), \end{aligned}$$

with $a_1 = -1, a_2 = \frac{1}{2}, b_1 = 2$, and $b_2 = \frac{5}{2}$. Then, according to (4.20), the perturbation $z(t)$ can be approximated either at the first order, $z(t) \approx z'(t)$, or even at the second order, $z(t) \approx z'(t) + z''(t)$, where

$$z'(t) = \begin{pmatrix} b_1 e \cos(t + \phi) \\ 0 \\ -b_1 e \sin(t + \phi) \\ \mu(1 - \mu) a_1 e \cos(t + \phi) \end{pmatrix}, \tag{4.21}$$

$$z''(t) = \begin{pmatrix} b_2 e^2 \cos(2(t+\phi)) \\ \frac{b_1^2}{2} e^2 (1 + \cos(2(t+\phi))) \\ -2 b_2 e^2 \sin(2(t+\phi)) \\ \mu(1-\mu) a_2 e^2 (1 + \cos(2(t+\phi))) \end{pmatrix}. \tag{4.22}$$

As regards instead the reference trajectory for our control task, in the present work we will take into account two possible quasi-halo orbits. The first one can be regarded as one of the equilibrium trajectories of the linearized dynamics of (4.19) assuming eccentricity $e = 0$ and making a proper modification so as to ensure satellite visibility from the Earth. Namely, such a trajectory is given by the first three components of the following unforced evolution:

$$\dot{x} = \left. \frac{\partial f(x,0)}{\partial x} \right|_{x=x_e} x, \tag{4.23}$$

where

$$x_e = \begin{pmatrix} L_2 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \approx \begin{pmatrix} 1.1556 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T$$

gives the coordinates, in the rotating reference frame, of one of the five equilibrium points of the nonlinear dynamics f in the absence of the disturbances $z(t)$ (i.e., $f(x_e, 0) = 0$), namely the coordinates of the so-called L_2 *translunar libration point*. It can be verified that periodic solutions for (4.23) exist. Given a suitable choice of $x(t_0)$, with $t_0 = 0$, we have

$$qho_1(t) := x_1(t) = \begin{pmatrix} -\frac{k(1-c_1+\Omega^2)}{2\Omega} \cos \Omega t \\ k \sin \Omega t \\ k \cos \Omega_z t \end{pmatrix}, \tag{4.24}$$

where

$$c_1 = \frac{1-\mu}{(L_2+\mu)^3} + \frac{\mu}{(L_2 - 1 + \mu)^3} \approx 3.18989,$$

$\Omega \approx 1.8636$ rad/s and $\Omega_z \approx 1.78$ rad/s are the natural frequencies, and k must be chosen so that it is small enough for the orbit to be as close as possible to the libration point and large enough for the satellite to stay visible from the Earth. As shown in [51], a good choice for k is 0.02. Moreover, to ensure visibility at all times, Ω_z must be replaced by Ω, thus obtaining a reference trajectory which is no more an equilibrium one for system (4.23).

Another quasi-halo orbit, which approximates a halo one with greater precision than orbit (4.24), can be obtained as a periodic equilibrium solution of the linearized dynamics of (4.19) assuming this time the presence of a non-zero perturbation $z(t)$ approximated at the first order. This means that such a quasi-halo orbit is given, for suitable initial conditions $x(t_0)$, by the

first three components of the following unforced evolution:

$$\dot{x} = \left.\frac{\partial f(x,0)}{\partial x}\right|_{x=x_e} x + P(x)|_{x=x_e} z', \tag{4.25}$$

where $P(x) = \frac{\partial f(x,z)}{\partial z}|_{z=0}$, which, computed at $x = x_e$, coincides with matrix P of equation (2.7) in [53]. It can be verified that the simplest periodic solutions of (4.25) are of the form

$$qho_2(t) := x_1(t) = \begin{pmatrix} -\frac{k(1-c_1+\Omega^2)}{2\Omega}\cos\Omega t + eh_1\cos(t+\phi) \\ k\sin\Omega t + eh_2\sin(t+\phi) \\ k\cos\Omega_z t \end{pmatrix}, \tag{4.26}$$

where Ω_z must be replaced by Ω, as previously explained. In particular,

$$h_1 = -\frac{1}{2c_1(1-c_1)}\{(2-c_1)[4L_2 - 2c_2\mu(1-\mu)] - 4L_2\}, \tag{4.27}$$

$$h_2 = -\frac{1}{2c_1(1-c_1)}\{-2[4L_2 - 2c_2\mu(1-\mu)] + 4L_2(1+c_1)\}, \tag{4.28}$$

and

$$c_2 = \frac{1}{(L_2+\mu)^3} - \frac{1}{(L_2-1+\mu)^3} \approx -210.88468.$$

Therefore, in the rotating reference frame, given the two quasi-halo orbits discussed above, the corresponding reference trajectories for the state of system (4.19), to be described around the translunar libration point L_2, are

$$x_{r,1}(t) = \begin{pmatrix} qho_1(t) + \begin{pmatrix} L_2 \\ 0 \\ 0 \end{pmatrix} \\ \dot{qho_1}(t) \end{pmatrix} \quad x_{r,2}(t) = \begin{pmatrix} qho_2(t) + \begin{pmatrix} L_2 \\ 0 \\ 0 \end{pmatrix} \\ \dot{qho_2}(t) \end{pmatrix}. \tag{4.29}$$

Nonlinear Regulation
with State and Disturbance Measurement

Following the methodological approach of nonlinear regulation with state and disturbance measurement discussed in [86], [53] introduces two interesting control laws designed with the aim of enforcing (i) asymptotically stable dynamics for the errors $e_i(t) = x(t) - x_{r,i}(t),\ i = 1, 2$, that is, for the errors between the real and the desired values of the state for the two cases of $x_{r,1}(t)$ and $x_{r,2}(t)$ in (4.29), and (ii) the simultaneous rejection of the periodic disturbances $z(t)$, considering either the first-order approximation or the second-order one.

Nonlinear Regulation for the Quasi-Halo Orbit qho_1

Assuming to consider only the first-order contribution of $z(t)$, the disturbances (4.21) can be rewritten in the following form:

$$z'(t) = \begin{pmatrix} b_1 e & 0 \\ 0 & 0 \\ 0 & -b_1 e \\ \mu(1-\mu)a_1 e & 0 \end{pmatrix} \begin{pmatrix} \cos(t+\phi) \\ \sin(t+\phi) \end{pmatrix} := T_1 w_{z,1}(t), \tag{4.30}$$

where $w_{z,1}(t)$ is a solution to

$$\dot{w}_{z,1} = S_{z,1} w_{z,1}, \text{ with } S_{z,1} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{4.31}$$

The reference trajectory, in this case given by $y_r(t) = qho_1(t)$, can be rewritten, according to (4.24), as $y_r(t) := -Q_1 w_{r,1}(t)$, where

$$w_{r,1}(t) = \begin{pmatrix} \cos \Omega t \\ \sin \Omega t \end{pmatrix} \text{ and } Q_1 = \begin{pmatrix} \frac{k(1-c_1+\Omega^2)}{2\Omega} & 0 \\ 0 & -k \\ -k & 0 \end{pmatrix}.$$

Defining

$$\omega(t) = \begin{pmatrix} \omega_1(t) \\ \omega_2(t) \end{pmatrix} := \begin{pmatrix} w_{z,1} \\ w_{r,1} \end{pmatrix},$$

we obtain the so-called *exosystem*, exhibiting linear dynamics in this case, namely

$$\dot{\omega} = S\omega, \quad \text{with } S = \begin{pmatrix} S_{z,1} & 0 \\ 0 & S_{r,1} \end{pmatrix} \text{ and } S_{r,1} = \begin{pmatrix} 0 & -\Omega \\ \Omega & 0 \end{pmatrix}, \tag{4.32}$$

while $S_{z,1}$ is given by (4.31). The corresponding regulation-based control law designed in [53] is

$$\begin{aligned} u(t) = &-f_2(\pi(\omega(t))) - P(\pi(\omega(t)))T_1\omega_1(t) - Q_1 S_{r,1}^2 \omega_2(t) + \\ &+ K(x(t) - \pi(\omega(t))) := \gamma_{qho_1}(\omega(t), \pi(\omega(t)), x(t)). \end{aligned} \tag{4.33}$$

where $\pi(\omega(t))$ is chosen as a C^k (with $k > 1$) mapping, defined in the neighborhood of $\omega = 0$ with $\pi(\omega(0)) = 0$, such that the regulator equations (8.14) in [86] are satisfied. Such a control law can then be plugged in the original nonlinear system (4.19), thus leading to the achievement of the desired objective in terms of asymptotic tracking and disturbance rejection.

Nonlinear Regulation for the Quasi-Halo Orbit qho_2

In this case, we choose to take into account also the presence of $z''(t)$. In particular, similarly to what has been done in (4.30), the disturbances (4.22) can be rewritten in the form $z''(t) := T_2 w_{z,2}(t)$, where $w_{z,2}(t)$ is a solution to

$$\dot{w}_{z,2} = S_{z,2} w_{z,2}, \text{ with } S_{z,2} = \begin{pmatrix} 0 & -2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The reference trajectory, which is given in this case by $y_r(t) = qho_2(t)$, can be rewritten, according to (4.26), as $y_r(t) := -Q_2 w_{r,2}(t)$, where

$$w_{r,2}(t) := \begin{pmatrix} \cos \omega t \\ \sin \omega t \\ \cos(t+\phi) \\ \sin(t+\phi) \end{pmatrix} = \begin{pmatrix} w_{r,1}(t) \\ w_{z,1}(t) \end{pmatrix}$$

is a solution to

$$\dot{w}_{r,2} = S_{r,2} w_{r,2} = \begin{pmatrix} S_{r,1} & 0 \\ 0 & S_{z,1} \end{pmatrix} w_{r,2},$$

$$Q_2 = \begin{pmatrix} \frac{k(1-c_1+\Omega^2)}{2\Omega} & 0 & -h_1 e & 0 \\ 0 & -k & 0 & -h_1 e \\ -k & 0 & 0 & 0 \end{pmatrix},$$

$S_{z,1}$ is given by (4.31), $S_{r,1}$ is given by (4.32) and the h_i's, for $i = 1, 2$, are given by (4.27) and (4.28), respectively. Defining

$$\omega(t) = \begin{pmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \end{pmatrix} := \begin{pmatrix} w_{r,1}(t) \\ w_{z,1}(t) \\ w_{z,2}(t) \end{pmatrix} = \begin{pmatrix} w_{r,2}(t) \\ w_{z,2}(t) \end{pmatrix},$$

the exosystem for the present case is therefore described by

$$\dot{\omega} = S\omega, \text{ where } S = \begin{pmatrix} S_{r,2} & 0 \\ 0 & S_{z,2} \end{pmatrix}.$$

The corresponding regulation-based control law designed in [53] is

$$\begin{aligned} u(t) = &-f_2(\pi(\omega(t)) - P(\pi(\omega(t))(T_1\omega_2(t) + T_2\omega_3(t)) - Q_2 S_{r,2}^2 \begin{pmatrix} \omega_1(t) \\ \omega_2(t) \end{pmatrix} + \\ &+ K(x(t) - \pi(\omega(t))) := \gamma_{qho_2}(\omega(t), \pi(\omega(t)), x(t)), \end{aligned} \tag{4.34}$$

which, plugged in the original nonlinear system (4.19), achieves the desired objective in terms of asymptotic tracking and disturbance rejection.

4.2.2 Sampled-Data Design by Emulation

Sampled-data controllers can be used with different purposes: from (i) the reproduction of the controller behaviours (i.e., controller discretization with possible continuous redesign) to (ii) the settlement of specific control objectives either at the sampling instants or in the inter-sampling (i.e., direct digital control) – see [70] for a detailed study of these control techniques in the linear context.

In the nonlinear context, the first approach is pursued by computing the continuous-time control law and maintaining it constant over the predefined sampling period: such a method of design is called *emulation* [123]. According

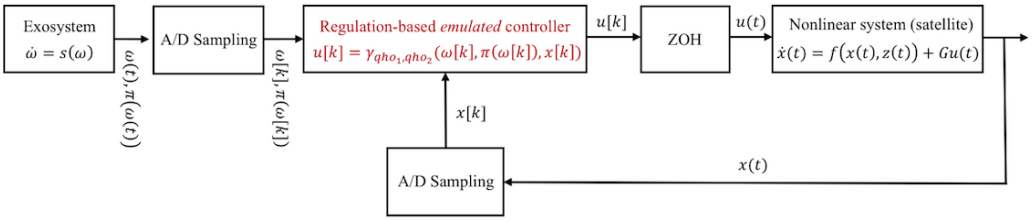


Figure 4.8: Sampled-data design by emulation.

to this last approach, we have therefore implemented a sampled-data emulated controller, which, receiving an input that consists of samples of $\omega(t), \pi(\omega(t))$ and $x(t)$, produces an output that should approximate the output of the continuous-time control laws (4.33) and (4.34), depending on the reference trajectory set by the exosystem. So, the control design is conducted as if the control system were continuous, then, after selecting a suitable sampling period, the controller γ_{qho_i} (with $i = 1, 2$) is digitized and the resulting sampled-data design is used in place of the continuous one (see Fig. 4.8 and the related block in red). The corresponding control samples $u[k]$ are then held in order to produce the continuous signal $u(t)$ which will act upon the physical process under control.

4.2.3 Remarks on Sampled-Data Design

The second approach discussed at the beginning of the previous section consists in designing a controller on the basis of the sampled-data equivalent model for the original continuous-time system. This will now be done with the purpose of improving the performances in terms of orbit following, with respect to the emulated controller. In this sense, roughly speaking, it can even be considered as a "redesign" procedure. Before introducing such an

improved controller, let us first recall some basic ideas about sampled-data design [123] [124] [157].

Let us consider the following continuous-time input-affine system, $\Sigma_C(f,h)$,

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + g(x(t))u(t) \\ y(t) &= h(x(t)), \end{aligned} \tag{4.35}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^q$, $y(t) \in \mathbb{R}^p$ with $p \leq n$ and $f(\cdot,\cdot)$ and $h(\cdot)$ are analytic in their arguments. Performing an m-th order multirate sampling means that the control signals are actuated faster than the rate which the data are measured at: $u(t)$ is consequently assumed to be constant over subintervals of amplitude $\bar{\delta} = \frac{\delta}{m}$,

$$u_i := u\left[k + \left(\frac{i-1}{m}\right)\right] = u(k\delta + \tau) \quad \text{for} \quad (i-1)\bar{\delta} \leq \tau < i\bar{\delta}, \quad i = 1, \ldots, m,$$

thus allowing m different control values on each interval of length δ, i.e., $u_i = u[k + (\frac{i-1}{m})]$ is active and constant for all $t \in [k\delta + (i-1)\bar{\delta}, k\delta + i\bar{\delta}[$ where $1 \leq i \leq m$. We describe the so-called *multirate sampled-data equivalent of order* m of Σ_C by the following discrete-time state-space representation, $\Sigma_D(F^{\delta}_{MR}, h)$,

$$\begin{aligned} x[k+1] &= F^{\delta}_{MR}(x[k], u_1, \ldots, u_m) \\ y[k] &= h(x[k]), \end{aligned}$$

where

$$\begin{aligned} F^{\delta}_{MR}(x, u_1, \ldots, u_m) &= e^{\bar{\delta}(f+gu_1)} \circ \cdots \circ e^{\bar{\delta}(f+gu_m)} x[k] = \\ &= e^{\mathscr{BCH}^m(\bar{\delta}(f+gu_1), \ldots, \bar{\delta}(f+gu_m))} x[k], \end{aligned} \tag{4.36}$$

if $x[0] = x(0)$ implies $x[k] = x(k\delta), k \geq 0$, and consequently $y[k] = y(k\delta)$, that is, if the discrete-time state behaviour $x[k]$ matches the continuous-time one $x(t)$ at the sampling instants. In particular, the expression $\mathscr{BCH}^m$ represents the Baker-Campbell-Hausdorff exponent associated with the non-commutative composition of m exponential operators, i.e., $e^{\mathscr{BCH}^m(f_1,\ldots,f_m)} := e^{f_1} \circ \ldots \circ e^{f_m}$.

In particular, F^{δ}_{MR} is nothing but the m-time composition of the single-rate sampled-data equivalent dynamics of (4.35). For any constant input $u[k]$ and δ small enough, the convergence of the series expansion for the exponential function in (4.36) is assumed to hold. Such a Σ_D is said to give an Exact Sampled Representation (ESR) of Σ_C. It becomes a Computable-Exact Sampled Representation (C-ESR) if a closed form exists for F^{δ}_{MR} and a Finitely Computable-Exact Sampled Representation (FC-ESR) when the resulting

series expansion in δ is of finite order. An exact calculus being difficult, it is in practice usual to consider an Approximate Sampled Representation (ASR) at a fixed order, corresponding to the truncation of the aforementioned series expansion. Once we have computed the sampled-data equivalent of Σ_C, we can re-state the control problem in terms of $\Sigma_D(F^{\delta}_{MR}, h)$ and choose the control law accordingly. In the literature such an approach is called *multirate direct digital design*.

The remarks above will prove useful for showing how a multirate sampled-data controller, which we will now define as the *improved* controller, sensibly enhances the performances in terms of reference tracking with respect to the previously discussed emulated controller.

4.2.4 Multirate Quasi-Halo Orbit Following

Starting from the input-affine satellite dynamics in (4.19), according to the definitions given in the previous section, we now compute the corresponding multirate ASR, choosing to truncate the Taylor series expansion of (4.36) at the first order. Hence, for the first semiperiod we have

$$\begin{aligned} x\left[k+\frac{1}{2}\right] &= x[k] + \frac{\delta}{2}\left\{ \begin{pmatrix} f_1(x[k]) \\ f_2(x[k], z[k]) \end{pmatrix} + Gu[k] \right\} = \\ &= \begin{pmatrix} x_1[k] \\ x_2[k] \end{pmatrix} + \frac{\delta}{2} \begin{pmatrix} f_1(x[k]) \\ f_2(x[k], z[k]) \end{pmatrix} + \frac{\delta}{2} \begin{pmatrix} 0_{3\times 3} \\ I_{3\times 3} \end{pmatrix} u[k]. \end{aligned}$$

In particular, as the active control variables are assumed to be u_4, u_5 and u_6, by resorting to a multirate approach of order 2 we can write the following:

$$u[k] = \begin{pmatrix} u_a[k] \\ u_b[k] \end{pmatrix} = \left(\begin{array}{c} u_1[k] \\ u_2[k] \\ u_3[k] \\ \hline u_4[k] \\ u_5[k] \\ u_6[k] \end{array}\right) = \left(\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline u_4[k] \\ u_5[k] \\ u_6[k] \end{array}\right),$$

with

$$\begin{aligned} u_4[k] = u_{41},\ u_4\left[k+\frac{1}{2}\right] = u_{42}, \\ u_5[k] = u_{51},\ u_5\left[k+\frac{1}{2}\right] = u_{52}, \\ u_6[k] = u_{61},\ u_6\left[k+\frac{1}{2}\right] = u_{62}. \end{aligned}$$

As a result of this choice, for the second semiperiod we have

$$x[k+1] = x\left[k+\frac{1}{2}\right] + \frac{\delta}{2}\begin{pmatrix} f_1(x[k+\frac{1}{2}]) \\ f_2(x[k+\frac{1}{2}], z[k+\frac{1}{2}]) \end{pmatrix} + \frac{\delta}{2} G u\left[k+\frac{1}{2}\right]. \quad (4.37)$$

Taking into account that $f_2(x[k+\frac{1}{2}], z[k+\frac{1}{2}])$ can be rewritten as follows

$$f_2\left(x\left[k+\frac{1}{2}\right], z\left[k+\frac{1}{2}\right]\right) = f_2'\left(x[k], z[k], z\left[k+\frac{1}{2}\right]\right) - \\ - 2N\frac{\delta}{2}u_b[k]\left(1 + z_1\left[k+\frac{1}{2}\right]\right)$$

if we separate a term depending on the state and on the disturbances from a term depending on the control inputs, and taking into account that, as a result,

$$\begin{pmatrix} f_1(x[k+\frac{1}{2}]) \\ f_2(x[k+\frac{1}{2}], z[k+\frac{1}{2}]) \end{pmatrix} = \begin{pmatrix} x_2[k] + \frac{\delta}{2} f_2(x[k], z[k]) \\ f_2'(x[k], z[k], z[k+\frac{1}{2}] \end{pmatrix} + \\ + \frac{\delta}{2}\begin{pmatrix} I \\ -2N(1 + z_1[k+\frac{1}{2}]) \end{pmatrix} u_b[k] = \\ := f'\left(x[k], z[k], z\left[k+\frac{1}{2}\right]\right) + g\left(z\left[k+\frac{1}{2}\right]\right) u_b[k],$$

eventually (4.37) becomes the following:

$$x[k+1] = x[k] + \frac{\delta}{2} f(x[k], z[k]) + \frac{\delta}{2} G u[k] + \frac{\delta}{2} f'\left(x[k], z[k], z\left[k+\frac{1}{2}\right]\right) + \\ + \frac{\delta}{2} g\left(z\left[k+\frac{1}{2}\right]\right) u_b[k] + \frac{\delta}{2} G u\left[k+\frac{1}{2}\right] = \\ = x[k] + \frac{\delta}{2}\left(f(x[k], z[k]) + f'\left(x[k], z[k], z\left[k+\frac{1}{2}\right]\right)\right) + \\ + \frac{\delta}{2}\left[\begin{pmatrix} 0 \\ I + g(z[k+\frac{1}{2}]) \end{pmatrix} \middle| \begin{pmatrix} 0 \\ I \end{pmatrix}\right] \begin{pmatrix} u[k] \\ u[k+\frac{1}{2}] \end{pmatrix}. \quad (4.38)$$

By rewriting (4.38) as follows

$$x[k+1] := x[k] + \frac{\delta}{2}\tilde{f}\left(x[k], z[k], z\left[k+\frac{1}{2}\right]\right) + \frac{\delta}{2}\tilde{g}\left(z\left[k+\frac{1}{2}\right]\right)\begin{pmatrix} u[k] \\ u[k+\frac{1}{2}] \end{pmatrix}, \quad (4.39)$$

we have now determined the expression of the multirate controlled dynamics of the satellite, where the multirate inputs are each (acting respectively on $\ddot{\xi}_r$, $\ddot{\eta}_r$, $\ddot{\zeta}_r$) of degree 2.

At this point, we choose to force such dynamics to be equal to

$$x[k+1] = P_D x[k] + B_D K \pi(\omega[k]), \tag{4.40}$$

with either $x_{r,1}(t)$ or $x_{r,2}(t)$ as the chosen reference trajectory. Note that (4.40) corresponds to an almost exact discretization of an asymptotically stable linear system

$$\begin{aligned} &\dot{x}(t) = \tilde{A}(t) + \tilde{B}v(t), \text{ with } v(t) = K(\pi(\omega(t)) - x(t)), \text{i.e.,} \\ &\dot{x} = (\tilde{A} - \tilde{B}K)x + \tilde{B}K\pi(\omega), \end{aligned}$$

with K chosen according to the rules of eigenvalue assignment in continuous time, $P_D = e^{(\tilde{A}-\tilde{B}K)\delta}$, $B_D = \int_0^\delta e^{(\tilde{A}-\tilde{B}K)s}\tilde{B}ds$, and $\tilde{A}$ and $\tilde{B}$, for instance, as given in [**?**]. Therefore, the asymptotically stable linear dynamics (4.40) can be forced via feedback onto the multirate satellite model (4.39), thus ensuring the achievement of the prescribed orbit-following objective. In other words, by performing map inversion, we choose the inputs $\begin{pmatrix} u[k] & u[k+\frac{1}{2}] \end{pmatrix}^T$ so that (4.39)≡(4.40), i.e., we choose the following multirate control law:

$$\begin{aligned} \begin{pmatrix} u[k] \\ u[k+\frac{1}{2}] \end{pmatrix} = \tilde{g}^{-1}\left(z\left[k+\frac{1}{2}\right]\right)\cdot\left[\frac{2}{\delta}\cdot\left(P_D x[k] + B_D K\pi(\omega[k]) - x[k]\right) - \right. \\ \left. - \tilde{f}\left(x[k], z[k], z\left[k+\frac{1}{2}\right]\right)\right], \end{aligned} \tag{4.41}$$

where $z[k+\frac{1}{2}]$ can be easily computed from the samples of either $z(t) \approx z'(t)$ in (4.21) or $z(t) \approx z'(t) + z''(t)$ in (4.22), depending on the chosen order of approximation of the disturbances. We refer to (4.41) as the *improved* controller.

4.2.5 Discussion

We now briefly discuss the main weaknesses and strengths of the presented approach. This is done by highlighting three important points that may be responsible of raising concern when resorting to a digital controller (e.g., the emulated or the improved one).

- First of all, the measurements the presented control laws resort to are entirely dependent on the availability of state feedback with the sampling frequency the digital control system is working at. Yet, in space missions it may happen either that state feedback is not available with a very high frequency, or worse, that the state of the controlled

system is not available for direct measurement at all. If the latter is the case, it is still possible to exploit the measurements of the outputs of the controlled system, along with the knowledge of its dynamics, in order to lead the overall closed-loop system to the desired control objective, as occurs, for instance, in nonlinear output regulation.

- We have designed the improved controller (4.41) on the basis of a multirate ASR at the first order of the original satellite continuous-time nonlinear dynamics (4.19). This has been done in order for the reader to have a good grasp of the methodological approach followed, without getting lost in the details of the calculations. Of course, relying on higher-order ASR's of (4.19) would yield better-performing multirate digital control laws, as an increasing order of approximation means that we are drawing closer and closer to an exact solution. Nonetheless, in comparison with (4.41), designing such further-improved controllers is just a matter of computational effort: the methodological approach is exactly the same so long as the necessary condition for performing the map inversion of (4.39) that leads to (4.41) is satisfied, that is, the sum of the multirate degrees r_i of each of the available inputs must be equal to the dimension n of the state vector, $\sum_{i=1}^{q} r_i = n$ (in the considered case, $n = q = 6, r_1 = r_2 = r_3 = 0, r_4 = r_5 = r_6 = 2$).

- Multirate digital design succeeds in forcing the state of the controlled system to exhibit the desired behaviour; in other words, the improved controller works as a *finite settling time* or *deadbeat* compensation – because the controlled system settles in a finite number of sampling periods – combined with trajectory tracking control – where the trajectory to be followed is one of the two quasi-halo orbits $x_{r,1}(t)$ and $x_{r.2}(t)$ in (4.29).

4.2.6 Simulations

The designed control laws have been tested in the MATLAB/Simulink environment. The following simulation results show that the improved controller sensibly enhances the performances with respect to the emulated controller. Both the emulated and the improved controllers have been applied to the original nonlinear satellite model (4.19).

For the sake of simplicity, in the simulations the disturbances modelling the perturbation due to eccentricity are not assumed to be measurable, which is why we have computed our control laws from the state and the output of a generator (i.e., the exosystem) modelling such disturbances with known frequency and amplitude.

Table 4.2: Values of $E(t)$ with $\delta = \frac{1}{9}$ s and $x_{r,1}(t)$ as the reference trajectory.

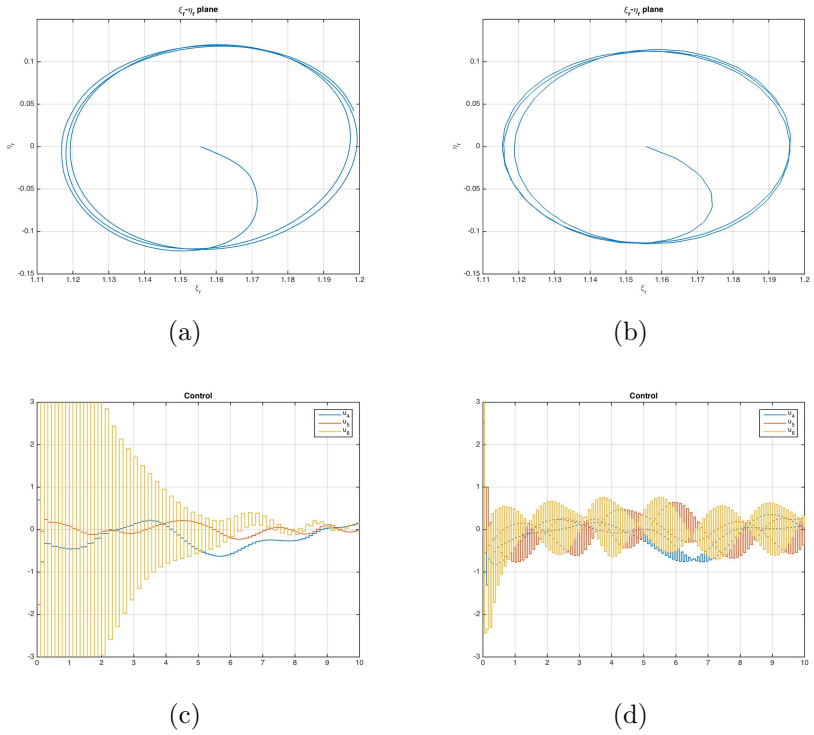
controller	time instant (s)					
	0.1	0.3	2.5	5	7.5	10
emulated	6.91	30.49	98.08	102.20	103.00	103.10
improved	4.14	5.05	6.40	7.26	8.85	9.70

Table 4.3: Values of $E(t)$ with $\delta = \frac{1}{9}$ s and $x_{r,2}(t)$ as the reference trajectory.

controller	time instant (s)						
	0.1	0.3	2.5	5	10	20	25
emulated	9.21	36.24	114.40	119.50	120.30	121.40	122.00
improved	5.61	6.49	7.93	9.25	11.73	16.13	18.34

Figs. 4.9-4.10 show the trajectory described by the spacecraft in the ξ_r-η_r plane, once it has been posed exactly in the L_2 translunar libration point with zero velocity (i.e., initial condition in L_2), for both control strategies (i.e., under the emulated and under the improved controller) and for both cases of reference choice $x_{r,1}(t)$ and $x_{r,2}(t)$, assuming $\delta = \frac{1}{9}$ s. Also, the evolution over time of the related piecewise-constant control inputs $u_b(t)$ is depicted.

In particular, below a certain threshold for the sampling period, the two control strategies exhibit very similar tracking performances (i.e., for $\delta < \tilde{\delta} = \frac{1}{9}$ s). However, as soon as the length of the sampling period rises above a specific threshold (i.e., $\delta \geq \tilde{\delta}$), the closed-loop system under the emulated controller goes entirely unstable. Instead, the improved controller preserves its stabilizing behaviour (see Fig. 4.11), even when the system's state is measured with lower sampling frequency, thus ensuring the desired result in terms of orbit following. Another difference is in the energy required for performing the tracking task: in this respect, the integral $\int_{t_0=0}^{t} |u(\tau)|^2 d\tau$ can be considered as an "energy-like" function $E(t)$ whose values under the emulated and the improved controller (considering the case when $\delta = \frac{1}{9}$ s) are collected in Tables 1 and 2. In particular, the time horizon for the simulations with the reference trajectory $x_{r,2}(t)$ has been set to 25 s in order to let the satellite's state produce a graphically meaningful description of the desired quasi-halo orbit (see Figs. 4.10-4.11). Indeed, as shown by the evolution over time of the integral introduced above, the improved controller entails a



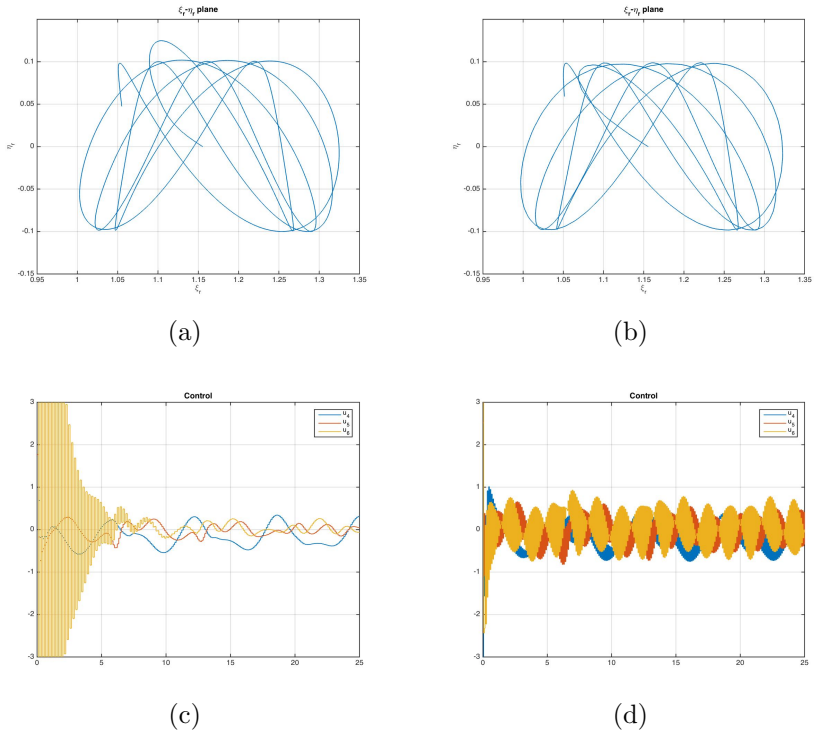
(a) (b)

(c) (d)

Figure 4.9: Subplots **(a)** and **(b)** show the trajectories described by the spacecraft in the ξ_r-η_r plane under the emulated and the improved controller, respectively, with $x(0) = x_e$, $\delta = \frac{1}{9}$ s and $x_{r,1}(t)$ as the reference trajectory. Subplots **(c)** and **(d)** show the piecewise-constant control inputs $u_b(t)$ plotted over time (measured in conventional seconds) for the emulated and the improved controller, respectively, with $\delta = \frac{1}{9}$ s and $x_{r,1}(t)$ as the reference trajectory.

less expensive control action with respect to the emulated one – not only at first, when moving from the initial position to the desired orbit, but on the whole, even during the orbit maintenance phase.

Moreover, as δ is raised above the threshold $\tilde{\delta}$, the energy contribution required by the improved controller for following the desired orbit does not experience a significant growth with respect to the values reported in Tables 1 and 2. This is another reason why a properly designed sampled-data multirate controller should be preferred to the emulated one.



(a) (b)

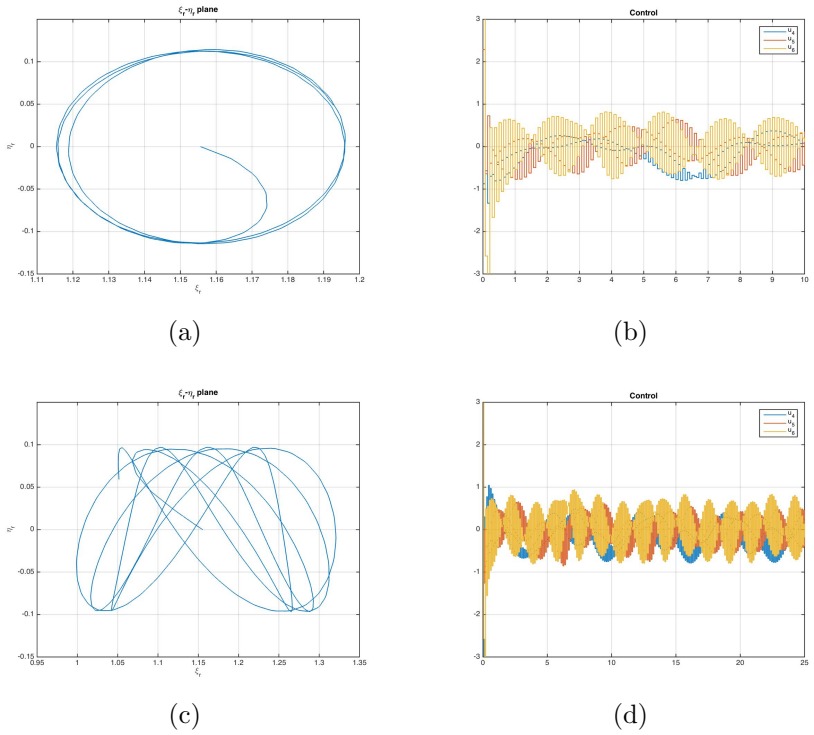
(c) (d)

Figure 4.10: Subplots **(a)** and **(b)** show the trajectories described by the spacecraft in the ξ_r-η_r plane under the emulated and the improved controller, respectively, with $x(0) = x_e$, $\delta = \frac{1}{9}$ s and $x_{r,2}(t)$ as the reference trajectory. Subplots **(c)** and **(d)** show the piecewise-constant control inputs $u_b(t)$ plotted over time (measured in conventional seconds) for the emulated and the improved controller, respectively, with $\delta = \frac{1}{9}$ s and $x_{r,2}(t)$ as the reference trajectory.

4.2.7 Conclusions

From the presented simulations, it is possible to conclude that, with respect to the emulation of a continuous-time regulation-based controller satisfying the tracking requirements, there are some advantages to relying instead on a properly designed sampled-data multirate stabilizing controller – namely, this last one yields better results in terms of the amplitude of the control variables and allows to use larger sampling periods.

We must point out that all of the proposed sampled-data control laws are designed on the basis of approximations, more or less accurate, of the nonlinear satellite model. This means that there is some unknown nonlinear contribution from the system whose effect is equivalent to an unknown disturbance term. In



(a) (b)

(c) (d)

Figure 4.11: Subplots **(a)** and **(c)** show the trajectories described by the spacecraft in the ξ_r-η_r plane under the improved controller, with $x(0) = x_e$, $\delta = \frac{1}{7}$ s, $x_{r,1}(t)$ as the reference trajectory in **(a)** and $x_{r,2}(t)$ as the reference trajectory in **(c)**. Subplots **(b)** and **(d)** show the piecewise-constant control inputs $u_b(t)$ plotted over time (measured in conventional seconds) for the improved controller, with $\delta = \frac{1}{7}$ s, $x_{r,1}(t)$ as the reference trajectory in **(b)** and $x_{r,2}(t)$ as the reference trajectory in **(d)**.

this respect, the simulation results suggest that a certain degree of robustness is guaranteed by the proposed control schemes. Further studies will address the investigation of such robustness aspects.

In general, the closer the initial state is to the reference orbit that has to be described, the more satisfactory the performances yielded by the proposed sampled-data control laws will be. Moreover, a sampled-data control law exhibits its major drawback during the phase when the satellite is reaching the desired orbit, especially if the same control strategy is employed both for orbit acquisition and maintenance. Therefore, a reasonable solution to the overall control problem could consist in a "mixed" strategy, using first a continuous-time nonlinear regulator while performing orbit acquisition, and then a sampled-data multirate controller for ensuring orbit maintenance.

Conclusions and Future Work

Across different fields, scientists are making a dramatic progress and pushing network analysis and design to its limits. In engineering, in particular, researchers study how to coordinate individual physical devices into a coherent whole with the aim of performing a common task. This has given rise to an exciting research field, i.e., the field of networked multi-agent dynamical systems, with several applications requiring high adaptivity and scalability. Such objectives are typically pursued by designing distributed algorithms that rely only on local interaction in order to achieve global group behaviours.

This PhD thesis presents and discusses the activities carried out by the PhD candidate (i) from the point of view of the applications of networked multi-agent systems in the context of Italian and European research projects (see Chapter 1), and (ii) from the point of view of the innovative methodological results obtained during the three-year PhD programme (see Chapters 2, 3 and 4).

In particular, in the PLATINO project, focused on the telecommunications domain (see Chapter 1), a Multi-Agent Reinforcement Learning based approach to Quality of Experience (QoE) Control has been proposed, enabling a dynamic Class of Service selction aimed at reducing the error between the personalized Perceived QoE and the personalized Target QoE levels by properly driving the control procedures that handle the underlying networks. The related algorithm is characterized by a very good degree of scalability, but assumes the time-invariance of the Target QoE level. So, further studies are being conducted based on concept drift in telecommunication systems in order to address the case of a time-varying Target QoE. Moreover, further research is currently being carried out, based on a combinatorial multi-armed bandit approach to cooperative online learning, in order to overcome the centralized paradigm and consequently develop a solution where QoE Control functions are fully distributed into the agents.

In the T-NOVA project, still focused on the telecommunications domain (see Chapter 1), a distributed non-cooperative load balancing algorithm based on mean field game theory is proposed: it is proved to asymptotically

converge to the exact Wardrop equilibrium, in contrast with the existing discrete-time algorithms which converge to approximate Wardrop equilibria only. Future work will be aimed at analyzing the effects of time delays and of communications constraints which may hinder the migration among network paths. Moreover, a Wardrop load balancing algorithm was implemented in a real SDN network, achieving convergence to an arbitrarily small neighborhood of a Wardrop equilibrium (thus yielding set stability). In this respect, a proof-of-concept implementation is provided and future work will be aimed at validating the algorithm on larger use cases.

In the ATENA project, focused on the domain of power networks (see Chapter 1), a protection scheme for power transmission grids making use of energy storage systems in support of primary frequency regulation services for improving reaction to closed-loop dynamic load altering attacks has been proposed. Such a protection scheme allows to solve an optimization problem subject to a Lyapunov stability constraint for the autonomous representation of the power system obtained after linearization and application of the attack and frequency control laws. The reported results show that the proposed iterative algorithm allows to determine a solution to the problem of optimizing the number and location of energy storage systems while ensuring grid stability. Nonetheless, the deployability of the obtained solution depends on the availability on the market of suitably-sized energy storage systems. Future work will be aimed at tackling the same problem by means of alternative methods for finding a sparse solution, on the one hand, and reformulating the problem in order to exploit energy storage also as a support to secondary regulation services, on the other hand.

After extensively discussing the above-mentioned project-related results, the focus of the thesis is shifted from the applications to the methodological framework, first of all by providing an overview on the control-theoretic methods for the distributed coordination of multi-agent systems (see Chapter 2). Such an overview ranges from first- and second-order mathematical models to networked Lagrangian systems and sampled-data control, thus paving the way for the research results presented in the subsequent Chapter.

Chapter 3 introduces multi-consensus and provides insight into how the network topology determines the properties of the dynamical evolution of a multi-agent system, ultimately showing that at steady-state the agents group into properly organized clusters whose composition is dictated by a suitable graph partition which is proved to be almost equitable. Future work will be aimed at exploiting the notion of almost equitable partition in order to alter the network topology, by enforcing a control action onto properly chosen lead nodes, with the aim of achieving desired multi-consensus conditions.

Eventually, Chapter 4 discusses the results achieved with respect to

sampled-data design in the aerospace domain. In particular, the exact computation of the state predictor is shown for a nonlinear system that admits a finite-order sampled-data equivalent model under feedback. In this respect, an exact delay-compensating piecewise-continuous steering controller is designed and simulated on the mathematical model of a Planar Vertical Take-Off and Landing aircraft with delays in the inputs. Future work will be aimed at extending the obtained results to the case when the exact values of the delays are unknown and at achieving good performances in terms of robustness with respect to disturbances. Also, the problem of spacecraft quasi-halo orbit following around the L_2 translunar libration point via continuous-time nonlinear regulation is investigated. In this respect, a multirate sampled-data stabilizing control law is compared with the sampled-data controller designed by emulation of the continuous-time nonlinear regulator, showing that the former yields sensibly better tracking performances than the latter. Since the discussed sampled-data control laws are designed on the basis of approximations of nonlinear models, the simulation results suggest that a certain degree of robustness is guaranteed. Further studies will address the investigation of such robustness aspects.

All in all, this PhD thesis gives an overview of the challenges and research opportunities arising in the field of networked dynamical systems, together with some interesting accomplishments, both from the point of view of practical applications (as regards the simulations and implementations devised in the context of the PLATINO, T-NOVA and ATENA projects) and from the methodological point of view (see Chapters 2 and 3). A concise digression is devoted to sampled-data control with aerospace applications (see Chapter 4): its contents account for a brief follow-up to the work carried out by the PhD candidate, namely in the period ranging from his M.Sc. thesis and the end of the first year of the PhD programme.

It is worth recalling that, while carrying out this research, the PhD candidate benefited greatly not just from *Sapienza, Università di Roma*, but also from the collaboration and exchange with the *Laboratoire des Signaux et Systèmes* (L2S) at *Centrale Supélec, Université Paris-Saclay* within the context of the *co-tutelle* with the *Ecole Doctorale de Sciences et Technologies de l'Information et de la Communication* (ED STIC).

Bibliography

[1] H. Ackermann, S. Fischer, M. Hoefer, M. Schngen, "Distributed algorithms for QoS load balancing," *Distrib. Comput.*, vol. 23, no. 56, pp. 321330, Dec. 2010.

[2] D. Aeyels, F. De Smet, "Emergence and evolution of multiple clusters of attracting agents," *Physica D: Nonlinear Phenomena*, vol. 239, no. 12, pp. 1026-1037, 2010.

[3] R. Agaev, P. Chebotarev, "On the spectra of nonsymmetric Laplacian matrices," *Linear Algebra and Its Applications*, vol. 399, pp. 157-168, 2005.

[4] C.O. Aguilar, B. Gharesifard, "Almost equitable partitions and new necessary conditions for network controllability," *Automatica*, vol. 80, pp. 25-31, 2017.

[5] S. Amini, F. Pasqualetti, and H. Mohsenian-Rad, "Dynamic Load Altering Attacks against Power System Stability: Attack Models and Protection Designs," in *IEEE Trans. on Smart Grid*, in press, 2016.

[6] J. Anselmi, U. Ayesta, A. Wierman, "Competition yields efficiency in load balancing games," *Perform. Eval.*, vol. 68, no. 11, pp. 9861001, Nov. 2011.

[7] E. Anshelevich, D. Kempe, J. Kleinberg, "Stability of Load Balancing Algorithms in Dynamic Adversarial Systems," *SIAM Journal on Computing*, vol. 37, no. 5, pp. 1656-1673, 2008.

[8] Z. Artstein, "Linear systems with delayed controls: A reduction," *Automatic Control, IEEE Transactions on*, vol. 27, no. 4, pp. 869-879, 1982.

[9] S. Barbarossa, S. Sardellitti, A. Farina, "On sparse controllability of graph signals," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (CASSP)*, pp. 4104-4108, 2016.

[10] M. Barbetta, M.C. Falvo, C. D'Adamo, L. D'Orazio, E. Duca, "Energy storage systems and distribution grids: A real case study in Italy," *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, doi: 10.1109/EEEIC.2016.7555881.

[11] D. Barth, O. Bournez, O. Boussaton, J. Cohen, "Distributed learning of wardrop equilibria," *Lecture Notes in Computer Science*, vol. 5204, 2008, pp. 1932.

[12] S. Battilotti, C. Gori Giorgi, S. Monaco, M. Panfili, A. Pietrabissa, L. Ricciardi Celsi, V. Suraci, "A multiagent reinforcement learning based approach to quality of experience control in Future Internet networks," in *Proc. of the 34th Chinese Control Conference (CCC2015)*, Hangzhou, China, pp. 6495-6500, July 28-30, 2015.

[13] D. Bauso, X. Zhang, A. Papachristodoulou, "Density Flow in Dynamical Networks via Mean-Field Games," *IEEE Transactions on Automatic Control*, 62 (3), art. no. 7499816, pp. 1342-1355, 2017.

[14] M. Beckmann, C.B. McGuire, C. B. Winsten, *Studies in the economics of transportation*. New Haven: Yale University Press, 1959.

[15] F. Benmansour, G. Carlier, G. Peyré, F. Santambrogio, "Numerical approximation of continuous traffic congestion equilibria," *Networks and Heterogeneous Media*, 4 (3), pp. 605-623, 2009.

[16] D.P. Bertsekas, J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., 1989.

[17] V.D. Blondel, J.M. Hendrickx, J.N. Tsitsiklis, "Continuous-time average-preserving opinion dynamics with opinion-dependent communications," *SIAM Journal on Control and Optimization*, vol. 48, no. 8, pp. 5214-5240, 2010.

[18] V. S. Borkar, P.R. Kumar, "Dynamic Cesaro-Wardrop equilibration in networks," in *IEEE Trans. Automat. Contr*, vol. 48, no. 3, pp. 382396, Mar. 2003.

[19] W.F. Boyer, S.A. McBride, "Study of security attributes of smart grid systems – current cyber secrity issues," *USDOE*, 2009.

[20] S. Boyd, L. Vandenberghe, *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[21] J.V. Breakwell, A.A. Kamel, M.J. Ratner, "Station-keeping for a translunar communication station," *Celestial Mechanics*, vol. 10, no. 3, pp. 357-373, 1974.

[22] R.W. Brockett, "Asymptotic stability and feedback stabilization," in *Dierential Geometric Control Theory*, Birkhauser, Boston, USA. pp. 181-191, 1983.

[23] C. Bruni, F. Delli Priscoli, G. Koch, A. Palo, A. Pietrabissa, "Quality of experience provision in the Future Internet," *IEEE Syst. J.*, vol. 10, no. 1, pp. 302-312, 2016.

[24] A. Buscarino, L. Fortuna, M. Frasca, S. Frisenna, "Interaction between synchronization and motion in a system of mobile agents," *Chaos*, 26, art. no. 116302, 2016.

[25] A. Buscarino, M. Frasca, M. Branciforte, L. Fortuna, J.C. Sprott, "Synchronization of two Rssler systems with switching coupling," *Nonlinear Dynamics*, vol. 88, pp. 673-683, 2017.

[26] L. Busoniu, R. Babuska, B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *Syst. Man, Cybern. Part C Appl. Rev. IEEE Trans.*, vol. 43, pp. 309-319, 2001.

[27] S. Canale, F. Facchinei, R. Gambuti, L. Palagi, V. Suraci, "User profile based quality of experience," *Proceedings of the 18th International Conference on Circuits, Systems, Communications and Computers (CSCC 2014)*, Santorini Island, Greece, Advances in Information Science and Applications Volume II, 2014.

[28] S. Canale, A. Di Giorgio, F. Lisi, M. Panfili, L. Ricciardi Celsi, V. Suraci, F. Delli Priscoli, "A Future Internet oriented user centric extended intelligent transportation system," in *Proc. of the 24th Mediterranean Conference on Control and Automation (MED)*, Athens, pp. 1133-1139, 2016.

[29] D.M. Cardoso, C. Delorme, P. Rama, , "Laplacian eigenvectors and eigenvalues and almost equitable partitions," *European Journal of Combinatorics*, 28 (3), pp. 665-673, 2007.

[30] G. Carlucci, L. De Cicco, S. Holmer, S. Mascolo, "Congestion Control for Web Real-Time Communication," *IEEE/ACM Transactions on Networking*, in press, 2017.

[31] M. Castrucci, F. Delli Priscoli, A. Pietrabissa, V. Suraci, "A Cognitive Future Internet Architecture," *Lect. Notes Comput. Sci.*, vol. 6656, pp. 91-102, 2011.

[32] J.S. Caughman, J.J.P. Veerman, "Kernels of Directed Graph Laplacians," *Electronic Journal of Combinatorics*, 13 (1 R), pp. 1-8, 2006.

[33] S. Chatterjee, E. Seneta, "Towards consensus: some convergence theorems on repeated averaging," *Journal of Applied Probability*, vol. 14, pp. 89-97, 1977.

[34] Y. Chen, J. Lu, F. Han, X. Yu, "On the cluster consensus of discrete-time multi-agent systems," *Systems and Control Letters*, vol. 60, no. 7, pp. 517-523, 2011.

[35] S.W. Chiou, "Optimization of a nonlinear area traffic control system with elastic demand," *Automatica*, vol. 46, no. 10, pp. 16261635, Oct. 2010.

[36] N. Chopra, D.M. Stipanovic, M.W. Spong, "On synchronization and collision avoidance for mechanical systems," in *Proceedings of the American Control Conference*, pp. 3713 - 3718, June 2008.

[37] S.J. Chung, 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.

[38] F. Cimorelli, F. Delli Priscoli, A. Pietrabissa, L. Ricciardi Celsi, V. Suraci, L. Zuccaro, "A Distributed Load Balancing Algorithm for the Control Plane in Software Defined Networking," in *Proceedings of the 24th Mediterranean Conference on Control and Automation (MED 2016)*, pp. 1033-1040, June 21-24, 2016, Athens, Greece, DOI: 10.1109/MED.2016.7535946.

[39] G. Como, K. Savla, D. Acemoglu, M.A. Dahleh, E. Frazzoli, "Robust distributed routing in dynamical networks – Part I: Locally responsive policies and weak resilience," *IEEE Transactions on Automatic Control*, 58 (2), art. no. 6247471, pp. 317-332, 2013.

[40] J.R. Correa, N.E. Stier-Moses, "Wardrop Equilibria," in *Wiley Encyclopedia of Operations Research and Management Science*, J.J. Cochran, L.A. Cox, P. Keskinocak, J.P. Kharoufeh, and J.C. Smith, Eds. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2011.

[41] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726 - 737, 2008.

[42] L. De Cicco, S. Mascolo, A Mathematical Model of the Skype VoIP Congestion Control Algorithm," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 790-795, 2010.

[43] L. De Cicco, S. Mascolo, S.I. Niculescu, "Robust stability analysis of Smith predictor-based congestion control algorithms for computer networks," *Automatica*, vol. 47, no. 8, pp. 1685-1692, 2011.

[44] L. De Cicco, G. Cofano, S. Mascolo, "Local SIP Overload Control: Controller Design and Optimization by Extremum Seeking," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 3, art. no. 7035079, pp. 267-277, 2015.

[45] M.H. De Groot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, pp. 118-121, 1974.

[46] P. DeLellis, F. Garofalo, F. Lo Iudice, "Partial pinning control of complex networks," in *Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC)*, ARIA Resort & Casino, December 12-14, 2016, Las Vegas, USA.

[47] F. Delli Priscoli, V. Suraci, A. Pietrabissa, M. Iannone, "Modelling quality of experience in Future Internet networks," in *Proc. of the Future Network & Mobile Summit (FutureNetw)*, 2012.

[48] F. Delli Priscoli, A. Fiaschetti, V. Suraci, "The SHIELD Framework: how to control Security, Privacy and Dependability in Complex Systems," *2012 IEEE Workshop on Complexity in Engineering*, Doi: 2-s2.0-84866534292.

[49] F. Delli Priscoli, A. Di Giorgio, F. Lisi, S. Monaco, A. Pietrabissa, L. Ricciardi Celsi, V. Suraci, "Multi-agent quality of experience control," *International Journal of Control, Automation and Systems*, 15 (2), pp. 892-904, 2017. DOI: 10.1007/s12555-015-0465-5.

[50] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 19, no. 4, pp. 112, 1989.

[51] P. Di Giamberardino, S. Monaco, "Nonlinear regulation in halo orbits control design," in *Proc. of the 31st IEEE Conference on Decision and Control*, pp. 536-541, 1992.

[52] P.Di Giamberardino, S. Monaco, M.D. Normand-Cyrot, "Digital control through finite feedback discretizability," in *Robotics and Automation (ICRA), Proceedings of the IEEE International Conference on*, Minneapolis, Minnesota, vol. 4, pp. 3141-3146, 1996.

[53] P. Di Giamberardino, S. Monaco, "On halo orbits spacecraft stabilization," *Acta Astronautica*, vol. 38, no. 12, pp. 903-925, 1996.

[54] P. Di Giamberardino, M. Djemai, S. Monaco, M.D. Normand-Cyrot, "Exact steering and stabilization of a PVTOL aircraft," *Decision and Control (CDC), Proceedings of the 36th IEEE Conference on*, San Diego, California, vol. 3, pp. 2049-2054, 1997.

[55] A. Di Giorgio, L. Pimpinella, F. Liberati, "A model predictive control approach to the load shifting problem in a household equipped with an energy storage unit," in *Proc. of the 20th Mediterranean Conference on Control and Automation MED12*,1491-1498, DOI: 10.1109/MED.2012.6265850.

[56] A. Di Giorgio, F. Liberati, A. Lanna, "Real Time Optimal Power Flow integrating Large Scale Storage Devices and Wind Generation," *23rd Mediterranean Conference on Control and Automation, MED15*, pp.480-486, 2015, DOI: 10.1109/MED.2015.7158794.

[57] A. Di Giorgio, F. Liberati, A. Lanna, A. Pietrabissa, F. Delli Priscoli, "Model Predictive Control of Energy Storage Systems for Power Tracking and Shaving in Distribution Grids," *IEEE Transactions on Sustainable Energy*, IEEE, 2016, ISSN: 1949-3029.

[58] C. Estan, S. Savage, and G. Varghese, "Automatically inferring patterns of resource consumption in network traffic," *Proc. 2003 Conf. Appl. Technol. Archit. Protoc. Comput. Commun. – SIGCOMM 03*, pp. 137-148, 2003.

[59] E. Even-Dar, A. Kesselman, Y. Mansour, *Convergence time to Nash equilibria*. Springer-Verlag, Berlin Heidelberg, 2003.

[60] E. Even-Dar, Y. Mansour, "Fast convergence of selfish rerouting," in *16th annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics*, 2005, pp. 772781.

[61] M.C. Falvo, L. Martirano, D. Sbordone, E. Bocci,, "Technologies for smart grids: A brief review," *2013 12th International Conference on*

Environment and Electrical Engineering, Wroclaw, 2013, pp. 369-375. doi: 10.1109/EEEIC.2013.6549544.

[62] R.W. Farquhar, "The control and use of libration-point satellites," *NASA TR R-346*, sept. 1970.

[63] R.W. Farquhar, "The utilization of halo orbits in advanced lunar operations," *NASA TN D-6365*, July 1971.

[64] M. Fiedler, T. Hossfeld, P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Network*, vol. 24, no. 2, pp. 36-41, 2010.

[65] S. Fischer, N. Kammenhuber, A. Feldmann, "REPLEX: dynamic traffic engineering based on wardrop routing policies," in *Proceedings of the 2nd Conference on Future Networking Technologies (CoNext)*, 2006, pp. 617.

[66] S. Fischer, L. Olbrich, and B. Vocking, "Approximating Wardrop equilibria with finitely many agents," *Distrib. Comput.*, vol. 21, no. 2, pp. 129139, Mar. 2008.

[67] S. Fischer, B. Vocking, "Adaptive routing with stale information," *Theor. Comput. Sci.*, vol. 410, no. 36, pp. 33573371, Aug. 2009.

[68] S. Fischer, H. Racke, B. Vocking, "Fast Convergence to Wardrop Equilibria by Adaptive Sampling Methods," *SIAM J. Comput.*, vol. 39, no. 8, pp. 37003735, Jan. 2010.

[69] R. Francisco, F. Mazenc, S. Mondié, "Global asymptotic stabilization of a PVTOL aircraft model with delay in the input," in *Applications of Time Delay Systems*, Lecture Notes in Control and Information Sciences, vol. 352, pp. 343-356, Springer, Berlin/Heidelberg, 2007.

[70] G.F. Franklin, J.D. Powell, A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Pearson, 2010.

[71] Y. Gai, B. Krishnamachari, Q. Zhao, "Combinatorial network optimization with unknown variables: Mult-iarmed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 5, pp. 1466-1478, 2012.

[72] Y. Gao, L. Wang, "Consensus of multiple double-integrator agents with intermittent measurement," *International Journal of Robust Nonlinear Control*, vol. 20, no. 10, pp. 1140 - 1155, 2009.

[73] Y. Gao, L. Wang, G. Xie, B. Wu, "Consensus of multi-agent systems based on sampled-data control," *International Journal of Control*, vol. 82, no. 12, pp. 2193 - 2205, 2009.

[74] D. Gayme, U. Topcu., "Optimal power flow with large-scale storage integration," in *Power Systems, IEEE Transactions on*, 28(2):709717, May 2013.

[75] M. Ghanes, J. De Leon, J.P. Barbot, "Observer design for nonlinear systems under unknown time-varying delays," *IEEE Transactions on Automatic Control*, vol. 58, no. 6, art. no. 6340311, pp. 1529-1534, 2013.

[76] J.D.D. Glover, M. S. Sarma, T. J. Overbye, *Power System Analysis and Design*, 5th ed. Cengage Learning, 2009.

[77] C. Godsil, G.F. Royle, *Algebraic graph theory*. Springer Science & Business Media, 2013.

[78] D. Grosu, A.T. Chronopoulos, "Algorithmic Mechanism Design for Load Balancing in Distributed Systems," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 34, no. 1, pp. 7784, Feb. 2004.

[79] D. Grosu and A. T. Chronopoulos, "Noncooperative load balancing in distributed systems," *J. Parallel Distrib. Comput*, vol. 65, no. 9, pp. 10221034, Sep. 2005.

[80] D. Grosu, A.T. Chronopoulos, M.Y. Leung, "Cooperative load balancing in distributed systems," *Concurr. Comput. Pract. Exp.*, vol. 20, no. 16, pp. 19531976, Nov. 2008.

[81] J. Hauser, S. Sastry, G. Meyer, "Nonlinear control design for slightly non minimum-phase systems: Application to V/STOL aircraft," *Automatica*, vol. 28, no. 4, pp. 665-679, 1992.

[82] T. Hayakawa, T. Matsuzawa, S. Hara, "Formation control of multi-agent systems with sampled information," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 4333 - 4338, December 2006.

[83] H. Hermes, A. Lundell, D. Sullivan, "Nilpotent bases for distributions and control systems," *Journal of Differential Equations*, vol. 55, no. 3, pp. 385-400, 1984.

[84] Y.N. Hu, W.D. Wang, X.Y. Gong, X.R. Que, S.D. Cheng, "On the placement of controllers in software-defined networks," *J. China Univ. Posts Telecommun.*, vol. 19, no. S2, pp. 92171, Oct. 2012.

[85] J. Huang, "The Consensus for Discrete-Time Linear Multi-Agent Systems Under Directed Switching Networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 4086-4092, 2017.

[86] A. Isidori, *Nonlinear Control Systems*. Springer Verlag, 1995.

[87] T. Jaakkola, M.I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Computation*, vol. 6, pp. 1185-1201, 1994.

[88] A. Jadbabaie, J. Lin, A.S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, 48 (6): 988-1001, 2003.

[89] S. Jelassi, G. Rubino, H. Melvin, H. Youssef, G. Pujolle, "Quality of experience of VoIP service: a survey of assessment approaches and open issues," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 12, pp. 491-513, 2012.

[90] J. Jia, K. Ma, Y. Yao, "Nonlinear geometry control of spacecraft for halo orbit maintenance," in *Proc. of the 30th Chinese Control Conference*, pp. 721-726, 2011.

[91] Q. Jiang, H. Xi, B. Yin, "Dynamic file grouping for load balancing in streaming media clustered server systems," *International Journal of Control, Automation and Systems*, vol. 7, no. 4, pp. 630-637, 2009.

[92] J. Jin, Y. Zheng, "Consensus of multi-agent system under directed network: A matrix analysis approach," *Proceedings of the 2009 IEEE International Conference on Control and Automation, ICCA 2009*, pp. 280-284, 2009.

[93] H. Kameda, J. Li, C. Kim, Y. Zhang, *Optimal Load Balancing in Distributed Computer Systems*. London: Springer London, 1997.

[94] Kandula et al., "The nature of data center traffic: measurements & analysis," in *Proceedings of IMC 2009*, ACM, pp. 202-208, 2009.

[95] K. Kemih, A. Kemiha, M. Ghanes, "Chaotic attitude control of satellite using impulsive control," *Chaos, Solitons and Fractals*, vol. 42, no. 2, pp. 735-744, 2009.

[96] H.K. Khalil, *Nonlinear Systems*. Prentice-Hall, Inc., 2002.

[97] S.U. Khan, I. Ahmad, "A Cooperative Game Theoretical Technique for Joint Optimization of Energy Consumption and Response Time in Computational Grids," *IEEE Trans. Parallel Distrib. Syst*, vol. 20, no. 3, pp. 346360, Mar. 2009.

[98] M. Krstic, *Delay Compensation for Nonlinear, Adaptive and PDE Systems*. Birkhauser, Boston, 2009.

[99] G. Lafferriere, A. Williams, J. Caughman, J.J.P. Veerman, "Decentralized control of vehicle formations," *Systems and Control Letters*, vol. 54, no. 9, pp. 899-910, 2005.

[100] S. Lange et al., "Heuristic Approaches to the Controller Placement Problem in Large Scale SDN Networks," *IEEE Transactions on Network and Service Management*, Year: 2015, Volume: 12, Issue: 1 Pages: 4 - 17, DOI: 10.1109/TNSM.2015.2402432.

[101] Y.O. Lee, B. Rengarajany, A.L. Narasimha Reddy, "Reaching approximate Wardrop equilibrium at reduced costs of link state updates," *3rd International Conference on Communication Systems and Networks, COMSNETS 2011*, art. no. 5716481, 2011.

[102] Y. Levron, J.M. Guerrero, Y. Becki, "Optimal power flow in microgrids with energy storage," *Power Systems, IEEE Transactions on*, 28(3): 32263234, Aug 2013.

[103] M. M. L. Littman, "Friend-or-foe Q-learning in general-sum Games," *ICML*, vol. 1, pp. 322-328, 2001.

[104] T. Li, J.F. Zhang, "Consensus Conditions of Multi-Agent Systems With Time-Varying Topologies and Stochastic Communication Noises," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2043-2057, 2010.

[105] J. Li, Z.H. Guan, G. Chen, "Multi-consensus of nonlinearly networked multi-agent systems," *Asian Journal of Control*, vol. 17, no. 1, pp. 157-164, 2015.

[106] P. Lin, Y. Jia, "Distributed rotating formation control of multi-agent systems," *Systems and Control Letters*, vol. 59, no. 10, pp. 587 - 595, 2010.

[107] Y. Liu, P. Ning, M.K. Reiter "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security*, vol. 14, no. 1, 2011.

[108] D. Liuzza, D.V. Dimarogonas, M. Di Bernardo, K.H. Johansson, "Distributed model based event-triggered control for synchronization of multi-agent systems," *Automatica*, vol. 73, pp. 1-7, 2016.

[109] F. Lo Iudice, F. Garofalo, F. Sorrentino, "Structural permeability of complex networks to control signals," *Nature Communications*, vol. 6, art. no. 8349, 2015.

[110] X.Q. Lu, F. Austin., S.H. Chen, "Cluster consensus of nonlinearly coupled multi-agent systems in directed graphs," *Chinese Physics Letters*, vol. 27, no. 5, 2010.

[111] W. Lu, B. Liu, T. Chen, "Cluster synchronization in networks of coupled nonidentical dynamical systems," *Chaos*, vol. 20, no. 1, 2010.

[112] L. Marconi, A. Isidori, A. Serrani, "Autonomous vertical landing on an oscillating platform: An internal-model based approach," *Automatica*, vol. 38, no. 1, pp. 21-32, 2002.

[113] S. Martini, M. Egerstedt, A. Bicchi, "Controllability analysis of multi-agent systems using relaxed equitable partitions," *International Journal of Systems, Control and Communications*, vol. 2 (nos. 1-3), pp. 100-121, 2010.

[114] S. Mascolo, "Modeling the Internet congestion control using a Smith controller with input shaping," *Control Engineering Practice*, vol. 14, no. 4, pp. 425-435, 2006.

[115] F. Mazenc, S.I. Niculescu and M. Krstic, "Lyapunov-Krasovskii functionals and application to input delay compensation for linear time-invariant systems," *Automatica*, International Federation of Automatic Control, 2012.

[116] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, "OpenFlow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev*, vol. 38, pp. 69-74, Mar. 2008.

[117] J. Medved et al., "OpenDaylight: Towards a model-driven SDN controller architecture," in *2014 IEEE 15th International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2014.

[118] J. Mei, W. Ren, G. Ma, "Distributed coordinated tracking for multiple Euler-Lagrange systems," in *Proceedings of the IEEE Conference on Decision and Control*, December 2010.

[119] M. Mesbahi, M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, 2010.

[120] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[121] A. Molina-Garcia, F. Bouffard, D.S. Kirschen, "Decentralized demand-side contribution to primary frequency control," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 411-419, 2011.

[122] S. Monaco, M.D. Normand-Cyrot, "An introduction to motion planning under multirate digital control," in *Decision and Control (CDC), Proceedings of the 31st IEEE Conference on*, Tucson, Arizona, vol. 2, pp. 1780-1785, 1992.

[123] S. Monaco, M.D. Normand-Cyrot, D., "On nonlinear digital control," in *Nonlinear Systems*, pp. 127153. A.J. Fossard and M.D. Normand-Cyrot Eds., Chapman & Hall and Masson, 1997.

[124] S. Monaco, M.D. Normand-Cyrot, "Issues on nonlinear digital control," *European Journal of Control*, vol. 7, pp. 160-177, 2001.

[125] S. Monaco, M.D. Normand-Cyrot, F. Tiefensee, "Sampled-data stabilizing feedback: A PBC approach," *IEEE Transactions on Automatic Control*, 56(4), pp. 907-912, 2011.

[126] S. Monaco, M.D. Normand-Cyrot, V. Tanasa, "Digital stabilization of delayed-input strict-feedforward dynamics," in *Decision and Control (CDC), Proceedings of the IEEE 51st Annual Conference on*, Maui, Hawaii, pp. 7535-7540, 2012.

[127] S. Monaco, L. Ricciardi Celsi, "On Multi-Consensus and Almost Equitable Graph Partions," submitted to *Automatica*, under review, 2018.

[128] N. Monshizadeh, S. Zhang, M.K. Camlibel, "Disturbance decoupling problem for multi-agent systems: A graph topological approach," *Systems and Control Letters*, vol. 76, pp. 35-41, 2015.

[129] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, pp. 169 - 182, 2005.

[130] A. Nagarajan, R. Ayyanar, "Design and strategy for the deployment of energy storage systems in a distribution feeder with penetration of renewable resources," *IEEE Transactions on Sustainable Energy*, 2015.

[131] B.K. Natarajan, "Sparse Approximate Solutions to Linear Systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227-234, 1995.

[132] N. Nisan, A. Ronen, "Algorithmic mechanism design," *Games and Economic Behavior*, vol. 35, no. 1-2, pp. 166-196, 2001.

[133] N. O'Clery, Y. Yuan, G.B. Stan, M. Barahona, "Observability and coarse graining of consensus dynamics through the external equitable partition," *Physical Review E – Statistical, Nonlinear, and Soft Matter Physics*, 88 (4), art. no. 042805, 2013.

[134] G. Oddi, A. Pietrabissa, "A distributed multi-path algorithm for wireless ad-hoc networks based on Wardrop routing," in *Proc. of the 21st Mediterranean Conference on Control and Automation*, pp. 930935, 2013.

[135] G. Oddi, A. Pietrabissa, F. Delli Priscoli, V. Suraci, "A decentralized load balancing algorithm for heterogeneous wireless access networks," in *World Telecommunications Congress 2014 (WTC2014)*, 2014, pp. 16.

[136] G. Oddi, A. Pietrabissa, F. Delli Priscoli, F. Facchinei, L. Palagi, A. Lanna, "A QoE-aware dynamic bandwidth allocation algorithm based on game theory," in *Proceedings of the 23rd Mediterranean Conference on Control and Automation (MED 2015)*, 2015, pp. 979-985.

[137] R. Olfati-Saber, "Global configuration stabilization for the VTOL aircraft with strong input coupling," *Automatic Control, IEEE Transactions on*, vol. 47, no. 11, pp. 1949-1952, 2002.

[138] R. Olfati-Saber, R.M. Murray, "Consensus Protocols for Networks of Dynamic Agents," *Proceedings of the American Control Conference*, vol. 2, pp. 951-956, 2003.

[139] R. Olfati-Saber, R.M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520-1533, 2004.

[140] A. Palomino, P. Castillo, I. Fantoni, R. Lozano, C. Pégard, "Control strategy using vision for the stabilization of an experimental PVTOL aircraft setup," *Control Systems Technology, IEEE Transactions on*, vol. 13, no. 5, pp. 847-850, 2005.

[141] E. Panteley, A. Loria, "Synchronization and Dynamic Consensus of Heterogeneous Networked Systems," in *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3758-3773, 2017.

[142] F. Pasqualetti, A. Bicchi, F. Bullo, "A graph-theoretical characterization of power network vulnerabilities," in *Proc. Of the IEEE American Control Conference*, San Francisco, CA, 2011.

[143] F. Pasqualetti, F. Dorfler, F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715-2729, 2013.

[144] F. Pasqualetti, F. Dorfler, F. Bullo, "Control-theoretic methods for cyberphysical security: Geometric principles for optimal cross-layer resilient control systems," *IEEE Control Systems*, 2009.

[145] A. Pietrabissa, "A reinforcement learning approach to call admission and call dropping control in links with variable capacity," *European Journal of Control*, vol. 17, no. 1, pp. 89-103, 2011.

[146] A. Pietrabissa, V. Suraci, *Automatica*, 2017, in press, doi: 10.1016/j.automatica.2017.07.021.

[147] A. Pietrabissa, L. Ricciardi Celsi, "Discrete-Time Load Balancing Converging to the Wardrop Equilibrium," submitted to the *IEEE Transactions on Automatic Control*, 2017.

[148] A. Pietrabissa, L. Ricciardi Celsi, F. Cimorelli, V. Suraci, F. Delli Priscoli, A. Di Giorgio, S. Monaco, "Lyapunov-based design of a distributed Wardrop load balancing algorithm with application to Software Defined Networking," submitted to the *IEEE Transactions on Control Systems Technology*, 2017.

[149] A. Rahmani, M.A. Jalali, S.H. Pourtakdoust, "Optimal approach to halo orbit control," in *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003.

[150] W. Ren, R.W. Beard, E.M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proceedings of the American Control Conference*, vol. 3, pp. 1859-1864, 2005.

[151] W. Ren, R. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*, Springer, 2008.

[152] W. Ren, "Synchronization of coupled harmonic oscillators with local interaction," *Automatica*, vol. 44, no. 12, pp. 3195 - 3200, 2008.

[153] W. Ren, "Collective motion from consensus with Cartesian coordinate coupling – Part ii: Double-integrator dynamics," *Proceedings of the IEEE Conference on Decision and Control*, pp. 1012 - 1017, December 2008.

[154] W. Ren, "Distributed leaderless consensus algorithms for networked Euler-Lagrange systems," *International Journal of Control*, 82(11), pp. 2137-2149, 2009.

[155] W. Ren, Y. Cao, *Distributed Coordination of Multi-agent Networks*, Communications and Control Engineering Series, Springer-Verlag, London, 2011.

[156] L. Ricciardi Celsi, S. Battilotti, F. Cimorelli, C. Gori Giorgi, S. Monaco, M. Panfili, V. Suraci, F. Delli Priscoli, "A Q-learning based approach to quality of experience control in cognitive Future Internet networks," in *Proc. of the 23rd Mediterranean Conference on Control and Automation (MED15)*, pp. 1045-1052, June 16-19, 2015, Torremolinos, Spain, DOI: 10.1109/MED.2015.7158895.

[157] L. Ricciardi Celsi, R. Bonghi, S. Monaco, M.D. Normand-Cyrot, "On the exact steering of finite sampled nonlinear dynamics with input delays," in *Proc. of the 1st IFAC Conference on Modelling, Identification and Control of Nonlinear Systems (IFAC-PapersOnLine)*, vol. 48, no. 11, pp. 674-679, 2015.

[158] L. Ricciardi Celsi, R. Bonghi, S. Monaco, M.D. Normand-Cyrot, "Sampled-Data Stabilization Around the L_2 Translunar Libration Point," in *Proceedings of the 3rd IAA Conference on University Satellite Missions and CubeSat Workshop & International Workshop on Lean Satellite Standardization*, Roma, Italy, November 30 - December 5, 2015.

[159] A. Di Giorgio, A. Giuseppi, F. Liberati, A. Ornatelli, A. Rabezzano, L. Ricciardi Celsi, "On the optimization of energy storage system placement for protecting power transmission grids against dynamic load altering attacks," in *Proceedings of the 25th Mediterranean Conference on Control and Automation (MED 2017)*, art. no. 7984247, pp. 986-992, 2017, DOI: 10.1109/MED.2017.7984247

[160] I. Rish, G. Grabarnik, *Sparse Modeling: Theory, Algorithms, and Applications*. CRC Press, 2014.

[161] A. Rodriguez-Angeles, H. Nijmeijer, "Mutual synchronization of robots via estimated state feedback: A cooperative approach," *IEEE Trans-*

actions on Control Systems Technology, vol. 9, no. 4, pp. 380 - 386, 2004.

[162] A. Sallahi, M. St-Hilaire, "Optimal Model for the Controller Placement Problem in Software Defined Networks," *IEEE Communications letters*, vol. 19, no. 1, january 2015. DOI: 10.1109/LCOMM.2014.2371014.

[163] G. Santhi, A. Nachiappan, M. Z. Ibrahime, R. Raghunadhane, M.K. Favas, "Q-learning based adaptive QoS routing protocol for MANETs," in *Proc. of IEEE Int. Conf. on Recent Trends in Information Technology (ICRTIT)*, pp.1233-1238, 2011.

[164] M.T. Schaub, N. O'Clery, Y.N. Billeh, J.C. Delvenne, R. Lambiotte, M. Barahona, "Graph partitions and cluster synchronization in networks of oscillators," *Chaos*, vol. 26, art. no 094821, 2016.

[165] R. Sepulchre, M. Jankovic and P.V. Kokotovic, *Constructive Nonlinear Control*. Springer-Verlag, London, 1997.

[166] S. Shah, R. Kothari, "Convergence of the dynamic load balancing problem to Nash equilibrium using distributed local interactions," *Inf. Sci. (Ny).*, vol. 221, pp. 297305, Feb. 2013.

[167] G.A. Shanholt, "Set stability for difference equations," *International Journal of Control*, vol. 19, no. 2, pp. 309-314, 1974.

[168] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, *Robotics: Modelling, Planning and Control*, Springer, 2009.

[169] S. Singh, J. G. Andrews, G. de Veciana, "Interference shaping for improved quality of experience for real-time video streaming," *IEEE J. Sel. Areas Commun*, vol. 30, no. 7, pp. 1259-1269, 2012.

[170] O.J. Smith, "A controller to overcome dead time," *ISA Journal*, volume 6, pages 28-33, 1959.

[171] J. Stamp, A. McIntyre, and B. Ricardson, "Reliability impacts from cyber attacks on electric power systems," in *Power Systems Conference and Exposition, 2009, PSCE09, IEEE/PES*, IEEE, 2009, pp. 1-8.

[172] R. Subrata, A.Y. Zomaya, B. Landfeldt, "A Cooperative Game Framework for QoS Guided Job Allocation Schemes in Grids," *IEEE Trans. Comput.*, vol. 57, no. 10, pp. 14131422, Oct. 2008.

[173] H. Su, X. Wang, Z. Lin, "Synchronization of coupled harmonic oscillators in a dynamic proximity network," *Automatica*, vol. 45, no. 10, pp. 2286 - 2291, 2009.

[174] R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Massachusetts, 1998.

[175] V. Tanasa, S. Monaco, M.D. Normand-Cyrot, "Digital stabilization of finite sampled nonlinear dynamics with delays: the unicycle example," in *Proc. of the 2013 European Control Conference (ECC)*, Zurich, Switzerland, pp. 2591-2596, 2013.

[176] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, O. Shochet, "Novel Type of Phase Transition in a System of Self-Driven Particles," *Phys. Rev. Lett.*,75 (6), pp. 1226-1229, 1995.

[177] M.A. Viscio, N. Viola, S. Corpino, F. Stesina, S. Fineschi, F. Fumenti, C. Circi, "Interplanetary CubeSats system for space weather evaluations and technology demonstration," *Acta Astronautica*, vol. 104, no. 2, pp. 516-525, 2014.

[178] J.G. Wardrop, "Some Theoretical Aspects of Road Traffic Research, *ICE Proc. Eng. Div.*, vol. 1, no. 3, pp. 325362, Jan. 1952.

[179] C.J.C.H. Watkins, P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279-292, 1992.

[180] L. Xie, Y. Mo, B. Sinopoli, "False data injection attacks in electricity markets," in *IEEE International Conference on Smart Grid Communications*, Brussels, Belgium, 2011.

[181] J. Xu, C. Tekin, S. Zhang, M. van der Schaar, "Distributed multi-agent online learning based on global feedback," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2225-2238, 2015.

[182] J.W. Yi, Y.W. Wang, J.W. Xiao, "Reaching cluster consensus in multi-agent systems," *Proceedings of the 2nd International Conference on Intelligent Control and Information Processing, ICICIP 2011*, pp. 569-573, 2011e.

[183] J. Yu, L. Wang, "Group consensus in multi-agent systems with switching topologies and communication delays," *Systems and Control Letters*, vol. 59, no. 6, pp. 340-348, 2010.

[184] Y. Zhang, Y.P. Tian, "Consentability and protocol design of multi-agent systems with stochastic switching topology," *Automatica*, vol. 45, no. 5, pp. 1195 - 1201, 2009.

[185] S. Zhang, M. Kao, M. Kanat Camlibel, "Upper and Lower Bounds for Controllable Subspaces of Networks of Diffusively Coupled Agents," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 745-749, 2014.

[186] C. Zhao, U. Topcu, S.H. Low, "Optimal load control via frequency measurement and neighborhood area communication," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3576-3587, 2013.

[187] I. Zliobaité, "Learning under concept drift: an overview," *arXiv preprint arXiv:1010.4784*, 2010.

[188] L. Zuccaro, F. Cimorelli, F. Delli Priscoli, C. Gori Giorgi, S. Monaco, V. Suraci, "Distributed Control in Virtualized Networks," in *Proc. of the 10th International Conference on Future Networks and Communications (FNC 2015)*, pp. 276-283, 2015.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Commande non linéaire multi-agents : applications aux systèmes en réseau.

Mots clés : Stabilisation, systèmes non linéaires, systèmes discrèts et à données échantillonées, systèmes multi-agents, réseaux complexes, consensus.

Résumé : L'objectif de cette thèse de doctorat est (i) d'étudier et de développer des méthodes d'analyse et de commande de systèmes de contrôle en réseau linéaires et non linéaires et (ii) de montrer le potentiel de ces approches dans des applications complexes pertinentes. À cet égard, la théorie des systèmes à plusieurs agents, la théorie des graphes algébriques et le consensus sont des outils méthodologiques les plus intéressants. Une attention particulière est accordée à la caractérisation des relations entre, d'une part, la topologie du graphe de communication qui sous-tend l'évolution du système à plusieurs agents considéré et, d'autre part, les propriétés spectrales de la matrice Laplacienne associée au graphe lui-même. Le contrôle d'un groupe d'agents autonomes est étudié sous différents angles.

Le principal objectif de contrôle est de s'assurer que les agents travaillent ensemble de manière coopérative, où la coopération représente la relation étroite entre tous les agents de l'équipe, le partage de l'information jouant un rôle important.

En particulier, beaucoup de problèmes de consensus/accord/ synchronisation /rendez-vous sont étudiés afin de guider un groupe d'agents vers un état commun. Le consensus est étudié dans un contexte à temps discret parce que la dynamique du système est en general continue alors que les mesures et les entrées de contrôle sont des données échantillonnées. En outre, la théorie des jeux est utilisée pour faire face aux problèmes de coordination distribués à plusieurs agents, avec une application aux réseaux connus sous le nom de Software Defined Networks. À cet égard, on peut montrer que, sous des protocoles correctement conçus, les joueurs convergent vers un équilibre unique de Wardrop.

On concentre l'attention sur le contrôle distribué, car cette approche présente des avantages évidents par rapport à la centralisation, comme l'évolutivité et la robustesse. Pourtant, le contrôle distribué a également ses propres inconvénients : avant tout, un inconvénient est que chaque agent ne peut pas prédire efficacement le comportement global du groupe en se basant uniquement sur des informations locales.

Une certaine attention est également accordée à la nécessité de sécuriser les réseaux électriques contre le danger des attaques cyber-physiques grâce au développement de technologies d'intelligence distribuée. À cet égard, sur la base de topologies de réseaux d'énergie réalistes, nous présentons brièvement la conception d'un schéma de protection contre les attaques dynamiques à un point et à points multiples en boucle fermée. Nous formulons et résolvons un problème d'optimisation non convexe soumis à une contrainte de stabilité de Lyapunov pour la représentation à plusieurs agents autonome d'un réseau électrique obtenue après la linéarisation et l'application des lois d'attaque et de contrôle de fréquence.

Finalement, nous présentons des résultats obtenus sur : le pilotage exact de la dynamique non linéaire finie à données échantillonnées avec des retards sur les entrées, au sujet de la stabilisation à données échantillonnées et de la poursuite de l'orbite quasi-halo autour du point de libration translunaire L_2, et au sujet des algorithmes heuristiques basés sur des méthodes d'apprentissage par renforcement à plusieurs agents capables d'effectuer un contrôle adaptatif optimal de qualité de service / qualité de l'expérience dans des scénarios sans modèle.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title : Nonlinear Multi-Agent Control with Application to Networked Systems

Keywords : Stabilization, nonlinear systems, discrete-time and sampled-data control systems, multi-agent systems, complex networks, consensus.

Abstract : The objective of this PhD thesis is (i) to investigate and develop methods for the analysis and design of linear and nonlinear networked control systems and (ii) to show the potential of such approaches in relevant complex applications. In this respect, multi-agent systems theory, algebraic graph theory and consensus are the most interesting methodological tools, and specific attention is paid to the characterization of the relationships between, on the one hand, the topology of the communication graph that underlies the evolution of the considered multi-agent system and, on the other hand, the spectral properties of the Laplacian matrix associated with the graph itself.

The control of a group of autonomous agents is investigated from different perspectives. The main control objective is to make sure that the agents work together in a cooperative fashion, where cooperation accounts for the close relationship among all agents in the team, with information sharing playing an important role.

In particular, various problems regarding consensus/agreement/synchronization/rendezvous are investigated with the specific aim of driving a group of agents to some common state. Consensus is investigated in a discrete-time setting due to the fact that the system dynamics is normally continuous while the measurements and control inputs might only be made in a sampled-data setting. Moreover, game theory is relied upon in order to cope with distributed multi-agent coordination problems, with application to Software Defined Networks. In this respect, it can be shown that, under properly designed protocols, the players converge to a unique Wardrop equilibrium.

We focus on distributed control, since this approach shows obvious benefits over centralization, such as scalability and robustness. Yet, it also has its own drawbacks: among all, one drawback is that each agent cannot effectively predict the overall group behaviour based on only local information.

Some attention is also devoted to the need for securing power grids against the danger of cyber-physical attacks through the development of distributed intelligence technologies accompanied by appropriate security enforcements. In this respect, based on realistic power network topologies, we briefly present the design of a protection scheme against closed-loop single-point and multi-point dynamic load altering attacks. This is done by formulating and solving a non-convex optimization problem subject to a Lyapunov stability constraint for the autonomous multi-agent representation of a power system obtained after linearization and application of the attack and frequency control laws.

Eventually, we show some other results achieved in terms of the exact steeering of finite sampled nonlinear dynamics with input delays, of sampled-data stabilization and quasi-halo orbit following around the L_2 translunar libration point, and of heuristic algorithms based on multi-agent reinforcement learning methods capable of performing optimal adaptive Quality of Service/Quality of Experience control in model-free scenarios.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France